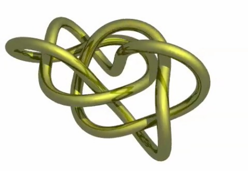
QFT and complexity of link invariants of quantum doubles of finite groups

Hari Krovi
(joint work with Alexander Russell)

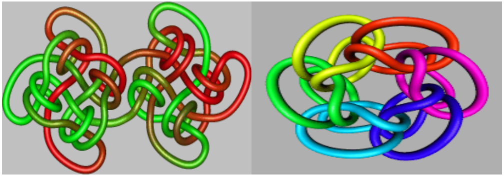
Quantum Information Processing group
Raytheon BBN Technologies

Knots and links

Mathematically, a knot is an embedding of S^1 into $\mathbb{R}^3$ such that it is invariant up to ambient isotopy.

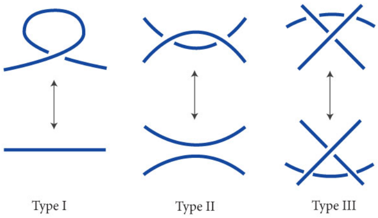


A link is an embedding of many copies of S^1 i.e., many pieces of string, which could be knotted with each other.



Link invariants

Equivalence of knots - Reidemeister moves



- A link invariant is a function from the link (to the complex numbers) such that if two links are equivalent, then the numbers are the same.
- Possible that two non-equivalent links have the same numbers.

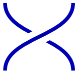
Braid group

B_n is generated by σ_i and their inverses subject to the following conditions.

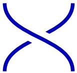
$$\sigma_i \sigma_j = \sigma_j \sigma_i \quad \text{for } |i - j| \geq 2$$

and

$$\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}$$



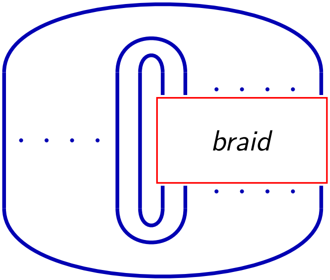
(a) σ_i



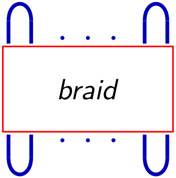
(b) σ_i^{-1}

Links from braids

- Any link can be formed from a braid by closing the strands of the braid.
- Braids on n strands form an infinite group called the braid group (B_n) generated by σ_i and σ_i^{-1}.



(c) The trace closure.



(d) The plat closure.

Algebraic approach to link invariants

- If one has a braid group representation, then by taking the normalized trace of b one can construct a link invariant.
- The trace should satisfy Markov properties.
- One way to produce braid group representations is via the Yang Baxter Equation (YBE).

$$(R \otimes I)(I \otimes R)(R \otimes I) = (I \otimes R)(R \otimes I)(I \otimes R)\,.$$

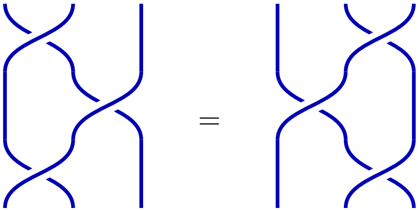


Figure: The Yang-Baxter relations $\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}$.

Quantum double or Drinfeld double

- Drinfeld defined the quantum double of two braided Hopf algebras as a way to construct solutions of the QYBE.
- For finite groups, the quantum double looks like the semidirect product.

$$(g_1 h_1^*)(g_2 h_2^*) = \delta(h_1^{g_2}, h_2) g_1 g_2 h_2^* \,.$$

- This generates a finite dimensional algebra denoted $\mathrm{D}(G)$, from which one gets the R matrix (solution of YBE).

$$R = \sum_g g \otimes g^* \,.$$

Quantum double or Drinfeld double

- Drinfeld defined the quantum double of two braided Hopf algebras as a way to construct solutions of the QYBE.
- For finite groups, the quantum double looks like the semidirect product.

$$(g_1 h_1^*)(g_2 h_2^*) = \delta(h_1^{g_2}, h_2) g_1 g_2 h_2^* \,.$$

- This generates a finite dimensional algebra denoted $\mathrm{D}(G)$, from which one gets the R matrix (solution of YBE).

$$R = \sum_g g \otimes g^* \,.$$

- The R matrix generates σ_i and thus the braid group. So for any representation V of the quantum group, we get a representation of B_n on $V^{\otimes n}$.
- The (non) denseness of this representation of B_n in $U(\dim(V)^n)$ depends on the quantum group. For $\mathrm{D}(G)$, it is finite.

Dense invariants

- About 12 years ago, in a series of papers, certain link invariants were shown to be closely related to quantum computing.
- Algorithms to additively approximate link invariants were found (Freedman-Kitaev-Wang, Aharonov-Jones-Landau, Wocjan-Yard).
- Additive approximations of dense invariants such as the Jones polynomial were shown to be BQP complete. Exactly computing them was shown to $\#$P complete.

