Quantum Hashing with the Icosahedral Group

Michele Burrello,1,2 Haitan Xu,3 Giuseppe Mussardo,1,4,2 and Xin Wan5,6,3

1International School for Advanced Studies (SISSA), Via Beirut 2-4, I-34014 Trieste, Italy
2Instituto Nazionale di Fisica Nucleare, Sezione di Trieste, Italy
3Zhejiang Institute of Modern Physics, Zhejiang University, Hangzhou 310027, P.R. China
4International Centre for Theoretical Physics (ICTP), I-34014 Trieste, Italy
5Asia Pacific Center for Theoretical Physics (APCTP), Pohang, Gyeongbuk 790-784, Korea
6Department of Physics, Pohang University of Science and Technology, Pohang, Gyeongbuk 790-784, Korea

(Dated: March 24, 2010)

We study an efficient algorithm to hash any single qubit gate (or unitary matrix) into a braid of Fibonacci anyons represented by a product of icosahedral group elements. By representing the group elements by braid segments of different lengths, we introduce a series of pseudo-groups. Joining these braid segments in a renormalization group fashion, we obtain a Gaussian unitary ensemble of random-matrix representations of braids. With braids of length $O((\log(1/\varepsilon))^2)$, we can approximate all SU(2) matrices to an average error $\varepsilon$ with a cost of $O((\log(1/\varepsilon))^2)$ in time. The algorithm is applicable to generic quantum compiling.

PACS numbers:

Quantum gates are the building blocks for quantum circuits. A reliable implementation of quantum computation would need a universal set of fault-tolerant gates. How to use the set of universal gates to construct quantum circuits is an important question. The question also arises if we want to simulate the circuits of the universal set by using those of another set. The Solovay-Kitaev algorithm guarantees good approximations to any desired gates, provided that a dense $\varepsilon$-net exists. Instead of using quantum error-correction codes, topological quantum computation proposes to realize fault-tolerant quantum gates by topology embedded in hardware. In two-dimensional topological states of matter, a collection of non-Abelian anyonic excitations with fixed positions spans a multi-dimensional Hilbert space and, in such a space, the quantum evolution of the multi-component wave function of the anyons is realized by their braidings. The evolution can be represented by non-trivial unitary matrices that implement quantum computation. A prototype of non-Abelian anyons is known as the Fibonacci anyons, which exist in the Read-Rezayi quantum Hall state at filling fraction $\nu = 3/5$ (whose particle-hole conjugate is a candidate for the observed $\nu = 12/5$ quantum Hall plateau) and in the non-Abelian spin-singlet state at $\nu = 4/7$. In topological quantum computation, the topology of the quantum braids precludes errors induced by local noises; unfortunately, this does not eliminate the errors in approximating quantum gates by braids.

Bonesteel et al. pioneered the implementation of quantum gates using Fibonacci anyons with a brute-force search algorithm, which finds the best approximation to a unitary matrix $T$ in the set of all braids up to a certain length $L$. As for all quantum computation schemes, the complexity (thus inefficiency) in brute-force search is dictated by the necessity to sample the whole space of unitary matrices with almost equal weight, while the target gate is just a zero-measure point inside. Thus the distance of the approximation depends on $L$ as $e^{-L/\xi}$ (with $\xi \approx 7.3$). However, the run time grows exponentially in $L$, rendering the algorithm impractical to achieve a distance below a certain threshold. In fact, the most probable braids generated by the brute-force algorithm have the largest distance to the desired gate due to the geometry of the unitary matrix space, as illustrated in Fig. 1. Subsequent algorithms enhance the sampling of the target point by mapping it to a higher-dimensional object, although the search remains time-consuming. The inefficiency in these algorithms is also reflected in the fact that a new unitary matrix needs a new brute-force search, which is exponentially hard. The existing implementation of the Solovay-Kitaev algorithm is not efficient enough in terms of either braid length or searching time.

The question is thus the following: can one implement a more efficient search algorithm to find braids for single-qubit gates? Technically, we can think of a braid as an index to the corresponding unitary matrix, which can be regarded as a definition, like in a dictionary. Given an index, it is straightforward to find its definition, but finding the index for a definition is exponentially hard. In computer science, the task of quickly locating a data record given its content (or search key) can be achieved by the introduction of hash functions. In the context of topological quantum computation, we thus name this task topological quantum hashing. In general, such a hashing function, being imperfect, still maps a unitary matrix to a number of braids rather than one. But narrowing the search down to only a fixed (rather than exponentially large) number of braids is already a great achievement.

