

Research Directions and Foundations in Topological Quantum Computation Methods

Nicholas J. Daras¹

Abstract

We give a short overview of the recent research perspectives and mathematical foundations in topological quantum computation theory. In particular, we will be interested in braid representation theory, topological invariants of braids, approximation with braiding generators and quantum hashing with the icosahedral group.

Mathematics Subject Classification: 16T25; 20F36; 54H13; 57M25; 57M27; 81P68; 81P94; 94A60

Keywords: Quantum Cryptography; Quantum Computation; Invariants of Knots; Representation of Braid Groups; Fibonacci Braiding Generators; Quantum Hashing

¹ Department of Mathematics and Engineering Sciences, Faculty of Military Sciences, Hellenic Military Academy, Vari - 16673, Greece. E-mail: darasn@sse.gr

1 Introduction

A conventional computer uses bits, which are classical two-state systems. On the contrary, a quantum computer uses quantum two-state systems (called qubits). A quantum computer could efficiently decrypt many of the cryptographic systems in use today, including the prime integer factorization problem or the related discrete logarithm problem ([45]) and represent data to perform operations with polynomial speedup (including quantum database search, finding collisions in two-to-one functions and evaluating NAND trees). Frequently, quantum computers offer a more than polynomial speedup over the best known classical algorithm have been found for several problems, including simulation of quantum physical processes from chemistry and solid state physics, the approximation of Jones polynomial, and solving Pell's equation.

However, other existing cryptographic algorithms do not appear to be broken by these algorithms ([4]). For instance, some public-key algorithms are based on problems other than the integer factorization and discrete logarithm problems, like the McEliece and Niederreiter cryptosystems based on a problem in coding theory ([21]). Further, lattice-based cryptosystems are also not known to be broken by quantum computers, and finding a polynomial time algorithm for solving the dihedral hidden subgroup problem, which would break many lattice based cryptosystems, is a well-studied open problem ([31]).

The main problems in realizing a quantum computer are local errors, thermic noise and quantum decoherence. *Errors* are typically corrected in classical computers by keeping multiple copies of information and checking against these copies. With a quantum computer, however, the situation is more complex. If we measure a quantum state during an intermediate stage of a calculation to see if an error has occurred, we collapse the wave function and thus ruin the calculation. Remarkably, in spite of these difficulties, error correction is possible for quantum computers ([16]). One can represent information redundantly so that errors can be identified without measuring information. However, the error correction process

may itself be a little noisy. More errors could then occur during error correction, and the whole procedure will fail unless the basic error rate is very small ([3]). Random errors are also caused by the interaction between the quantum computer and the environment. As a result of this interaction, the quantum computer, which is initially in a pure superposition state, becomes entangled with its environment. This can cause observable errors. Since we cannot measure the state of the environment accurately, information is lost. In other words, the environment has caused *decoherence*. It was universally assumed until the advent of quantum error correction that quantum computation is intrinsically impossible since decoherence-induced quantum errors simply cannot be corrected in any real physical system ([44]). However, when error-correcting codes are used, the entanglement is transferred from the quantum computer to ancillary qubits so that the quantum information remains pure while the entropy is in the ancillary qubits.

One of the greatest challenges is controlling or removing quantum decoherence. This usually means isolating the system from its environment as interactions with the external world causes the system to decohere. This effect is irreversible and is usually something that should be highly controlled, if not avoided. Decoherence times for candidate systems, in particular the transverse relaxation time, typically range between nanoseconds and seconds at low temperature ([18]). These issues are more difficult for optical approaches as the timescales are orders of magnitude shorter and an often-cited approach to overcoming them is optical pulse shaping. Error rates are typically proportional to the ratio of operating time to decoherence time, hence any operation must be completed much more quickly than the decoherence time.

If the error rate is small enough, it is thought to be possible to use quantum error correction, which corrects errors due to decoherence, thereby allowing the total calculation time to be longer than the decoherence time. However, the use of error correction brings with it the cost of a greatly increased number of required qubits. The number required to factor integers using Shor's algorithm is still

polynomial, and thought to be between L and L^2 , where L is the number of bits in the number to be factored; error correction algorithms would inflate this figure by an additional factor of L ([19]).

A very different approach to the stability-decoherence problem is to create a topological quantum computer with anyons, quasi-particles used as threads and relying on braid theory to form stable logic gates ([22]).

*A topological quantum computer is a theoretical quantum computer that employs two-dimensional quasiparticles called anyons, whose world lines cross over one another to form **braids** in a three-dimensional spacetime (i.e., one temporal plus two spatial dimensions). These braids form the logic gates that make up the computer.*

The advantage of a quantum computer based on (quantum) braids over using trapped quantum particles is that the former is much more stable. When anyons are braided, the transformation of the quantum state of the system depends only on the topological class of the anyons' trajectories (which are classified according to the braid group): the smallest perturbations do not change the topological properties of the braids. This is like the effort required to cut a string and reattach the ends to form a different braid, as opposed to a ball (representing an ordinary quantum particle in four-dimensional spacetime) simply bumping into a wall. Thus, the quantum information which is stored in the state of the system is impervious to small errors in the trajectories.

While the elements of a topological quantum computer originate in a purely mathematical realm, recent experiments indicate these elements can be created in the real world using semiconductors made of gallium arsenide near absolute zero and subjected to strong magnetic fields.

Anyons form from the excitations in an electron gas in a very strong magnetic field, and carry fractional units of magnetic flux in a particle-like manner. This phenomenon is called the fractional quantum Hall effect. The electron "gas" is sandwiched between two flat plates of gallium arsenide, which

create the two-dimensional space required for anyons, and is cooled and subjected to intense transverse magnetic fields.

Topological quantum computers are equivalent in computational power to other standard models of quantum computation, in particular to the quantum circuit model and to the quantum Turing machine model. That is, any of these models can efficiently simulate any of the others. Nonetheless, certain algorithms may be a more natural fit to the topological quantum computer model. For example, algorithms for evaluating the Jones polynomial were first developed in the topological model, and only later converted and extended in the standard quantum circuit model.

In what follows, we will be interested in *give a short overview of the recent research perspectives and mathematical foundations in topological quantum computation theory.* In particular, we will be interested in *braid representation theory, topological invariants of braids, approximation with braiding generators and quantum hashing with the icosahedral group.*

2 Quantum Computation

2.1. Background

Suppose \mathcal{H} is a complex Hilbert space. We use Dirac's bra-ket notation as commonly used in quantum physics. This means vectors in \mathcal{H} are denoted as ket's

$$| \rangle.$$

Without loss of generality, one may assume that $\mathcal{H} \equiv \mathbb{C}^n$.

Definition 2.1 Let $|\varphi\rangle = (\varphi_1, \varphi_2, \dots, \varphi_n)^T \in \mathcal{H}$.

i. If $|\psi\rangle = (\psi_1, \psi_2, \dots, \psi_n)^T \in \mathcal{H}$, then the inner product of $|\varphi\rangle$ and $|\psi\rangle$ is defined by

$$\langle\varphi|\psi\rangle = |\varphi\rangle^T \overline{|\psi\rangle} = \sum_{j=1}^n \varphi_j \overline{\psi_j}.$$

ii. $\langle\varphi|$ is the linear functional $\mathcal{H} \rightarrow \mathbb{C}$ that maps every $|\psi\rangle \in \mathcal{H}$ to the inner product of $|\varphi\rangle$ and $|\psi\rangle$:

$$\langle\varphi|: \mathcal{H} \rightarrow \mathbb{C}: |\psi\rangle \mapsto \langle\varphi|\psi\rangle.$$

It follows that $\langle\varphi|$ can be thought as the transpose complex-conjugate row vector

$$\langle\varphi| = \overline{|\varphi\rangle}^T = (\overline{\varphi_1}, \overline{\varphi_2}, \dots, \overline{\varphi_n}).$$

iii. If $|\psi\rangle = (\psi_1, \psi_2, \dots, \psi_n)^T \in \mathcal{H}$, then the outer product of $|\varphi\rangle$ and $|\psi\rangle$ is defined by

$$|\varphi\rangle\langle\psi| \equiv |\varphi\rangle\overline{|\psi\rangle}^T = \underbrace{\begin{pmatrix} \varphi_1\overline{\psi_1} & \varphi_1\overline{\psi_2} & \cdots & \varphi_1\overline{\psi_n} \\ \varphi_2\overline{\psi_1} & \varphi_2\overline{\psi_2} & \cdots & \varphi_2\overline{\psi_n} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_n\overline{\psi_1} & \varphi_n\overline{\psi_2} & \cdots & \varphi_n\overline{\psi_n} \end{pmatrix}}_{n \times n \text{ matrix}},$$

iv. We also write $|\varphi\psi\rangle$ as a short hand for the tensor product of $|\varphi\rangle$ and $|\psi\rangle$. In other words, we have

$$|\varphi\rangle\otimes|\psi\rangle \equiv |\varphi\psi\rangle := (\varphi_1\psi_1, \varphi_1\psi_2, \dots, \varphi_1\psi_n, \dots, \varphi_n\psi_1, \varphi_n\psi_2, \dots, \varphi_n\psi_n)^T \in \mathcal{H}^n (\equiv \mathbb{C}^{n^2}).$$

2.2. The state vector of a quantum system

In quantum mechanics, a quantum system (: a theoretical or actual system based on quantum physics, as a supercomputer) is represented by a state vector $|\varphi\rangle$ in the state space \mathcal{H} . A quantum system with a state vector $|\varphi\rangle$ is called a pure state. However, it is also possible for a system to be in a statistical ensemble of different state vectors. For example, there may be a 50% probability that the state vector is $|\varphi_1\rangle$ and a 50% chance that the state vector $|\varphi_2\rangle$. This system would be in a mixed state. To distinguish pure and/or mixed states, one often uses the expressions 'coherent' and/or 'incoherent superposition' of quantum states.

States are "really" in the projective space associated with \mathcal{H} . More precisely, *the space of pure states of a quantum system is given by the one-dimensional subspaces of the corresponding Hilbert space (or the "points" of the projective Hilbert space). In a two-dimensional Hilbert space this is simply the complex projective line, which is a geometrical sphere.*

Given basis kets $|e_j\rangle$, any ket $|\varphi\rangle$ can be written

$$|\varphi\rangle = \sum_{j=1}^n c_j |e_j\rangle$$

where c_j are complex numbers. In physical terms, this is described by saying that $|\varphi\rangle$ has been expressed as a *quantum superposition* of the states $|e_j\rangle$. Recall that a quantum superposition of the states $|\varphi_j\rangle$ means that *if a quantum system has distinct states $|\varphi_1\rangle, |\varphi_2\rangle, \dots, |\varphi_k\rangle$, then it has infinitely many states of the form*

$$a_1|\varphi_1\rangle + a_2|\varphi_2\rangle + \dots + a_k|\varphi_k\rangle$$

where a_1, a_2, \dots, a_k are complex numbers taken up to a common multiple.