Dense invariants

- About 12 years ago, in a series of papers, certain link invariants were shown to be closely related to quantum computing.
- Algorithms to additively approximate link invariants were found (Freedman-Kitaev-Wang, Aharonov-Jones-Landau, Wocjan-Yard).
- Additive approximations of dense invariants such as the Jones polynomial were shown to be BQP complete. Exactly computing them was shown to #P complete.
- Kuperberg showed that one can obtain the complexity of additive, multiplicative approximations and exact computations using denseness.
- Any quantum computation can be arbitrarily close to the plat closure of a braid in the dense representation. So additive approximations are BQP hard, multiplicative SBQP hard and exact #P hard.
- Finally, density implies that any anyonic computer can be simulated efficiently using the circuit model.

Additive and multiplicative approximations

For a function $f(x)$, if the output $g(x)$ of any probabilistic algorithm can be mainly of two kinds.

- **Additive approximation**

$$\Pr[|f(x) - g(x)| > \epsilon u(|x|)] < 1/4\,,$$

  where u is a normalization.

- **Multiplicative approximation**

$$\Pr[|f(x) - g(x)| > \epsilon f(x)] < 1/4\,.$$

Our results on D(G)

Algorithms:

- We develop the quantum Fourier transform over D(G) subject to the condition that one can do QFTs over centralizer subgroups. We show explicitly that this can be done for D(S_n).
- We use this to give efficient additive approximations of link invariants coming from D(G).

Our results on D(G)

Algorithms:

- We develop the quantum Fourier transform over D(G) subject to the condition that one can do QFTs over centralizer subgroups. We show explicitly that this can be done for D(S_n).
- We use this to give efficient additive approximations of link invariants coming from D(G).

Complexity

- We show that for certain kinds of irreps (fluxons), the value of the plat closure of a link can be made arbitrarily close to the success probability of a randomized computation.
- This implies (like Kuperberg's result) that additive approximations are BPP hard, multiplicative SBP hard and exact computations are #P hard.
- However, we needed to assume that the group G be of fixed size.

Our results on D(G)

Simulation

- In order to simulate a D(G) computer efficiently, one needs to (in addition to the QFT) perform the Clebsch-Gordan transform over D(G).
- We show that this can be done for fluxon irreps.
- We show that for general irreps, this can be done subject to some conditions - such as CG over centralizers and another transform over intersections of centralizers.
- We show that for D($\mathbb{Z}_p \rtimes \mathbb{Z}_q$), this can be done for all irreps. Here p and q are prime and $q|(p-1)$.
- This quantum group has been shown to be universal for quantum computing (Mochon).

Algorithms over D(G)

- The irreducible representations of the quantum double are all induced representations of the group G.
- For any element $g \in G$, the centralizer subgroup is the the set $C_G(g) = \{z \in G | zg = gz\}$.
- Suppose that ρ is an irrep of $C_G(g)$, then the irreducible representations of D(G) are of the type $\uparrow_{C_G(g)}^{G} \rho$.
- If ρ is the trivial irrep, the $\uparrow_{C_G(g)}^{G} \rho$ is called a **fluxon** irrep.

Algorithms over D(G)

- The irreducible representations of the quantum double are all induced representations of the group G.
- For any element $g \in G$, the centralizer subgroup is the the set $C_G(g) = \{z \in G | zg = gz\}$.
- Suppose that ρ is an irrep of $C_G(g)$, then the irreducible representations of D(G) are of the type $\uparrow_{C_G(g)}^{G} \rho$.
- If ρ is the trivial irrep, the $\uparrow_{C_G(g)}^{G} \rho$ is called a **fluxon** irrep.
- We reduce the problem of constructing the QFT over D(G) to that of constructing a QFT over $C_G(g)$ for each g.
- Since $C_G(e) = G$, this involves knowing the QFT over G as well.
- When $G = S_n$, we get $C_{S_n}(\pi) = \mathbb{Z}_k \wr S_{c_k}$. For these groups, we give an explicit transversal and QFT using Clifford theory.

Complexity

- For this, we take the group size to be fixed and focus on fluxon irreps.
- First, we take an arbitrary randomized computation and write its probability of success as

$$P_s = \langle \phi | R | \phi \rangle \,, \quad |\phi\rangle = \frac{1}{\sqrt{d^m}} \sum_r |r, c\rangle \,,$$

where R is a reversible deterministic computation. r is the random $d$it string of length m and c is a string of zeros.