In this Letter, we explore topological quantum hashing with the finite icosahedral group $I$ and its algebra. The building blocks of the algorithm are a preprocessor and a main processor; the aim of the preprocessor is to give an initial approximation $\tilde{T}$ of the target gate $T$, while that of the main processor is to reduce the discrepancy between $T$ and $\tilde{T}$ with extremely high efficiency. We discuss the iteration of the algorithm in a renormalization group fashion and the results which follow from this approach. The algorithm is also applicable to...
generic quantum compiling and, remarkably, its efficiency can be quantified using random matrix theory.

We illustrate our algorithm with Fibonacci anyons (denoted as $\phi$, with a fusion rule $\phi \times \phi = 1 + \phi$, where 1 is the vacuum) [11–13, 15, 16]. If we create two pairs of $\phi$ (illustrated graphically by dots) out of the vacuum, both pairs (small ellipses) must have the same fusion outcome, 1 or $\phi$, forming a qubit (large ellipse), in which the braiding of $\phi$'s can be generated by two fundamental braiding matrices

\[
\sigma_1 = \begin{bmatrix}
e^{-i4\pi/5} & 0 \\
0 & -e^{-i2\pi/5}
\end{bmatrix},
\]
\[
\sigma_2 = \begin{bmatrix}
-\tau e^{-i2\pi/5} & \sqrt{\tau} e^{i2\pi/5} \\
-\sqrt{\tau} e^{i2\pi/5} & -\tau
\end{bmatrix},
\]

and their inverses $\sigma_1^{-1}$, $\sigma_2^{-1}$. Here $\tau = (\sqrt{5} - 1)/2$. The matrix representation generates a four-strand braid group $B_4$ (or an equivalent three-strand braid group $B_3$): this is an infinite dimensional group consisting of all possible sequences of length $L$ of the above generators and with increasing $L$ the whole set of braidings generates a dense cover of the SU(2) single-qubit rotations. Earlier works [11, 13, 15, 16] have demonstrated that the two-qubit gate construction can be mapped to the single-qubit gate construction; thus, we will not discuss the construction of two-qubit gates here.

**Icosahedral group.** The icosahedral rotation group $\mathcal{I}$ of order 60 is the largest finite subgroup of SU(2) excluding reflection. Therefore, it has been often used to replace the full SU(2) group for practical purposes, as for example in earlier Monte Carlo studies of SU(2) lattice gauge theories [17], and this motivated us to apply the icosahedral group representation in the braid construction. $\mathcal{I}$ is composed by the 60 rotations around the axes of symmetry of the icosahedron (platonic solid with twenty triangular faces) or of its dual polyhedron, the dodecahedron (regular solid with twelve pentagonal faces); there are six axes of the fifth order, ten of the third and fifteen of the second. Let us for convenience write $\mathcal{I} = \{g_0, g_1, \ldots, g_{59}\}$, where $g_0 = e$ is the identity element.

Thanks to the homomorphism between SU(2) and SO(3), we start by associating a $2 \times 2$ unitary matrix to each group element. In other words, each group element can be approximated by a braid of Fibonacci anyons of a certain length $N$ using the brute-force search [11] and neglecting an overall phase. In this way, we obtain an approximate representation in SU(2) of the icosahedral group, $\tilde{\mathcal{I}}(N) = \{\tilde{g}_0(N), \tilde{g}_1(N), \ldots, \tilde{g}_{59}(N)\}$. Choosing, for instance, a fixed braid length of $N = 24$, the distance (or error) of each braid representation to its corresponding exact matrix representation varies from 0.003 to 0.094 (see Fig. 2 for an example).

We point out that the 60 elements of $\tilde{\mathcal{I}}(N)$ (for any finite $N$) do not close any longer the composition laws of $\mathcal{I}$; in fact, they form a pseudo-group, not a group, isomorphic to $\mathcal{I}$ only in the limit $N \to \infty$. In other words, if the composition law $g_i g_j = g_k$ holds in the original icosahedral group, the product of the corresponding elements $\tilde{g}_i(N)$ and $\tilde{g}_j(N)$ is not $\tilde{g}_k(N)$, although it can be very close to it for large enough $N$. Interestingly, the distance between the product $\tilde{g}_i(N) \tilde{g}_j(N)$ and the corresponding element $g_k$ of $\mathcal{I}$ can be linked to the Wigner-Dyson distribution, which we will discuss later.