Remark 2.2 If the basis kets are chosen to be orthonormal (as is often the case), then

$$c_j = \langle e_j | \varphi \rangle \quad (j = 1, 2, \dots, n).$$

One property worth noting is that *the normalized states $|\varphi\rangle$ are characterized by*

$$\sum_{j=1}^n |c_j|^2 = 1.$$

If, in particular, $|e_j\rangle$ are eigenstates (with eigenvalues e_j) of an *observable*, and that observable is measured on the normalized state $|\varphi\rangle$, then the probability that the result of the measurement is $|e_j\rangle$ equals $|c_j|^2$. Recall that an observable is given by a finite collection $\{\Pi_j: j \in J = 1, 2, \dots, n\}$ of orthogonal projections ($\Pi_j^2 = \Pi_j$) $\Pi_j \in \text{End}(\mathcal{H})$ that satisfy the condition $\sum_{j=1}^n \Pi_j = \mathbb{I}$, where \mathbb{I} denotes the identity in $\text{End}(\mathcal{H})$. \square

Remark 2.3 (http://en.wikipedia.org/wiki/Density_matrix) A mixed state is different from a quantum superposition. An example of pure and mixed state is light polarization. Photons can have two helicities, corresponding to two

orthogonal quantum states, $|R\rangle$ (right circular polarization) and $|L\rangle$ (left circular polarization). A photon can also be in a superposition state, such as $(|R\rangle + |L\rangle)/\sqrt{2}$ (vertical polarization) or $(|R\rangle - |L\rangle)/\sqrt{2}$ (horizontal polarization). More generally, it can be in any state $a|R\rangle + b|L\rangle$, corresponding to linear, circular, or elliptical polarization. However, unpolarized light (such as the light from an incandescent light bulb) is different from any of these. Unlike linearly or elliptically polarized light, it passes through a polarizer with 50% intensity loss whatever the orientation of the polarizer; and unlike circularly polarized light, it cannot be made linearly polarized with any wave plate. Indeed, unpolarized light cannot be described as *any* state of the form $a|R\rangle + b|L\rangle$. However, unpolarized light *can* be described perfectly by assuming that each photon is either $|R\rangle$ with 50% probability or $|L\rangle$ with 50% probability. The same behavior would occur if each photon was either vertically polarized with 50% probability or horizontally polarized with 50% probability. Therefore, unpolarized light cannot be described by any pure state, but can be described as a statistical ensemble of pure states in at least two ways (the ensemble of half left and half right circularly polarized, or the ensemble of half vertically and half horizontally linearly polarized). These two ensembles are completely indistinguishable experimentally, and therefore they are considered the same mixed state. One of the advantages of the density matrix is that there is just one density matrix for each mixed state, whereas there are many statistical ensembles of pure states for each mixed state. Nevertheless, the density matrix contains all the information necessary to calculate any measurable property of the mixed state.

Where do mixed states come from? To answer that, consider how to generate unpolarized light. One way is to use a system in thermal equilibrium, a statistical mixture of enormous numbers of microstates, each with a certain probability (the Boltzmann factor), switching rapidly from one to the next due to thermal fluctuations. Thermal randomness explains why an incandescent light bulb, for example, emits unpolarized light. A second way to generate unpolarized

light is to introduce uncertainty in the preparation of the system, for example, passing it through a birefringent crystal with a rough surface, so that slightly different parts of the beam acquire different polarizations. A third way to generate unpolarized light uses an EPR setup: A radioactive decay can emit two photons traveling in opposite directions, in the quantum state $(|R, L\rangle + |L, R\rangle)/\sqrt{2}$. The two photons *together* are in a pure state, but if you only look at one of the photons and ignore the other, the photon behaves just like unpolarized light. More generally, mixed states commonly arise from a statistical mixture of the starting state (such as in thermal equilibrium), from uncertainty in the preparation procedure (such as slightly different paths that a photon can travel), or from looking at a subsystem entangled with something else. \square

2.3. The density matrix of a quantum system

The density matrix (see below) is especially useful for mixed states, because any state, pure or mixed, can be characterized by a single density matrix.

Definition 2.4 A state vector $|\varphi\rangle$ of \mathcal{H} can also be represented by a self-adjoint (or Hermitian) positive-semi definite ($: \operatorname{Re}(z^T \rho z) > 0$, whenever $z \in \mathcal{H} \equiv \mathbb{C}^n$) matrix

$$\rho = \begin{pmatrix} \rho_{1,1} & \cdots & \rho_{1,n} \\ \vdots & \vdots & \vdots \\ \rho_{n,1} & \cdots & \rho_{n,n} \end{pmatrix} \in \operatorname{End}(\mathcal{H})$$

of trace one ($\Leftrightarrow \operatorname{tr}(\rho) := \sum_{j=1}^n \rho_{j,j} = 1$), that describes the statistical state of a quantum state. Such an operator is called a density matrix (or density operator). We denote by $\mathcal{D}(\mathcal{H})$ the set of all density matrices $\rho \in \operatorname{End}(\mathcal{H})$, and we write $\rho \geq 0$ to express that the operator ρ is positive semi-definite.

Remark 2.5 Any density matrix can be written as

$$\rho = \sum_j p_j |\varphi_j\rangle\langle\varphi_j|$$

where p_j is the fraction of the ensemble in each pure state $|\varphi_j\rangle$. In particular, *a state vector is pure if and only if there exists $|\varphi\rangle \in \mathcal{H}$ such that*

$$\rho = |\varphi\rangle\langle\varphi|,$$

where the trace condition on ρ implies that $|\varphi\rangle$ is normalized, i.e. $\|\varphi\|^2 = \langle\varphi|\varphi\rangle = 1$. Equivalently, *a state vector is pure if its density matrix $\rho \in \mathcal{D}(\mathcal{H})$ has rank 1, which is equivalent to saying that $\rho \in \mathcal{D}(\mathcal{H})$ satisfies*

$$\rho^2 = \rho,$$

i.e. the state is idempotent. \square

Remark 2.6 From a geometric point of view, *the pure states are given by the extremal points of the convex set $\mathcal{D}(\mathcal{H})$, in particular, any $\rho \in \mathcal{D}(\mathcal{H})$ can be written as a convex linear combination*

$$\rho = \sum_{j=1}^K p_j |\varphi_j\rangle\langle\varphi_j| \quad (p_1, p_2, \dots, p_K \geq 0 \text{ and } \sum_{j=1}^K p_j = 1)$$

of pure states. Such a system can alternatively be understood to be in pure state $|\varphi_j\rangle$ with probability p_j . \square

To simplify the language, *we will sometimes be somewhat sloppy in distinguishing between a quantum system, its state, and the density matrix or state vector describing the state.*

2.4. Measuring a quantum system

The only way to gain information on the state of a quantum system is by means of a measurement. Given a state $|\varphi\rangle = \sum_k a_k |\varphi_k\rangle$ in \mathcal{H} , a measurement returns $|\varphi_k\rangle$ with probability $|\langle\varphi|\varphi_k\rangle|^2 / \langle\varphi|\varphi\rangle$. This model of measurement is a simple instance of the situation with a quantum mechanical system that is in a mixed state until it is observed. The result of observation is to put the system into one of the basis states.

A measurement is described by an observable, which is given by a finite collection $\{\Pi_i: i \in I = 1, 2, \dots, m\}$ of orthogonal projections ($\Pi_i^2 = \Pi_i$) $\Pi_i \in \mathcal{E}nd(\mathcal{H})$ that satisfy the condition $\sum_{i=1}^m \Pi_i = \mathbb{I}$, where \mathbb{I} denotes the identity in $\mathcal{E}nd(\mathcal{H})$. For a quantum system represented by a state vector $|\varphi\rangle \in \mathcal{H}$ and a density matrix ρ , measuring the system with respect to the observable $\{\Pi_i: i \in I\}$ has the following two effects.

- 1) An outcome $i \in I$ is observed, with the probability that a specific $i \in I$ is observed given by $p_i = \text{tr}(\Pi_i \rho)$.
- 2) After the measurement, the state with density matrix ρ has *collapsed* to a state with density matrix $\rho' = \Pi_i \rho \Pi_i^* / p_i$ where the outcome observed is i and Π_i^* is the adjoint (conjugate transpose) matrix $(\overline{\Pi_i})^T$ of Π_i .

We often consider measurements where the Π_j 's are projections onto a *basis* $\{|e_j\rangle: j \in J = 1, 2, \dots, n\}$ of \mathcal{H} . In this case, we say that the state (vector) $|\varphi\rangle$ is measured *in basis* $\{|e_j\rangle: j \in J\}$. Measurement in basis returns basis elements $|e_j\rangle$ of \mathcal{H} with probability $|\langle\varphi|e_j\rangle|^2 / \langle e_j|e_j\rangle$. If, in particular, $|e_j\rangle$ lies in an orthonormal basis of \mathcal{H} , then the observable becomes $\Pi_j = |e_j\rangle\langle e_j|$ and the measurement of $|\varphi\rangle$ in basis $\{|e_j\rangle: j \in J\}$ returns coordinates of $|\varphi\rangle$ with corresponding probabilities $|\langle\varphi|e_j\rangle|^2$.

2.4.1. Quantum computers

As is common in quantum information processing, we consider the quantum state of a system to be *static*, meaning that it does not change over time, unless it is actively operated on.

Definition 2.7.i *A quantum system $|\varphi\rangle$ in the state space \mathcal{H} can be operated on by means of applying a unitary transformation $U \in \mathcal{E}nd(\mathcal{H})$ ($\mathcal{E}nd(\mathcal{H})$ represents the set of \mathcal{H} 's endomorphisms). As a result, the density matrix $\rho \in \mathcal{D}(\mathcal{H})$ describing $|\varphi\rangle$ evolves to a new density matrix $\rho' = U\rho U^\dagger$*

representing the new state. In case of a pure state described by its state vector $|\varphi\rangle \in \mathcal{H}$, the state evolves as

$$|\varphi'\rangle = U|\varphi\rangle = |U\varphi\rangle.$$

ii. A quantum process (or even more a physical process) occurs in k steps

$$U_j: \mathcal{D}(\mathcal{H}) \rightarrow \mathcal{D}(\mathcal{H}): \rho \mapsto U_j \rho U_j^\dagger \quad (j = 1, 2, \dots, k),$$

where U_j is a unitary linear transformation. Note that since U_j is unitary, it follows that probability is preserved in the course of a quantum process.

iii. According to the Kraus quantum process representation, the most general quantum evolution process of a single-qubit density matrix is given by

$$\rho' = \sum_k U_k \rho U_k^\dagger, \text{ with } \sum_k U_k U_k^\dagger = \mathbb{I} \quad (: \text{ the identity matrix}).$$

iv. The initial state of a quantum process is a vector $|\varphi_0\rangle$ in the complex vector space \mathcal{H} .

v. A quantum computer is, abstractly, a composition of unitary transformations, together with an initial state and a choice of measurement basis. One runs the computer by repeatedly initializing it, and then measuring the result of applying the unitary transformation U to the initial state. The results of these measurements are then analyzed for the desired information that the computer was set to determine. The key to using the computer is the design of the initial state and the design of the composition of unitary transformations. For more specific examples of quantum algorithms, the reader should consult the reference book "Quantum Computation and Quantum Information" by M. A. Nielsen and I. L. Chuang ([39]).

Remark 2.8 One of the details required for any specific quantum problem is the nature of the unitary evolution. This is specified by knowing appropriate information about the classical physics that supports the phenomena. This information is used to choose an appropriate Hamiltonian through which the unitary operator is constructed via a correspondence principle that replaces classical variables with appropriate quantum operators. \square

2.4.2. The two-dimensional case: qubits and Bloch sphere

Definition 2.9 A qubit (or quantum bit) is a quantum system with state space $\mathcal{H} = \mathbb{C}^2$.