Complexity

- For this, we take the group size to be fixed and focus on fluxon irreps.
- First, we take an arbitrary randomized computation and write its probability of success as

$$P_s = \langle \phi | R | \phi \rangle \,, \quad |\phi\rangle = \frac{1}{\sqrt{d^m}} \sum_r |r, c\rangle \,,$$

where R is a reversible deterministic computation. r is the random $d$it string of length m and c is a string of zeros.
- The plat closure has a similar expression

$$Pl = \langle \psi^{\otimes n} | B | \psi^{\otimes n} \rangle \,, \quad |\psi\rangle = \frac{1}{|C|} \sum_{g \in C} |g, g^{-1}\rangle$$

Complexity

- Using the Ogburn-Preskill encoding, the d levels are on two anyons and are of the form $|g, g^{-1}\rangle$.
- So the probability of success is now

$$P_s = \langle \Phi | R | \Phi \rangle \,, \quad |\Phi\rangle = \frac{1}{\sqrt{d^m}} \sum_{\bar{g}} |(c, c^{-1})^k g_1, g_1^{-1} \ldots g_m, g_m^{-1}\rangle \,.$$

Complexity

- Using the Ogburn-Preskill encoding, the d levels are on two anyons and are of the form $|g, g^{-1}\rangle$.
- So the probability of success is now

$$P_s = \langle \Phi | R | \Phi \rangle \,, \quad |\Phi\rangle = \frac{1}{\sqrt{d^m}} \sum_{\bar{g}} |(c, c^{-1})^k g_1, g_1^{-1} \dots g_m, g_m^{-1}\rangle \,.$$

- To generate group constants c, we generate equations whose solutions are the group constants.

$$x_i = x_1^w \,, \quad \text{where } w \text{ is a word in the } x_i$$

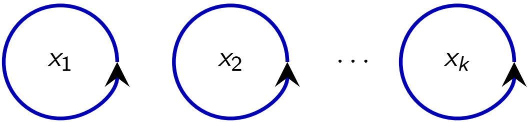


Figure: Initial circles.

Complexity

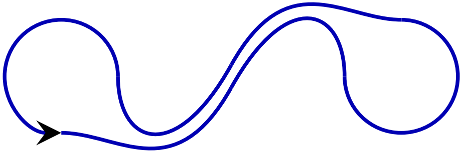


Figure: A band between two circles.

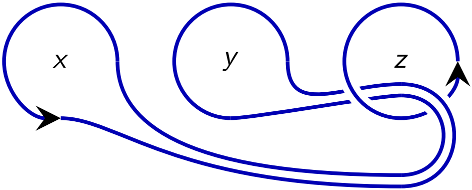


Figure: A simple relation: $x^z = y$.

Complexity

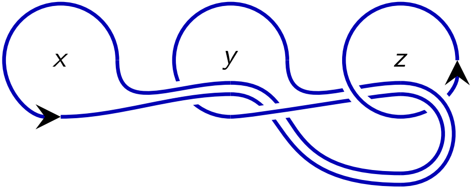


Figure: The equation $x^{zy} = y$.

- To kill unwanted solutions, we generate equations of the type $y = x_1^{w(x)y}$ such that $w(d) = 1$ and $w(c) = \alpha$ (Here c is the wanted solution and d is unwanted).
- We show that there are simple groups such as A_n, which have non-trivial α such that the equation $y = x_1^{\alpha y}$ has multiple solutions.

Simulation

- We need three capabilities in order to do universal quantum computation.
- Prepare any state in the Hilbert space of a pair of anyons which correspond to conjugate irreps.
- Perform braiding of anyons around each other and around ancillas.
- Fuse pairs of anyons and measure the flux and charge of the resulting particle.
- In order to simulate anyonic computation, we need to simulate these on the circuit model.
- In this part, we assume that the group size is asymptotically growing again.

Simulation

- In order to simulate this using the circuit model, we only need to focus on the last of the conditions.
- The last one can be done if we can do the Clebsch-Gordan transform over this group.
- The CG transform is a unitary that breaks up a tensor product of irreps into irreps.
- We use a tensor product theorem and adapt it to our situation.

$$\rho \uparrow_H^G \otimes \sigma \uparrow_K^G = \bigoplus_d (\rho \downarrow_{H \cap K^d} \otimes \sigma \downarrow_{H \cap K^d}) \uparrow^G$$

- For fluxon irreps, we obtain a transform.
- For dyons, we obtain a transform assuming one can do CG transforms over centralizers etc.

Conclusions and open problems

- The denseness (or the lack of it) seems to be related to the complexity of approximating the link invariant.
- Also related to the computational power of the anyonic system.
- For dense invariants, the relationship is clearer, whereas little is known for non-dense invariants.
- Could lead to insights into what kind of gates sets lead to a certain computational power.

Conclusions and open problems

- The denseness (or the lack of it) seems to be related to the complexity of approximating the link invariant.
- Also related to the computational power of the anyonic system.
- For dense invariants, the relationship is clearer, whereas little is known for non-dense invariants.
- Could lead to insights into what kind of gates sets lead to a certain computational power.
- Extend the hardness result to asymptotically growing groups (need new techniques).
- Extend the Clebsch-Gordan transform to other groups.
- These techniques could help with other problems as they involve finite groups.