Using the pseudo-group structure of $\tilde{\mathcal{I}}$, we can generate a set $\mathcal{S}$ made of a large number of braids only in the vicinity of the identity matrix: this is a simple consequence of the original icosahedral group algebra, in which the composition laws allow us to obtain the identity group element in various ways. The set $\mathcal{S}$ is instrumental to achieve an important goal, i.e. to search among the elements of $\mathcal{S}$ the best correction to apply to a first rough approximation of the target single-qubit gate $T$ we want to hash. We can create such a set, labeled by $\mathcal{S}(L, n)$, considering all the possible ordered products $\tilde{g}_{i_1}(L) \tilde{g}_{i_2}(L) \ldots \tilde{g}_{i_n}(L)$ of $n \geq 2$ elements of $\tilde{\mathcal{I}}(L)$ of

![FIG. 2:](color online) Approximation to the $-iX$ gate (an element of the icosahedral group) in terms of braids of the Fibonacci anyons of length $L = 24$ in the graphic representation. In this example the error is 0.0031.
length \( L \) and multiplying them by the matrix \( \hat{g}_{n+1}(L) \in \hat{I}(L) \) such that \( g_{n+1}^{-1} = g_{n}^{-1} \cdot g_{i_1}^{-1} \cdot g_{i_2}^{-1} \cdot \cdots \cdot g_{i_m}^{-1} \). In this way we generate all the possible combinations of \( n + 1 \) elements of \( \hat{I} \) whose result is the identity, but, thanks to the errors that characterize the braid representation \( \hat{I} \), we obtain \( 60^n \) small rotations in SU(2), corresponding to braids of length \((n + 1)L\).

The hashing procedure. The first step in the hashing procedure of the target gate is to find a rough braid representation of \( T \) using a preprocessor, which associates to \( T \) the element in \( [\hat{I}(l)]^m \) (of length \( m \times l \)) that best approximates it. Thus we obtain a starting braid

\[
\hat{T}_{0}^{l,m} = \hat{g}_{j_{1}}(l) \hat{g}_{j_{2}}(l) \cdots \hat{g}_{j_{m}}(l)
\]

characterized by an initial error we want to reduce. The preprocessor procedure relies on the fact that choosing a small \( l \) we obtain a substantial discrepancy between the elements \( g \) of the icosahedral group and their representatives \( \hat{g} \). Due to these random errors the set \([\hat{I}(l)]^m\) of all the products \( \hat{g}_{j_{1}} \hat{g}_{j_{2}} \cdots \hat{g}_{j_{m}} \) is well spread all over SU(2) and can be considered as a random discretization of this group.

In the main processor we use the set of fine rotations \( S(L, n) \) to efficiently reduce the error in \( \hat{T}_{0}^{l,m} \). Multiplying \( \hat{T}_{0}^{l,m} \) by all the elements of \( S(L, n) \), we generate \( 60^n \) possible braid representations of \( T \):

\[
\hat{T}_{0}^{l,m} \hat{g}_{j_{1}} \hat{g}_{j_{2}} \cdots \hat{g}_{j_{n+1}}
\]

Among these braids of length \((n + 1) L + ml\), we search the one which minimizes the distance with the target gate \( T \). This braid, \( \hat{T}_{0}^{l,m} \), is the result of our algorithm. Fig. 3 shows the distribution of final errors for 10,000 randomly selected target gates obtained with a preprocessor of \( l = 8 \) and \( m = 3 \) and a main processor of \( L = 24 \) and \( n = 3 \).

To illustrate our algorithm, it is useful to consider a concrete example: suppose we want to find the best braid representation of the target gate

\[
T = iZ = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix}
\]

Out of all combinations in \([\hat{I}(8)]^3\), the preprocessor selects a \( \hat{T}_{0}^{8,3} = \hat{g}_{p_{1}}(8) \hat{g}_{p_{2}}(8) \hat{g}_{p_{3}}(8) \), which minimizes the distance to \( T \) to 0.038. Applying now the main processor, the best rotation in \( S(24,3) \) that corrects \( \hat{T}_{0}^{8,3} \) is given by a \( \hat{g}_{q_{1}}(24) \hat{g}_{q_{2}}(24) \hat{g}_{q_{3}}(24) \), where \( g_{q_{1}} = g_{q_{2}}^{-1} g_{q_{3}}^{-1} \).