Notation 2.10 $\{|0\rangle, |1\rangle\}$ denotes the (orthonormal) computational basis:

$$|0\rangle = (1, 0)^T \in \mathcal{H} = \mathbb{C}^2 \text{ and } |1\rangle = (0, 1)^T \in \mathcal{H} = \mathbb{C}^2$$

and $\{|+\rangle, |-\rangle\}$ the Hadamard basis

$$|+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \in \mathcal{H} = \mathbb{C}^2 \text{ and}$$

$$|-\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \in \mathcal{H} = \mathbb{C}^2. \quad \square$$

Remark 2.11 Note that

$$|+\rangle = H|0\rangle \text{ and } |-\rangle = H|1\rangle$$

where H is the Hadamard transform

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

Thus, $H^b\{|0\rangle, |1\rangle\} = \{H^b|0\rangle, H^b|1\rangle\}$ denotes the computational basis if $b = 0$ and the Hadamard basis if $b = 1$. \square

Obviously, every qubit can be written as a unique linear combination of the computational basis elements. In particular, *the physical state of a qubit* $|\varphi\rangle$ is *the superposition* $|\varphi\rangle = a|0\rangle + b|1\rangle$ ($a \in \mathbb{C}, b \in \mathbb{C}$). According to the Remark 1, when we try to measure the qubit in this basis in order to determine its state, we get either $|0\rangle$ with probability $|a|^2$ or $|1\rangle$ with probability $|b|^2$. Since $|a|^2 + |b|^2 = 1$, *the qubit is a unit vector in the aforementioned two-dimensional Hilbert space.*

Definition 2.12 *The Bloch sphere is the 2-sphere, with each pair of antipodal points corresponding to mutually orthogonal state vectors. The north and south poles of the Bloch sphere are typically chosen to correspond to the standard computational basis vectors $\{|0\rangle, |1\rangle\}$, respectively, which in turn might*

correspond e.g. to the spin-up and spin-down states of an electron. (This choice is arbitrary, however.)

Given an orthonormal basis, any pure state $|\varphi\rangle$ of a two-level quantum system can be written as a complex superposition of the computational basis vectors $|0\rangle$ and $|1\rangle$. Since global phase factors do not have any physical meaning, we can take the coefficient of $|0\rangle$ to be real and non-negative.

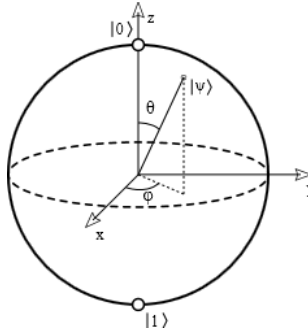


Figure 1: The Bloch sphere

Thus $|\varphi\rangle$ has the normalized representation

$$|\varphi\rangle = \cos(\theta/2)|0\rangle + e^{i\omega} \sin(\theta/2)|1\rangle$$

with $0 \leq \theta \leq \pi$ and $0 \leq \omega \leq 2\pi$. Except in the case where $|\varphi\rangle$ is one of the ket vectors $|0\rangle$ or $|1\rangle$ the representation is unique. The parameters θ and ω , re-interpreted as spherical coordinates, specify a point

$$a = \vec{a} = \begin{pmatrix} \sin\theta \cos\omega \\ \sin\theta \sin\omega \\ \cos\theta \end{pmatrix}$$

on the unit sphere in \mathbb{R}^3 (:the three-dimensional space embedding the Bloch sphere).

As we shall see below, a simple general criterion for checking whether a quantum state is pure or mixed is that the von Neumann quantum state entropy is 0 for a pure state, and strictly positive for a mixed state. When $\mathcal{H} = \mathbb{C}^2$, another, equivalent, criterion for checking whether a two-dimensional density matrix ρ is describing a pure or mixed state is that the trace $\text{tr}(\rho)$ of ρ^2 is equal

to 1 if the state is pure, and less than 1 if the state is mixed. Indeed, we have the following ‘‘strong’’ geometrical representations for two-dimensional density matrices.

Theorem 2.13.i Any two-dimensional density matrix ρ can be written as follows:

$$\rho = \frac{1}{2} \begin{pmatrix} 1 + x_3 & x_1 - ix_2 \\ x_1 + ix_2 & 1 - x_3 \end{pmatrix} \quad (x_1, x_2, x_3 \in \mathbb{R}).$$

ii. As density operators must be positive semidefinite, we have

$$x_1^2 + x_2^2 + x_3^2 \leq 1.$$

iii. Equivalently, any two-dimensional density ρ can be expanded using the identity

$$\rho = \frac{1}{2} [\mathbb{I} + \vec{X}\sigma] = \frac{1}{2} \left[\mathbb{I} + (x_1, x_2, x_3) \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{pmatrix} \right].$$

Here, we have used the notation $\vec{X} = (x_1, x_2, x_3)$ and $\sigma = (\sigma_1, \sigma_2, \sigma_3)^T$, where σ_1, σ_2 and σ_3 are the following three 2×2 Hermitian, unitary Pauli matrices:

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

iv. The eigenvalues of the density matrix ρ are given by

$$\lambda = \lambda_{1,2} = \frac{1}{2} \left(1 \pm \sqrt{x_1^2 + x_2^2 + x_3^2} \right).$$

v. For pure states, we must have

$$\text{tr}(\rho^2) = \frac{1}{2} [1 + |\vec{X}|] \Leftrightarrow |\vec{X}| := \sqrt{x_1^2 + x_2^2 + x_3^2} = 1.$$

As a corollary, the points on the surface of the Bloch sphere correspond to the pure states of the system, whereas the interior points correspond to the mixed states.

2.4.3. The n-qubit systems

As it is already pointed out, for any two qubits

$$|\varphi\rangle = (\varphi_1, \varphi_2)^T \in \mathbb{C}^2 \quad \text{and} \quad |\psi\rangle = (\psi_1, \psi_2)^T \in \mathbb{C}^2.$$

the tensor product of $|\varphi\rangle$ and $|\psi\rangle$ is given by

$$|\varphi\rangle \otimes |\psi\rangle \equiv |\varphi\psi\rangle := (\varphi_1\psi_1, \varphi_1\psi_2, \varphi_2\psi_1, \varphi_2\psi_2)^T \in \mathbb{C}^4.$$

We say that $|\varphi\rangle \otimes |\psi\rangle \equiv |\varphi\psi\rangle$ is a 2-qubit.

Example 2.14 In particular, if $|0\rangle = (1,0)^T \in \mathbb{C}^2$ and $|1\rangle = (0,1)^T \in \mathbb{C}^2$, we have

$$\begin{aligned} |00\rangle &= (1,0,0,0)^T \in \mathbb{C}^4, |01\rangle = (0,1,0,0)^T \in \mathbb{C}^4, \\ |10\rangle &= (0,0,1,0)^T \in \mathbb{C}^4, |11\rangle = (0,0,0,1)^T \in \mathbb{C}^4. \quad \square \end{aligned}$$

More generally, we have the following.

Definition 2.15 A n -qubit system consists of n qubits, i.e., is a quantum system whose state space is the n -fold tensor product

$$(\mathbb{C}^2)^{\otimes n} = \underbrace{\mathbb{C}^2 \otimes \dots \otimes \mathbb{C}^2}_{n\text{-times}}.$$

Formally, we can write

$$\begin{aligned} (\mathbb{C}^2)^{\otimes n} &= \{|\varphi\rangle \otimes |\psi\rangle \otimes \dots \otimes |\chi\rangle \equiv |\varphi\psi \dots \chi\rangle \in \mathbb{C}^{2^n} : \\ &|\varphi\rangle = (\varphi_1, \varphi_2)^T \in \mathbb{C}^2, |\psi\rangle = (\psi_1, \psi_2)^T \in \mathbb{C}^2, \dots \\ &\dots, |\chi\rangle = (\chi_1, \chi_2)^T \in \mathbb{C}^2\}. \end{aligned}$$

Example 2.16 It is easily seen

$$\begin{aligned} \mathbb{C}^2 \otimes \mathbb{C}^2 &= \{|\varphi\rangle \otimes |\psi\rangle \equiv |\varphi\psi\rangle := (\varphi_1\psi_1, \varphi_1\psi_2, \varphi_2\psi_1, \varphi_2\psi_2)^T \in \mathbb{C}^4: \\ &|\varphi\rangle = (\varphi_1, \varphi_2)^T \in \mathbb{C}^2 \text{ and } |\psi\rangle = (\psi_1, \psi_2)^T \in \mathbb{C}^2\} \end{aligned}$$

and

$$\begin{aligned} \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 &= \\ &\{|\varphi\rangle \otimes |\psi\rangle \otimes |\chi\rangle \equiv |\varphi\psi\chi\rangle := \\ &(\varphi_1\psi_1\chi_1, \varphi_1\psi_1\chi_2, \varphi_1\psi_2\chi_1, \varphi_1\psi_2\chi_2, \varphi_2\psi_1\chi_1, \varphi_2\psi_1\chi_2, \varphi_2\psi_2\chi_1, \varphi_2\psi_2\chi_2)^T \in \mathbb{C}^8: \\ &|\varphi\rangle = (\varphi_1, \varphi_2)^T \in \mathbb{C}^2, |\psi\rangle = (\psi_1, \psi_2)^T \in \mathbb{C}^2 \text{ and } |\chi\rangle = (\chi_1, \chi_2)^T \in \mathbb{C}^2\}. \end{aligned}$$

In particular, we have

$$\begin{aligned} |000\rangle &= (1,0,0,0,0,0,0,0)^T \in \mathbb{C}^8, |001\rangle = (0,1,0,0,0,0,0,0)^T \in \mathbb{C}^8, \\ |010\rangle &= (0,0,1,0,0,0,0,0)^T \in \mathbb{C}^8, |011\rangle = (0,0,0,1,0,0,0,0)^T \in \mathbb{C}^8, \\ |100\rangle &= (0,0,0,0,1,0,0,0)^T \in \mathbb{C}^8, |101\rangle = (0,0,0,0,0,1,0,0)^T \in \mathbb{C}^8, \\ |110\rangle &= (0,0,0,0,0,0,1,0)^T \in \mathbb{C}^8, |111\rangle = (0,0,0,0,0,0,0,1)^T \in \mathbb{C}^8. \quad \square \end{aligned}$$

2.4.4. Entanglement

The true separation from the classical regime is most apparent when we analyse multiple qubits. Classically, the state of a non-random system of n bits can be expressed in entirety by specifying the individual states of the component bits. A quantum system of n qubits can be expressed as a vector in a 2^n -dimensional Hilbert space. As we will see shortly, it will not always be possible to fully specify the individual qubits.