The resulting braid \([18]\) is then represented by

\[
\hat{T}_{24,3}^{8,3} = \hat{g}_{p_{1}}(8) \hat{g}_{p_{2}}(8) \hat{g}_{p_{3}}(8) \hat{g}_{q_{1}}(24) \hat{g}_{q_{2}}(24) \hat{g}_{q_{3}}(24)\hat{g}_{q_{4}}(24)
\]

for the special set of \( p \)'s and \( q \)'s and, apart from an overall phase, the final distance is reduced to 0.00099 (Fig. 3).

**Relationship with random matrix theory.** The distribution of the distance between the identity and the so-obtained braids has an intriguing connection to the Gaussian unitary ensemble of random matrices, which helps us to understand how close we can approach the identity in this way, i.e. the efficiency of the hashing algorithm. Let us analyze the group property deviation for the pseudo-group \( \hat{I}(N) \) for braids of length \( N \). One can write \( \hat{g}_{i} = g_{i} e^{i\Delta_{i}} \), where \( \Delta_{i} \) is a Hermitian matrix, indicating the small deviation of the finite braid representation to the corresponding SU(2) representation for an individual element. For a product of \( \hat{g}_{i} \) that approximate \( g_{i_{1}} g_{i_{2}} \cdots g_{n+1} = e \), one has

\[
\hat{g}_{i_{1}} \hat{g}_{i_{2}} \cdots \hat{g}_{i_{n+1}} = g_{i_{1}} e^{i\Delta_{i_{1}}} g_{i_{2}} e^{i\Delta_{i_{2}}} \cdots g_{i_{n+1}} e^{i\Delta_{i_{n+1}}} = e^{i\Delta_{i}}
\]

where \( H_{n} \), related to the accumulated deviation, is

\[
H_{n} = g_{i_{1}} \Delta_{i_{1}} g_{i_{1}}^{-1} + g_{i_{2}} \Delta_{i_{2}} g_{i_{2}}^{-1} + \cdots + g_{i_{n}} \Delta_{i_{n}} g_{i_{n}}^{-1} + \Delta_{i_{n+1}} + O(\Delta^{2})
\]

The natural conjecture is that, for a long enough sequence of matrices product, the Hermitian matrix \( H_{n} \) tends to a random matrix corresponding to the Gaussian unitary ensemble. This is plausible as \( H_{n} \) is a Hermitian matrix that is the sum of random initial deviation matrices with random unitary transformations. A direct consequence is that the distribution of the eigenvalue spacing \( s \) obeys the Wigner-Dyson form \([19]\),

\[
P(s) = \frac{32}{\pi^{2} s_{0}} \left( \frac{s}{s_{0}} \right)^{2} e^{-(4/\pi)(s/s_{0})^{2}}
\]

where \( s_{0} \) is the mean level spacing. For small enough deviations, the distance of \( H_{n} \) to the identity, \( d(1, e^{iH_{n}}) = \|H_{n}\| + O(\|H_{n}\|^{3}) \), is proportional to the eigenvalue spacing of \( H \) and, therefore, should obey the same Wigner-Dyson distribution. The conjecture above is indeed well supported.
by our numerical analysis, even for $n$ as small as 3 or 4 (see Fig. 1). One can show that the final error of $T_{L,n}^{l,m}$ also follows the Wigner-Dyson distribution (as illustrated in Fig. 4) with an average final distance $f \sim 60^{n/3}/\sqrt{n+1}$ times smaller than the average error of $T_0^{l,m}$, where the factor 60 is given by the order of the icosahedral group. With a smaller finite subgroup of SU(2), we would need a greater $n$ to achieve the same reduction.

Conclusions. In this paper we have demonstrated that the problem of compiling an arbitrary SU(2) qubit gate $T$ in terms of Fibonacci anyons can be solved efficiently by using hashing functions based on the 60 elements of the icosahedral group $I$ and their composition laws. Our procedure can be generalized to other anyonic models, different quantum computational schemes, and in principle to multi-qubit gates.

The hashing algorithm uses a light brute-force search up to $L = 24$ to initialize the 60 elements of $I$ with an average precision of about 0.02. The remaining search operations are based on the composition laws of the group $I$, which do not need any longer to exhaust the exponentially growing number of possibilities as $L$ increases. Indeed, it takes less than a second on a 3 GHz Intel E6850 processor to reach an average precision of $7.1 \times 10^{-4}$ (Fig. 4) for an arbitrary gate [18].

We can further improve the precision with additional iterations in the main processor, as we move exponentially down in error scales in a renormalization group fashion. For that we need longer braid representations of $I$, which must be obtained separately, e.g., by the brute-force search, and can be stored for all future uses. It follows that $q$ iterations reduce the average error by $f^q$ within a run time linear in $q$. To achieve an error smaller than a given $\varepsilon$, one needs $q \sim \log(1/\varepsilon)$ consecutive iterations. Therefore, the run time grows as $T \sim \log(1/\varepsilon)$, better than the poly-logarithmic time of the efficient implementation of the Solovay-Kitaev algorithm [20]. The iterative hashing algorithm generates a final braid of length $O(\log^2(1/\varepsilon))$, competing favorably with the results of other efficient quantum compiling algorithms [1, 20]. We hope that the quantum hashing algorithm, with potential improvements and hybridizations with other algorithms, introduces a new direction for efficient quantum compiling.

This work is supported by the grants INSTANS (from ESF), 2007HLPEZ (from MIUR), and the PCSIRT Project No. IRT0754 (from MoE, PRC). X.W. acknowledges the Max Planck Society and the Korea Ministry of Education, Science and Technology for the support of the Independent Junior Research Group at APCTP.

During the write-up of this Letter, we noticed a recent paper [21] which discusses a geometrical approach with binary polyhedral groups.

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press (2000), Chapter 4, and Appendix 3.
[2] A. Yu. Kitaev, A. H. Shen, and M. N. Vyalii, *Classical and Quantum Computation*, Am. Math. Soc. (2002), Section 8.
[3] A. Yu. Kitaev, Ann. Phys. 303, 2 (2003); *ibid.* 321, 2 (2006).
[4] M. Freedman, M. Larsen, and Z. Wang, Commun. Math. Phys. 227, 605 (2002); 228, 177 (2002); M. Freedman, A. Kitaev, and Z. Wang, *ibid.* 227, 587 (2002).
[5] C. Nayak et al., Rev. Mod. Phys. 80, 1083 (2008).
[6] G. K. Brennen and J. K. Pochos, Proc. R. Soc. A, 464, 1 (2008).
[7] J. Preskill, *Lecture Notes on Topological Quantum Computation*; available online at www.theory.caltech.edu/~preskill/ph219/topological.pdf.
[8] N. Read and E. H. Rezayi, Phys. Rev. B 59, 8084 (1999).
[9] J. S. Xia et al., Phys. Rev. Lett. 93, 176809 (2004).
[10] E. Ardonne and K. Schoutens, Phys. Rev. Lett. 82, 5096 (1999); E. Ardonne et al., Nucl. Phys. B 607, 549 (2001); E. Ardonne and K. Schoutens, Ann. Phys. 322, 201 (2007).
[11] N. E. Bonesteel et al., Phys. Rev. Lett. 95, 140503 (2005).
[12] S. H. Simon et al., Phys. Rev. Lett. 96, 070503 (2006).
[13] H. Xu and X. Wan, Phys. Rev. A 78, 042325 (2008).
[14] The distance $d$ (also referred to as error) between two gates (or their matrix representations) $U$ and $V$ is the operator norm distance $d(U,V) \equiv \|U-V\| = \sup_{\|\psi\|=1} \| (U-V) \psi \|$.
[15] Note $\max (d) = \sqrt{2}$ when $U \psi$ is orthogonal to $V \psi$.
[16] H. Xu and X. Wan, Phys. Rev. A 80, 012306 (2009).
[17] L. Hormozi et al., Phys. Rev. B 75, 165310 (2007).
[18] C. Rebbi, Phys. Rev. D 21, 3350 (1980); D. Petcher and D. H. Wögingarten, Phys. Rev. D 22, 2465 (1980); G. Bhanot, K. Bitar, and R. Salvador, Phys. Lett. B 188, 246 (1987).
[19] Object-oriented source codes available for download at http://sites.google.com/site/braidanyons/.
[20] M. Mehta, *Random Matrices*, 2nd ed. (Academic, San Diego, 1991).
[21] R. Mosseri, J. Phys. A: Math. Theor. 41, 175302 (2008).