Assume that we have two qubits with pure states $|\varphi_1\rangle = a_1|0\rangle + b_1|1\rangle$ and $|\varphi_2\rangle = a_2|0\rangle + b_2|1\rangle$ respectively. The vector representing the total system, or joint state, is given by the tensor product of the individual qubits. The tensor product of the given qubits is

$$\begin{aligned} |\varphi_1\rangle \otimes |\varphi_2\rangle &= (a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle) = \\ &= a_1a_2|0\rangle \otimes |0\rangle + a_1b_2|0\rangle \otimes |1\rangle + b_1a_2|1\rangle \otimes |0\rangle + b_1b_2|1\rangle \otimes |1\rangle \\ &= a_1a_2|00\rangle + a_1b_2|01\rangle + b_1a_2|10\rangle + b_1b_2|11\rangle. \end{aligned}$$

Conversely, assume that we have a two-qubit system in the state

$$(1/\sqrt{2})|00\rangle + (1/\sqrt{2})|11\rangle.$$

Correspondence with the last equation above implies that

$$a_1a_2 = 1/\sqrt{2}, a_1b_2 = b_1a_2 = 0, \text{ and } b_1b_2 = 1/\sqrt{2}.$$

However, there exists no assignment to these parameters satisfying all conditions. From this we conclude that the two qubits sharing this state are entangled, meaning they cannot be expressed individually. If there exists a valid decomposition into individual qubits, we call the joint state a product state.

Entangled quantum states provide possibilities unavailable to classical computers. Specifically, using the Bell inequalities, we can verify that entanglement provides higher levels of correlation than anything possible in the classical regime. Subsequently, it is a feature that manifests in virtually all important quantum algorithms.

Entanglement can exist in varying degrees. States with the maximum amount of entanglement are called maximally entangled. Perhaps the most common maximally entangled two-qubit states are the four Bell states:

$$\begin{aligned} |\Phi^-\rangle &= (1/\sqrt{2})|00\rangle - (1/\sqrt{2})|11\rangle \\ |\Psi^+\rangle &= (1/\sqrt{2})|01\rangle + (1/\sqrt{2})|10\rangle \\ |\Psi^-\rangle &= -(1/\sqrt{2})|01\rangle + (1/\sqrt{2})|10\rangle \\ |\Phi^+\rangle &= (1/\sqrt{2})|00\rangle + (1/\sqrt{2})|11\rangle. \end{aligned}$$

The Bell states are mutually orthogonal, forming a basis for the 4 –dimensional Hilbert space.

Remark 2.17 A two-qubit pure state $|\varphi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$ is entangled exactly when $(ad - bc) \neq 0$. It is easy to use this fact to check when a specific matrix is, or is not, entangling. \square

2.4.5. Quantum gates

Definition 2.18 A *quantum gate* (or *quantum logic gate*) is a basic model for quantum computation (: *quantum circuit*) operating on a small number of qubits. A *quantum circuit* is a model for quantum computation in which a computation is a sequence of quantum gates, which are reversible transformations on a quantum mechanical analog of an n -bit register. This analogous structure is referred to as an n -qubit register.

Remark 2.19 They are the building blocks of quantum circuits, like classical logic gates are for conventional digital circuits. \square

Unlike many classical logic gates, quantum logic gates are reversible. *Reversible computing* is a model of computing where the computational process to some extent is reversible, i.e., time-invertible. A necessary condition for reversibility of a computational model is that the transition function mapping states to their successors at a given later time should be one-to-one. Reversible

computing is generally considered an unconventional form of computing. There are two major, closely related, types of reversibility that are of particular interest for this purpose: *physical reversibility* and *logical reversibility*.

However, classical computing can be performed using only reversible gates. For example, the reversible Toffoli gate can implement all Boolean functions. This gate has a direct quantum equivalent, showing that *quantum circuits can perform all operations performed by classical circuits*.

Quantum logic gates are represented by unitary matrices. A unitary matrix is a (square) $n \times n$ complex matrix U satisfying the condition $U^\dagger U = U U^\dagger = \mathbb{I}_n$ where \mathbb{I}_n is the identity matrix in n dimensions and U^\dagger is the conjugate transpose (also called the Hermitian adjoint) of U .

The most common quantum gates operate on spaces of one or two qubits, just like the common classical logic gates operate on one or two bits. This means that as matrices, quantum gates can be described by 2×2 or 4×4 unitary matrices.

2.4.5.i. Commonly used gates

Quantum gates are usually represented as matrices. A gate which acts on k qubits is represented by a $2^k \times 2^k$ unitary matrix. The number of qubits in the input and output of the gate have to be equal. The action of the quantum gate is found by multiplying the matrix representing the gate with the vector which represents the quantum state.

- **Hadamard gate**

The Hadamard gate acts on a single qubit and represents a rotation of π about the x – and z – axes. It is represented by the Hadamard matrix (Hadamard transform):

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

Since the rows of the matrix are orthogonal, H is indeed a unitary matrix. The Hadamard gate maps the basis state $|0\rangle$ to $|+\rangle$ and $|1\rangle$ to $|-\rangle$, where $\{|+\rangle, |-\rangle\}$ is the Hadamard basis of \mathbb{C}^2 :

$$|+\rangle = (1/\sqrt{2})(1,1)^T = (1/\sqrt{2})(|0\rangle + |1\rangle) \in \mathcal{H} = \mathbb{C}^2 \text{ and}$$

$$|-\rangle = (1/\sqrt{2})(1,-1) = (1/\sqrt{2})(|0\rangle - |1\rangle) \in \mathcal{H} = \mathbb{C}^2.$$

Recall that $|+\rangle = H|0\rangle$ and $|-\rangle = H|1\rangle$.

- **Pauli-X gate**

The Pauli-X gate acts on a single qubit. It is the quantum equivalent of a NOT gate. It equates to a rotation of the Bloch Sphere around the x –axis by π radians. It maps $|0\rangle$ to $|1\rangle$ and $|1\rangle$ to $|0\rangle$. It is represented by the Pauli \mathcal{X} matrix:

$$\sigma_1 = \sigma_x := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

- **Pauli-Y gate**

The Pauli-Y gate acts on a single qubit. It equates to a rotation around the y –axis of the Bloch Sphere by π radians. It maps $|0\rangle$ to $i|1\rangle$ and $|1\rangle$ to $-i|0\rangle$. It is represented by the Pauli \mathcal{Y} matrix:

$$\sigma_2 = \sigma_y := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}.$$

- **Pauli-Z gate**

The Pauli-Z gate acts on a single qubit. It equates to a rotation around the z –axis of the Bloch Sphere by π radians. Thus, it is a special case of a phase shift gate (see below) with $\theta = \pi$. It leaves the basis state $|0\rangle$ unchanged and maps $|1\rangle$ to $-|1\rangle$. It is represented by the Pauli \mathcal{Z} matrix:

$$\sigma_3 = \sigma_z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

- **Phase shift gates**

This is a family of single-qubit gates that leave the basis state $|0\rangle$ unchanged and maps $|1\rangle$ to $e^{i\theta}|1\rangle$. The probability of measuring a $|0\rangle$ or $|1\rangle$ is unchanged after applying this gate; however it modifies the phase of the quantum state. This

is equivalent to tracing a horizontal circle (a line of latitude) on the Bloch Sphere by θ radians:

$$R_{\theta} := \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$$

where θ is the *phase shift*. Some common examples are the $(\pi/8)$ gate where $\theta = (\pi/4)$, the phase gate where $\theta = (\pi/2)$ and the Pauli-Z gate where $\theta = \pi$.

- **Swap gate**

The swap gate swaps two qubits. It is represented by the matrix:

$$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

- **Controlled gates**

Controlled gates act on 2 or more qubits, where one or more qubits act as a control for some operation. For example, the controlled NOT gate (or CNOT) acts on 2 qubits, and performs the NOT operation on the second qubit only when the first qubit is $|1\rangle$, and otherwise leaves it unchanged. It is represented by the matrix

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

On the standard basis $\{|0\rangle, |1\rangle\}$, *CNOT* is the identity when the first qubit is $|0\rangle$, and it flips the second qubit, leaving the first alone, when the first qubit is $|1\rangle$.

More generally if U is a gate that operates on single qubits with matrix representation

$$U := \begin{pmatrix} x_{0,0} & x_{0,1} \\ x_{1,0} & x_{1,1} \end{pmatrix},$$

then the *controlled-U gate* is a gate that operates on two (2) qubits in such a way that the first qubit serves as a control. It maps the basis states as follows.

$$|00\rangle \mapsto |00\rangle,$$

$$|01\rangle \mapsto |01\rangle,$$

$$|10\rangle \mapsto |1\rangle U|0\rangle = |1\rangle(x_{0,0}|0\rangle + x_{1,0}|1\rangle) \text{ and}$$

$$|11\rangle \mapsto |1\rangle U|1\rangle = |1\rangle(x_{0,1}|0\rangle + x_{1,1}|1\rangle).$$

The matrix representing the controlled- U gate is

$$C(U) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & x_{0,0} & x_{0,1} \\ 0 & 0 & x_{1,0} & x_{1,1} \end{pmatrix}.$$

When U is one of the Pauli matrices σ_x , σ_y , or σ_z , the respective terms "controlled- X ", "controlled- Y ", or "controlled- Z " are sometimes used.

- ***Toffoli gate***

The Toffoli gate, also CCNOT gate, is a 3-bit gate, which is universal for classical computation. The quantum Toffoli gate is the same gate, defined for 3 qubits. If the first two bits are in the state $|1\rangle$, it applies a Pauli- X on the third bit, else it does nothing. It is an example of a controlled gate. Since it is the quantum analog of a classical gate, it is completely specified by its truth table.

INPUT	OUTPUT	INPUT	OUTPUT
0 0 0	0 0 0	1 0 0	1 0 0
0 0 1	0 0 1	1 0 1	1 0 1
0 1 0	0 1 0	1 1 0	1 1 1
0 1 1	0 1 1	1 1 1	1 1 0

It can be also described as the gate which maps $|a, b, c\rangle$ to $|a, b, c \oplus ab\rangle$.

- ***Fredkin gate***

The Fredkin gate (also CSWAP gate) is a 3-bit gate that performs a controlled swap. It is universal for classical computation (see below section I.1.h.ii). It has the useful property that the numbers of 0s and 1s are conserved throughout, which in the billiard ball model means the same number of balls are output as input. This

corresponds nicely to the conservation of mass in physics, and helps to show that the model is not wasteful.

INPUT			OUTPUT			INPUT			OUTPUT		
C	I_1	I_2	C	O_1	O_2	C	I_1	I_2	C	O_1	O_2
0	0	0	0	0	0	1	0	0	1	0	0
0	0	1	0	0	1	1	0	1	1	1	0
0	1	0	0	1	0	1	1	0	1	0	1
0	1	1	0	1	1	1	1	1	1	1	1

It can be also described as the gate which maps $|a, b, c\rangle$ to $|a, \bar{a}b + ac, \bar{a}c + ab\rangle$.

2.4.5.ii. Universal quantum gates

Informally,

Definition 2.20 *Let V be a two complex dimensional vector space.*

- i.** *We say that the gate $G: V \otimes V \rightarrow V \otimes V$ is universal for quantum computation (or just universal) if G together with local unitary transformations (unitary transformations from V to V) generates all unitary transformations of the complex vector space $V \otimes V$ of dimension 2^n to itself.*
- ii.** *A set of universal quantum gates is any set of gates to which any operation possible on a quantum computer can be reduced, that is, any other unitary operation can be expressed as a finite sequence of gates from the set.*

Technically, this is impossible since the number of possible quantum gates is uncountable, whereas the number of finite sequences from a finite set is countable. To solve this problem, we only require that any quantum operation can be approximated by a sequence of gates from this finite set. Moreover, for the specific case of single qubit unitaries the Solovay-Kitaev theorem guarantees that this can be done efficiently.

Remark 2.21 It is well known ([39]) that *CNOT* is a universal gate. \square

Theorem 2.22 One simple set of two-qubit universal quantum gates is

- i. the Hadamard gate H ,
- ii. the $(\pi/8)$ gate with phase shift $\theta = (\pi/4)$,
- iii. the controlled NOT gate.

Proposition 2.23 A single-gate set of universal quantum gates can also be formulated using the three-qubit Deutsch gate $D(\theta)$, which performs the transformation

$$|a, b, c\rangle \mapsto \begin{cases} i \cos(\theta)|a, b, c\rangle + \sin(\theta)|a, b, 1 - c\rangle, & \text{for } a = b = 1 \\ |a, b, c\rangle, & \text{otherwise.} \end{cases}$$

The universal classical logic gate, the Toffoli gate, is reducible to the Deutsch gate $D(\pi/2)$, thus showing that *all classical logic operations can be performed on a universal quantum computer.*

2.4.6. Quantum computations

So far we have not shown how quantum circuits are used to perform computations. Since many important numerical problems reduce to computing a unitary transformation \mathbf{U} on a finite dimensional space (the celebrated discrete *Fourier transform* being a prime example) one might expect that some quantum circuit could be designed to carry out the transformation U .

The sequence of N complex numbers x_0, \dots, x_{N-1} is transformed into another sequence of N complex numbers according to the *DFT formula*

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i 2\pi(k/N)n}.$$

The transform is sometimes denoted by the symbol \mathcal{F} , as in $X = \mathcal{F}\{x\}$ or $\mathcal{F}(x)$ or $\mathcal{F}x$. As a linear transformation on a finite-dimensional vector space, the DFT expression can also be written in terms of a DFT matrix; when scaled appropriately it becomes a unitary matrix and the X_k can thus be viewed as

coefficients of x in an orthonormal basis. The inverse discrete Fourier transform (IDFT) is given by

$$x_n = (1/N) \sum_{k=0}^{N-1} X_k e^{i 2\pi(k/N)n}.$$

These formulas can be interpreted or derived in various ways; for example, they can be interpreted as arising from the discrete-time Fourier transform (DTFT) and its inverse when applied to a periodic sequence. (Given a discrete set of real or complex numbers $x[n]$, $n \in \mathbb{Z}$ (integers), the *discrete-time Fourier transform* (or *DTFT*) of $x[n]$ is usually written

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-i \omega n}.)$$

In principle, one needs only to prepare a n qubit state ψ as an appropriate superposition of computational basis states for the input and measure the output $U\psi$. Unfortunately, there are two problems with this:

- One cannot measure the phase of ψ at any computational basis state so there is no way of reading out the complete answer. This is in the nature of measurement in quantum mechanics.
- There is no way to efficiently prepare the input state ψ .

This does not prevent quantum circuits for the discrete Fourier transform from being used as intermediate steps in other quantum circuits, but the use is more subtle. In fact quantum computations are *probabilistic*.

We now provide a mathematical model for how quantum circuits can simulate *probabilistic* but classical computations.

- a) Consider an r –qubit circuit U with register space $H_{QB(r)}$.
- b) U is thus a unitary map $H_{QB(r)} \rightarrow H_{QB(r)}$.
- c) In order to associate this circuit to a classical mapping on bitstrings, we specify
 - an *input register* $X = \{0,1\}^m$ of m (classical) bits
 - an *output register* $Y = \{0,1\}^n$ of n (classical) bits.
- d) The contents $x = x_1, \dots, x_m$ of the classical input register are used to initialize the qubit register in some way. Ideally, this would be done with the

computational basis state $|x, 0\rangle = |x_1, \dots, x_m, 0, \dots, 0\rangle$ where there are $(r - m)$ zeroed inputs.

Nevertheless, this perfect initialization is completely unrealistic. Let us assume therefore that the initialization is a mixed state given by some density operator S which is near the idealized input in some appropriate metric, e.g. $\text{tr}(|x, 0\rangle\langle x, 0| - S|) \leq \delta$.

- e) Similarly, the output register space is related to the qubit register, by a Y valued observable A . Note that observables in quantum mechanics are usually defined in terms of *projection valued measures* on \mathbb{R} ; if the variable happens to be discrete, the projection valued measure reduces to a family $\{E_\lambda\}$ indexed on some parameter λ ranging over a countable set. Similarly, a Y valued observable, can be associated with a family of pairwise orthogonal projections $\{E_y\}$ indexed by elements of Y such that

$$I = \sum_{y \in Y} E_y.$$

Given a mixed state S , there corresponds a probability measure on Y given by $\text{Pr}(y) = \text{tr}(SE_y)$.

- f) The function $F: X \rightarrow Y$ is computed by a circuit $U: H_{QB(r)} \rightarrow H_{QB(r)}$ to within ε if and only if for all bitstrings x of length m

$$\langle x, 0 | U^* E_{F(x)} U | x, 0 \rangle = \langle E_{F(x)}, U(|x, 0\rangle) | U(|x, 0\rangle) \rangle \geq 1 - \varepsilon.$$

- g) Now

$$\begin{aligned} |\text{tr}(SU^* E_{F(x)} U) - \langle x, 0 | U^* E_{F(x)} U | x, 0 \rangle| &\leq \\ \text{tr}(|x, 0\rangle\langle x, 0| - S|) \|U^* E_{F(x)} U\| &\leq \delta \end{aligned}$$

so that

$$\text{tr}(SU^* E_{F(x)} U) \geq 1 - \varepsilon - \delta.$$

Theorem 2.24 *If $\varepsilon + \delta < 1/2$, then the probability distribution $\text{Pr}(y) = \text{tr}(SE_y)$ on Y can be used to determine $F(x)$ with an arbitrarily small probability of error by majority sampling, for a sufficiently large sample size. Specifically, take k independent samples from the probability distribution Pr on Y and choose*

a value on which more than half of the samples agree. The probability that the value $F(x)$ is sampled more than $k/2$ times is at least

$$1 - \exp(-2\gamma^2 k) \text{ where } \gamma = (1/2) - \varepsilon - \delta.$$

(This follows by applying the Chernoff bound. In probability theory, the Chernoff bound, named after Herman Chernoff, gives exponentially decreasing bounds on tail distributions of sums of independent random variables. It is better than the first or second moment based tail bounds such as Markov's inequality or Chebyshev inequality, which only yield power-law bounds on tail decay. It is related to the (historically earliest) Bernstein inequalities, and to Hoeffding's inequality. Let X_1, \dots, X_n be independent Bernoulli random variables, each having probability $p > 1/2$. Then the probability of simultaneous occurrence of more than $n/2$ of the events $\{X_k = 1\}$ has an exact value P , where

$$P = \sum_{i=[n/2]+1}^n \binom{n}{i} p^i (1-p)^{n-i}.$$

The Chernoff bound shows that P has the following lower bound:

$$P \geq 1 - e^{-2n(p-\frac{1}{2})^2}.$$

This result admits various generalizations as outlined below. One can encounter many flavours of Chernoff bounds: the original *additive form* (which gives a bound on the absolute error) or the more practical *multiplicative form* (which bounds the error relative to the mean).)

3 Basic Concepts from Quantum Mechanics

Let us first give the topological interpretation of interchanging *two* identical particles in 2 and 3 spatial dimensions. We need the following definition.

Definition 3.1 *If the exchange of two identical particles leaves the state unchanged the particles are termed bosons, and if the state gains a negative sign the particles are termed fermions.*

In 1 spatial dimension and 1 time dimension, *quantum statistics is not well-defined and bosons are equivalent to fermions*. In 3 spatial dimensions and 1 time dimension, *there are only two possible symmetries — the wave function of bosons is symmetric under exchange while that of fermions is anti-symmetric*. However, in 2 spatial dimensions and 1 time dimension, the following proposition holds.

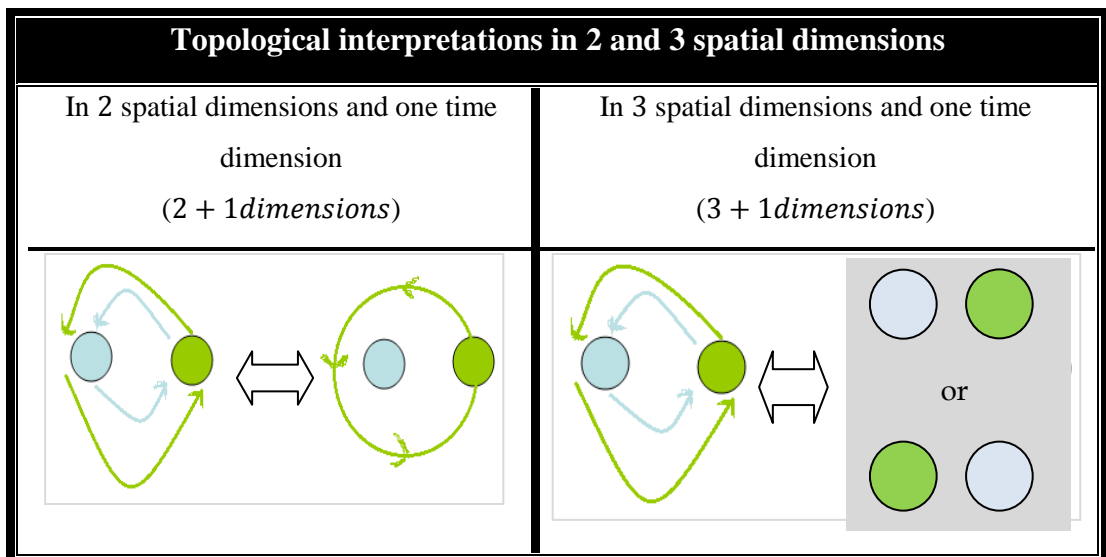
Proposition 2.1 *When one particle is exchanged in a counterclockwise manner with the other, the wave function can change by an arbitrary phase*

$$\psi(\mathbf{r}_1, \mathbf{r}_2) \mapsto e^{i\theta} \psi(\mathbf{r}_1, \mathbf{r}_2).$$

The phase need not be merely a \pm sign because a second counter-clockwise exchange need not lead back to the initial state but can result in a non-trivial phase:

$$\psi(\mathbf{r}_1, \mathbf{r}_2) \mapsto e^{2i\theta} \psi(\mathbf{r}_1, \mathbf{r}_2).$$

'statistical angle' θ	(identical) particles
$\theta = 0$	bosons
$\theta = \pi$	fermions
$\theta \neq 0, \pi$	anyons



<p>Exchanging two identical particles (:anyons) twice is topologically equivalent to bringing one particle in a closed circle enclosing the other particle.</p>	<p>Exchanging two identical particles (:bosons or fermions) twice is topologically equivalent to not moving either particle.</p>
---	--

Let us now give the topological interpretation of interchanging *several* identical particles in 2 and 3 spatial dimensions. To do so, we make use of the following assumption.

The *world lines* of n particles are being interchanged, so that

- the set of final coordinates of the particles is the same as the initial set of coordinates, while
- not requiring each one is returned to its initial position.

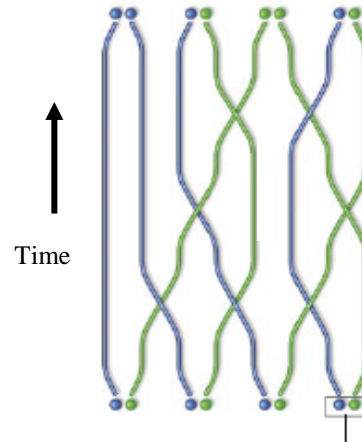
Under this condition, we can detail the aimed topological interpretation.

Topological interpretations in 2 and 3 spatial dimensions	
<p>In 2 spatial dimensions and one time dimension (2 + 1 dimensions)</p>	<p>In 3 spatial dimensions and one time dimension (3 + 1 dimensions)</p>
<p><i>The group formed by the homotopy classes is the braid group</i> B_n <i>on n anyons.</i></p> <p>The braid group is generated by clockwise switches of adjacent particles. That is the set of all s_i where s_i is the clockwise exchange of particles i and $i + 1$ generate the braid group.</p>	<p><i>The set of all homotopy classes of world lines has the group structure of</i> S^n, the permutation group on n letters.</p>

Keeping in mind the above considerations, we are in position to describe how topological quantum computing works.

General framework for topological quantum computations

- **1st** First, pairs of anyons are created and lined up in a row to represent the qubits, or quantum bits, of the computation.
- **2nd** Second, the anyons are moved around by swapping the positions of adjacent anyons in a particular sequence. These moves correspond to operations performed on the qubits.
- **3rd** Finally, pairs of adjacent anyons are brought together and measured to produce the output of the computation.



The output depends on the topology of the particular braiding produced by those manipulations. Small disturbances of the anyons do not change that topology, which makes the computation impervious to normal sources of errors.

4 Diagrammatic theories of braids and knots

4.1. Braids, knots and links

The purpose of this section is to give a quick introduction to the diagrammatic theory of braids and knots. But, why are knots of importance in braid theory? As we shall see in the next section, *knot theory can be used to produce unitary representations of the braid group*. Such representations can play a fundamental role in quantum computing.

Definition 4.1

i. A braid is an embedding of a collection of n strands in the three dimensional space that have their ends in two rows of points that are set one above the other

with respect to a choice of vertical. The n strands are not individually knotted and they are disjoint from one another. The braid set on n strands is denoted by B_n .

ii. A knot is an embedding of a circle in the three dimensional space, taken up to ambient isotopy. More precisely, let N and M be manifolds and g and h be embeddings of N in M . A continuous map $F: M \times [0,1] \rightarrow M$ is defined to be an ambient isotopy taking g to h if F_0 is the identity map, each map F_t is a homeomorphism from M to itself, and $F_1 \circ g = h$. This implies that the orientation must be preserved by ambient isotopies. For example, two knots which are mirror images of each other are in general not equivalent.

iii. A link is an embedding of a disjoint collection of circles in the three dimensional space, taken up to ambient isotopy.

Figure 2 illustrates some indicative knot diagrams. These diagrams are regarded both as schematic pictures of knots, and as plane graphs with extra structures at the nodes (indicating how the curve of the knots pass over or under itself by standard pictorial conventions).

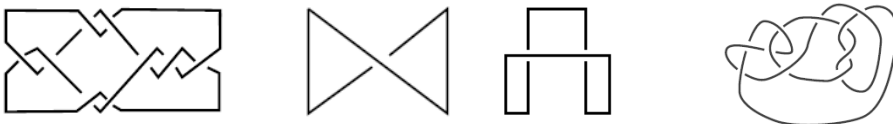


Figure 2: Knot diagrams

It is clear that *every braid can be converted into a knot (or link) by forming the closure*. The knot or link type resulting from performing this operation on a braid X is known as the *closure* of X and will be denoted by $b(X)$. Figure 3 illustrates how to close a braid by attaching the top strands to the bottom strands by a collection of parallel arcs.

Theorem 4.2 ([1], [2], [49], [52]) *Every knot or link can be represented as a closed braid.*

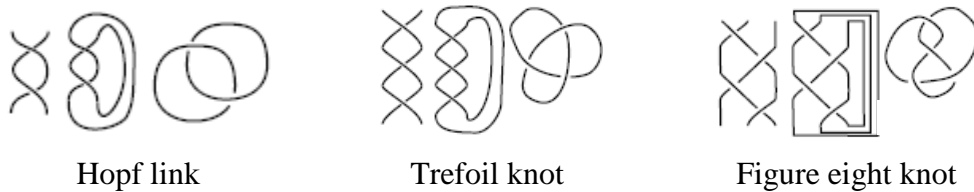


Figure 3: Closing braids to form knots and links

Alexander's theorem provides the converse statement.

Remark 4.3 The problem of deciding whether two knots are isotopic is an example of a *placement problem*, a problem of studying the topological forms that can be made by placing one space inside another. In the case of knot theory we consider the placements of a circle inside the three dimensional space.

Open Question 4.4 The *braid index* of a knot or link K is the minimum number n such that there exists a braid $X \in B_n$ whose closure $b(X)$ represents K . *It is an open problem to determine the braid index of a knot algorithmically.*

Ambient isotopy is mathematically the same as the equivalence relation generated on diagrams by the *Reidemeister moves*. These moves are illustrated in Figure 4 bellow. Each move is performed on a local part of the diagram that is topologically identical to the part of the diagram illustrated in this Figure (these figures are representative examples of the types of Reidemeister moves) without changing the rest of the diagram. The Reidemeister moves are useful in doing combinatorial topology with knots and links, notably in working out the behavior of knot invariants. Furthermore, the Reidemeister moves are of great use for analyzing the structure of knot invariants and they are closely related to the Artin braid group, which we discuss below.

Successive application of Reidmeister moves gives *equivalent knot diagrams*. A formal mathematical definition is that two knots are **equivalent**

if one can be transformed into the other via a type of deformation of \mathbb{R}^3 upon itself, known as an ambient isotopy.

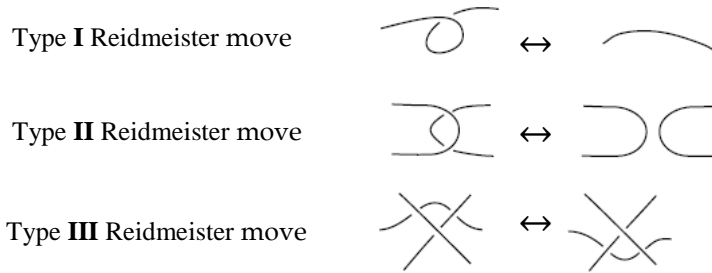


Figure 4: The Reidmeister moves

Conversely, we have the following.

Theorem 4.5([43])*Two diagrams in the three dimensional space represent the same knot if and only if the diagrams are ambient isotopic through a sequence of Reidmeister moves.*

Reidmeister’s theorem reduces the problem of distinguish two knot diagrams into the problem of relating conceptual objects to knot diagrams in a manner that is invariant under Reidmeister’s moves. The relating machine can be formalized as follows.

Theorem 4.6 ([43]) *A knot invariant with values in a set E is a function*

$$f: \{\text{knots diagrams}\} \rightarrow E$$

such that if the knot diagrams D and D' are ambient isotopic through a sequence of Reidmeister moves then $f(D) = f(D')$. In other words, a knot invariant is a "quantity" that is the same for equivalent knots. In particular, a knot polynomial is a knot invariant that is a polynomial.

A general pattern to produce knot invariants is to take any function

$$g : \{\text{classes of equivalent knots diagrams}\} \rightarrow E$$

and, then, consider the functional composition $f = g \circ \rho$ where

$$\rho: \{\text{knot diagrams}\} \rightarrow \{\text{classes of equivalent knots diagrams}\}$$

is the function associating to each knot diagram the equivalence class of this knot. Due to Reidmeister's theorem, any invariant can be obtained following this general pattern.

Of course, the main problem for such a technical construction is to formulate (and understand!) ρ . To be more specific, let us choose

$$E = \{\text{classes of equivalent knots diagrams}\} \text{ and } f(D) = \rho(D).$$

For this choice, the knowledge of ρ would render the corresponding knot invariant a very precious topological tool, since it would be equivalent to the knowledge of an efficient answer to the equivalent knot representation problem of two diagrams. Instead, we may simplify by taking relevant functions f that lose enough of information on knots represented by diagrams. In this direction, let us give some classic examples of knot invariants.

4.1.1. The crossing number

Of course, the most plausibly defined invariant would be the *number of crossings* in a knot diagram. But, such a number depends strongly on the diagram chosen to represent the knot and, therefore, is not an invariant. This requires considering the minimum number of crossings in a diagram of knot. This minimum number equals 0 for the single knot and 3 for the trefoil knots. This is exactly the complexity used for ordering the table of knots. The raised problem of this invariant is that theoretical calculation for knots represented by a given diagram would generate all equivalent diagrams and then take the minimum number of crossings. Since the number of these diagrams is infinite this is not feasible. However, we can immediately obtain an estimate from the top of this invariant: *the number of crossings of any knot diagram is greater than or equal to the number of crossings of the knot.*

4.1.2. The gordian number

Another classical invariant is the *gordian number*. Given a knot, it is possible to transform it into a single knot in the moving in space and allowing the segments that make up the cut to a finite number of times. The gordian number is the minimum number of such sections necessary to transform the knot to a single knot. Following this definition, the gordian number of a single knot is 0. Again, it is not generally easy to calculate that number, but it is not difficult to estimate it: given a diagram of knot, it is easily seen that changes of the data above / down around some crossings lead to a single knot diagram. So the gordian number is always less than or equal to the number of crossings in any knot diagram (and therefore the crossing number of the knot).

4.1.3. The “three-color” invariant

This invariant is a first example of completely computable invariant. Note that any diagram with $n(> 0)$ knot crossings is formed by n arcs whose ends are exactly the intersections. Around each intersection there are exactly three arcs: one who goes above and two pieces of that which passes underneath. Fix now three colors such as red, blue, and green.

Definition 4.7(three-color)

- i. A *three-color* in a diagram D is the choice of three colors for each arc of the diagram, so that around each crossing of the diagram appear either three times on the same color or three different colors.
- ii. A knot is said to be *tricolor* if it admits a diagram with a three-color at least once using each color.

What is most remarkable in this definition is that a priori the knot depends on the diagram of the knot chosen for the test, but in fact we can see that if D is a diagram tricolorable and D' is obtained by applying a Reidemeister move to D ,

then D' is also tricolorable. So, thanks to the Reidemeister theorem, the tricolorability does not depend of the diagram but only on the knot!

4.2 Finitely generated braid groups

4.2.1. The Artin's braid group on n strands. Theoretical presentation

The braid set on n strands, denoted by B_n , is a group which has an intuitive geometrical representation, and in a sense generalizes the symmetric group S_n . Here, n is a natural number; if $n > 1$, then B_n is an infinite group. Braid groups find applications in knot theory, since any knot may be represented as the closure of certain braids. The symmetric group S_n on a finite set of n symbols is the group whose elements are all the permutations of the n symbols, and whose group operation is the composition of such permutations, which are treated as bijective functions from the set of symbols to itself. Since there are $n!$ possible permutations of a set of n symbols, it follows that the order (the number of elements) of the symmetric group S_n is $n!$.

To explain how to reduce a braid group in the sense of Artin to a *fundamental group*, we consider a *connected manifold* M of dimension at least 2. The *symmetric product* of n copies of M means the quotient of M^n , the n -fold Cartesian product of M with itself, by the permutation action of the *symmetric group* S_n on n letters operating on the indices of coordinates. That is, an ordered n -tuple is in the same orbit as any other that is a re-ordered version of it. A path in the n -fold symmetric product of M is the abstract way of discussing n points of M , considered as an unordered n -tuple, independently tracing out n strings. Since we must require that the strings never pass through each other, it is necessary that we pass to the subspace Y of the symmetric product, of orbits of n -tuples of *distinct* points. That is, we remove all the subspaces of M^n defined by conditions $x_i = x_j$.

This is invariant under the symmetric group S_n , and Y is the quotient by the symmetric group of the non-excluded n -tuples:

$$Y = [M^n \setminus \{(x_1, \dots, x_n) | x_i = x_j \text{ for some } i \neq j\}] / S_n,$$

where the symmetric group S_n acts freely on $\mathbb{C}^n \setminus \Delta$ by permuting coordinates. Under the dimension condition Y will be connected. With this definition, then, we can call *the braid group of X with n strings the fundamental group of Y* (for any choice of base point – this is well-defined up to isomorphism). (For convenience of the reader, we will recall the definition of the *fundamental group*. Let X be a topological space, and let x_0 be a point of X . We are interested in the following set of continuous functions called loops with base point x_0 : $\{f: [0,1] \rightarrow X | f(0) = x_0 = f(1)\}$. Now, the fundamental group of X with base point x_0 is this set modulo homotopy h

$$\{f: [0,1] \rightarrow X | f(0) = x_0 = f(1)\} / h$$

equipped with the group multiplication defined by

$$(f * g)(t) := \begin{cases} f(2t) & \text{if } 0 \leq t \leq 1/2 \\ g(2t - 1) & \text{if } 1/2 \leq t \leq 1 \end{cases}$$

Thus the loop $f * g$ first follows the loop f with "twice the speed" and then follows g with "twice the speed". The product of two homotopy classes of loops $[f]$ and $[g]$ is then defined as $[f * g]$, and it can be shown that this product does not depend on the choice of representatives. With the above product, the set of all homotopy classes of loops with base point x_0 forms the fundamental group of X at the point x_0 and is denoted $\pi_1(X, x_0)$ or simply $\pi(X, x_0)$. The *identity element* is the constant map at the base point, and the *inverse of a loop f* is the loop g defined by $g(t) = f(1 - t)$. That is, g follows f backwards.

Although the fundamental group in general depends on the choice of base point, it turns out that, up to isomorphism, this choice makes no difference as long as the space X is path-connected. For path-connected spaces, therefore, we can write $\pi_1(X)$ instead of $\pi_1(X, x_0)$ without ambiguity whenever we care about the isomorphism class only. Here are two basic examples of fundamental groups.

- *Trivial fundamental group.* In Euclidean space \mathbb{R}^n , or any convex subset of \mathbb{R}^n , there is only one homotopy class of loops, and the fundamental group is therefore the trivial group with one element. A path-connected space with a trivial fundamental group is said to be simply connected.
- *Knot theory.* A somewhat more sophisticated example of a space with a non-abelian fundamental group is the complement of a trefoil knot in \mathbb{R}^3 .)

The case where M is the Euclidean plane \mathbb{C} is the original one of Artin. In such a case, identifying the vertical axis for a braid with time and taking the intersection of horizontal planes with the braids shows that the elements of B_n can be thought of as motions of n distinct points in the plane. Thus $B_n = \pi_1([\mathbb{C}^n \setminus \Delta]/S_n)$ when Δ is the set $\{(z_1, \dots, z_n) \in \mathbb{C}^n \mid z_i = z_j \text{ for some } i \neq j\}$. But Δ is the zero-set of the frequently encountered function $\prod_{i < j} (z_i - z_j)$ so the braid group may naturally be generalized as the fundamental group of \mathbb{C}^n minus the singular set of some algebraic function. Or, motions of points can be extended to motions of the whole plane and a braid defines a diffeomorphism of the plane \mathbb{C} minus n points. Thus the braid group may be generalized as the mapping class group of a surface with marked points.

Open Question 4.8 *Describe and characterize the fundamental group of \mathbb{C}^n minus the singular set of some algebraic function.*

4.2.2. The Artin's braid group intuitive presentation

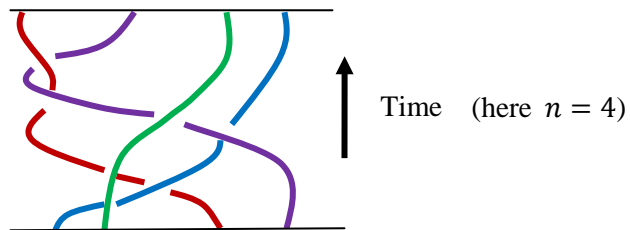
4.2.2. i. *Physical Background*

- *Visualization*

An element of the n – particle braid group B_n can be visualized by thinking of trajectories of n anyons as worldlines (or strands) in $2 + 1$ dimensional space-time originating at initial positions and terminating at final positions. These trajectories are robust with respect to local perturbations (*topology is preserved*).

- **Interpretation**

An element of the n -particle braid group is an equivalence class of such trajectories up to smooth deformations. *To represent an element of a class, we draw the trajectories on paper with the initial and final points ordered along of lines at the initial and final times.* Intuitively, a braid on n strings is a collection of curves in \mathbb{R}^3 joining n points in a horizontal plane to the n points directly above them on another horizontal plane. If, in particular, the initial and end points of the braid are on straight lines, the braid can be drawn as in the example below.

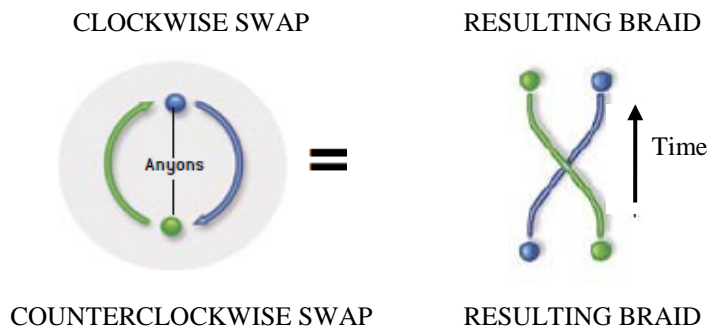


Remark 4.9 The crucial property of a braid is that the tangent vector to the curves can never be horizontal. Braids are considered up to isotopies which are supported between the top and bottom planes.

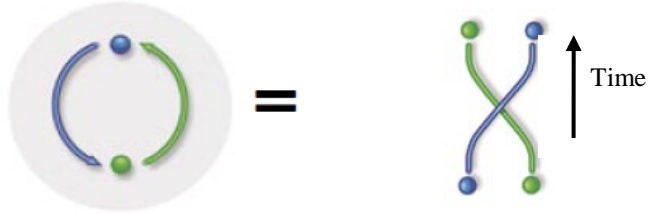
Remark 4.10 Waves are braids where only one anyon moves around the others.

4.2.2. ii. Topological Background

When drawing the trajectories in 2 spatial dimensions, we must be careful to distinguish when one strand passes over or under another, corresponding to a



clockwise or a counter-clockwise exchange (swap).

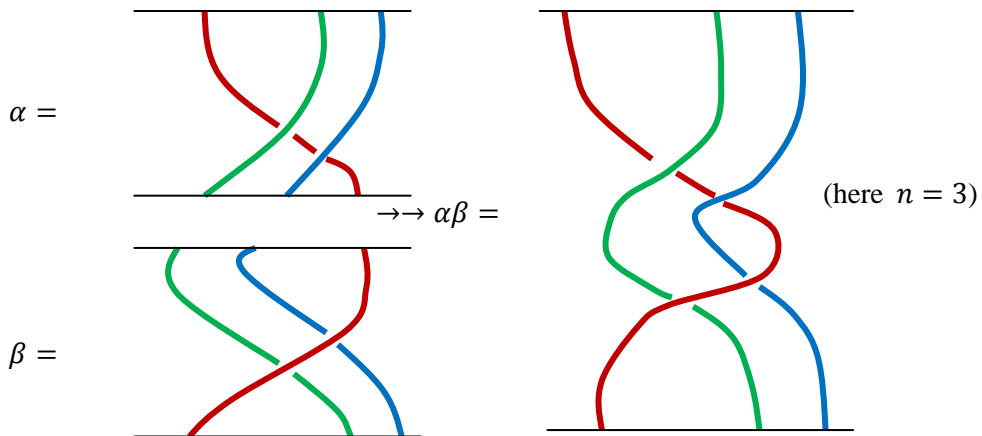


4.2.2. iii. Algebraic Background

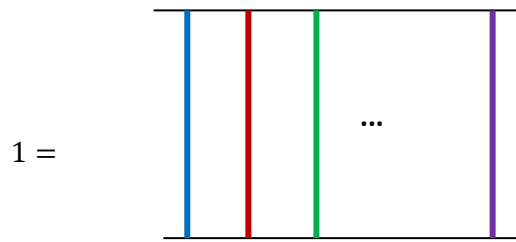
We make the following assumptions.

- Any intermediate time slice must intersect n strands.
- Strands cannot 'double back', which would amount to particle creation/annihilation at intermediate stages. We do not allow this because we assume that the particle number is known.

Then, the multiplication of two elements of the braid group is simply the successive execution of the corresponding trajectories, i.e. the vertical stacking of the two drawings. Intuitively, the (*not* commutative) multiplication of two braids on n strings is defined under concatenation (plus some isotopy) as below:



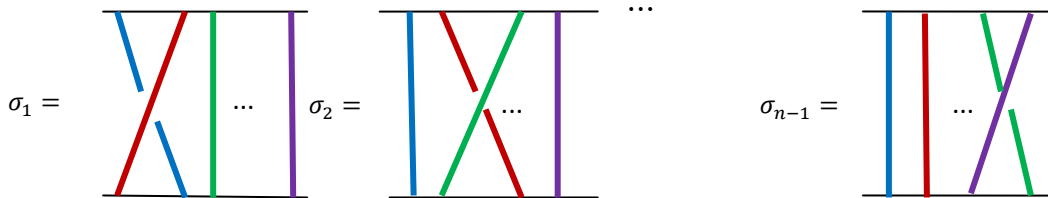
With this multiplication, the set B_n of braids on n strings becomes a (non-abelian) group with unit element defined as follows.



Artin’s braid group B_n can now be represented algebraically in terms of $n - 1$ generators. In general,

$$B_1 \simeq 0, B_2 \simeq \mathbb{Z}.$$

For $n \geq 3$, we let $\sigma_1, \sigma_2, \dots, \sigma_{n-1}$ be the $n - 1$ braids below:



σ_i is a counter-clockwise exchange of the i^{th} and $(i + 1)^{th}$ points. σ_i^{-1} is, therefore, a clockwise exchange of the i^{th} and $(i + 1)^{th}$ points. The σ_i s satisfy the defining relations

$$(1) \quad \begin{cases} \sigma_i \sigma_j = \sigma_j \sigma_i \text{ for } |i - j| \geq 2 \\ \sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1} \text{ for } 1 \leq i \leq n - 1 \\ \sigma_i^2 \neq 1 \text{ for } 1 \leq i \leq n \end{cases}.$$

Example

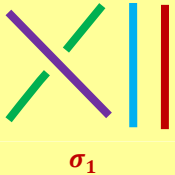
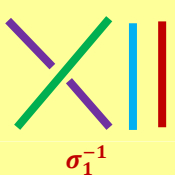
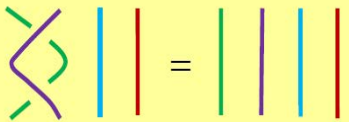
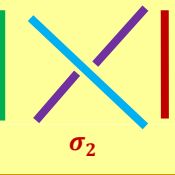
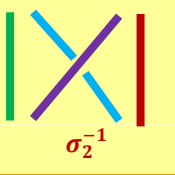
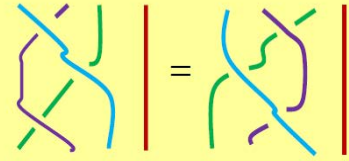
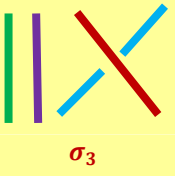
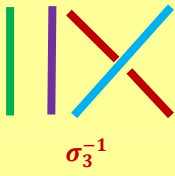
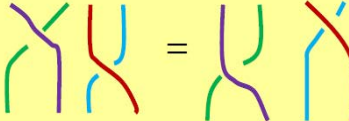
Constructing the braid generators of B_4

- ▶ We choose an arbitrary ordering of the particles 1, 2, 3, 4.
- ▶ σ_i is a counter-clockwise exchange of the i^{th} and $(i + 1)^{th}$ particles.
- ▶ σ_i^{-1} is a clockwise exchange of the i^{th} and $(i + 1)^{th}$ particles.

► The σ_i s satisfy the defining relations,

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

$$\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1} \text{ for } 1 \leq i \leq 3.$$

The braid generators of B_4		The basic braid relations in B_4	
 σ_1	 σ_1^{-1}	1 st line: the equation $\sigma_1 \sigma_1^{-1} = 1$	
 σ_2	 σ_2^{-1}	2 nd line: The braid relation $\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}$ ($1 \leq i \leq n - 1$)	
 σ_3	 σ_3^{-1}	3 rd line: $\sigma_1 \sigma_3 = \sigma_3 \sigma_1$. (However $\sigma_2 \sigma_1 \neq \sigma_1 \sigma_2$.)	

Remark 4.9 The only difference from the *symmetric (permutation) group* S_n is that $\sigma_i^2 \neq 1$, but this makes an enormous difference. While the permutation group is finite, the number of elements in the group $|S_n| = n!$, *the braid group is infinite*, even for just two particles. Furthermore, there are non-trivial topological classes of trajectories even when the particles are distinguishable, e.g. in the two-particle case those trajectories in which one particle winds around the other an integer number of times. These topological classes correspond to the elements of the “pure” braid group, which is the subgroup of the braid group containing only elements which bring each particle back to its own initial position, not the initial position of one of the other particles. The richness of the braid group is the key fact enabling quantum computation through quasiparticle braiding. □

4.3. Infinitely generated braid groups

There are many ways to generalize the notion of braid group on n strands to a braid group on an infinite number of strands. The simplest way is take the direct limit of braid groups, where the attaching maps $f: B_n \rightarrow B_{n+1}$ send the $n - 1$ generators of B_n to the first $n - 1$ generators of B_{n+1} (i.e., by attaching a trivial strand). Fabel has shown that there are two topologies that can be imposed on the resulting group each of whose completion yields a different group. One is a very tame group and is isomorphic to the mapping class group of the infinitely punctured disk — a discrete set of punctures limiting to the boundary of the disk. The second group can be thought of the same as with finite braid groups.

Place a strand at each of the points $(0, 1/n)$ and the set of all braids — where a braid is defined to be a collection of paths from the points $(0, 1/n, 0)$ to the points $(0, 1/n, 1)$ so that the function yields a permutation on endpoints — is isomorphic to this wilder group. An interesting fact is that the pure braid group in this group is isomorphic to both the inverse limit of finite pure braid groups P_n and to the fundamental group of the Hilbert cube (: the topological product of the intervals $[0, 1/n]$ for $n = 1, 2, 3, 4, \dots$) minus the set

$$\{(x_i)_{i \in \mathbb{N}} / x_i = x_j \text{ for some } i \neq j\}.$$

5 Representations of braid groups and invariant of knots

Before the discovery of Hecke algebra representations of the braid group (discussed in this section) very little was known about finite dimensional but infinite representations of B_n , except for the ubiquitous Burau representation. That matter changed dramatically in 1987 with a pioneer publication by V. Jones ([28]). Suddenly, we had more knot invariants and with them more braid group representations than anyone could deal with, and the issue became one of organizing them.

However, we shall not attempt to give a comprehensive overview of the rich theory of representations of braid groups in this section. Instead, we focus here on representations of B_n which have played the greatest roles in the development of that theory: the *Burau representation*, the *Hecke algebra representations*, the *Lawrence-Krammer representation* and the *representations of B_n from the solutions of the Yang-Baxter equation*.

Let us recall the two ways to say what a representation of Artin's braid group B_n is.

The first uses the idea of an action, generalizing the way that matrices act on column vectors by matrix multiplication. A *representation* of the braid group B_n on a vector space V is a map

$$\Phi: B_n \times V \rightarrow V$$

with two properties. First, for any g in B_n , the map

$$\rho(g): V \rightarrow V; v \mapsto \rho(g)(v) := \Phi(g, v)$$

is linear (over a field \mathbb{F}), and similarly in the algebra cases. Second, if we introduce the notation $g \cdot v$ for $\Phi(g, v)$, then for any g_1, g_2 in B_n and v in V :

$$(2) \quad e \cdot v = v,$$

$$(3) \quad g_1 \cdot (g_2 \cdot v) = (g_1 \cdot g_2) \cdot v$$

where e is the identity element of B_n and $g_1 \cdot g_2$ is product in B_n . The requirement for associative algebras is analogous, except that associative algebras do not always have an identity element, in which case equation (2) is ignored. Equation (3) is an abstract expression of the associativity of matrix multiplication.

The second way to define a *representation* of Artin's braid group B_n focuses on the map ρ sending g in B_n to

$$\rho(g): V \rightarrow V; v \mapsto \Phi(g, v).$$

This approach is both more concise and more abstract. A representation of the braid group B_n on a vector space V is a group homomorphism

$$\rho: B_n \rightarrow GL(V, \mathbb{F}).$$

with the following property

$$\rho(g_1 \cdot g_2) = \rho(g_1) \circ \rho(g_2) \text{ for all } g_1, g_2 \text{ in } B_n.$$

The vector space V is called the representation space of ρ and its dimension (if finite) is called the dimension of the representation (sometimes *degree*). It is also common practice to refer to V itself as the representation when the homomorphism ρ is clear from the context; otherwise the notation (V, ρ) can be used to denote a representation. When V is of finite dimension n , one can choose a basis for V to identify V with \mathbb{F}^n and hence recover a matrix representation with entries in the field \mathbb{F} .

A faithful or effective representation is a representation (V, ρ) for which the homomorphism ρ is injective.

In what follows we will only be concerned with representations of the second type.

5.1. The Burau representation

Burau first introduced his representation of the braid group in 1936 ([14]). Much later, it was realized that it could be thought of as a deformation of the standard representation of the symmetric group S_n corresponding to the partition $n = (n - 1) + 1$. For many years it was the focus of the representation theory of braid groups.

Note that, by (1), to find linear representations of B_n ($n \geq 3$), it suffices to find matrices $\rho_1, \rho_2, \dots, \rho_{n-1}$ satisfying (1) (with σ replaced by ρ). One such representation (of dimension n) called the *unreduced Burau representation* is given by the row-stochastic matrices

$$\rho_1 = \begin{pmatrix} 1-t & t & 0 & 0 & \dots \\ 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \rho_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots \\ 0 & 1-t & t & 0 & \dots \\ 0 & 1 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \dots \quad \rho_{n-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \dots \\ \vdots & \vdots & \vdots & 1-t & t \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Thus, we define the unreduced Burau representation of dimension n

$$\rho: B_n \rightarrow V = GL_n(\mathbb{Z}[t, t^{-1}])$$

as follows:

$$\sigma_i \mapsto \rho(\sigma_i) = \rho_i \equiv \mathbb{I}_{i-1} \oplus \begin{pmatrix} 1-t & t \\ 1 & 0 \end{pmatrix} \oplus \mathbb{I}_{n-i-1} \text{ for } 1 \leq i \leq n-1$$

where \mathbb{I}_k denotes the $k \times k$ identity matrix.

Substituting $t = 1$ gives back the representation (of B_n factoring through S_n), and this is why we say that it is a deformation of the standard representation of S_n . Like the representation of S_n , the Burau representation splits into a 1-dimensional representation and an $(n - 1)$ -dimensional irreducible representation known as the reduced Burau representation which we denote by

$$\tilde{\rho}: B_n \rightarrow GL_n(\mathbb{Z}[t, t^{-1}])$$

as follows:

$$\sigma_i \mapsto \tilde{\rho}(\sigma_i) = \tilde{\rho}_i \equiv \mathbb{I}_{i-2} \oplus \begin{pmatrix} 0 & -t & 0 \\ 0 & -t & 0 \\ 0 & -1 & 1 \end{pmatrix} \oplus \mathbb{I}_{n-i-2}$$

for $1 \leq i \leq n-2$

where the $-t$ in the middle of the 3×3 matrix is always in the $(i, i)^{th}$ spot.

Remark 5.1 Let us see how Burau’s representation can be defined in a more rigorous manner. To do so, consider the braid group B_n to be the mapping class group of a disc (: the group of isotopy-classes of automorphisms of a disc) with n marked points P_n . The homology group $H_1 P_n$ is free abelian of rank n . Moreover, the invariant subspace of $H_1 P_n$ (under the action of B_n) is primitive and infinite

cyclic. Let $\pi: H_1 p_n \rightarrow \mathbb{Z}$ be the projection onto this invariant subspace. Then there is a covering space \tilde{P}_n corresponding to this projection map. Much like in the construction of the *Alexander polynomial* (Remark 4.3), consider $H_1 \tilde{P}_n$ as a module over the group-ring of covering transformations $[\mathbb{Z}] \equiv \mathbb{Z}[t^\pm]$ (a Laurent polynomial ring). As such a $\mathbb{Z}[t^\pm]$ module, $H_1 \tilde{P}_n$ is free of rank $n - 1$. By the basic theory of covering spaces, B_n acts on $H_1 \tilde{P}_n$, and this representation is called the *reduced Burau representation*. The *unreduced Burau representation* has a similar definition, namely one replaces P_n with its (real, oriented) blow-up at the marked points. Then instead of considering $H_1 \tilde{P}_n$ one considers the relative homology $H_1(\tilde{P}_n, \tilde{\partial})$ where $\partial \subset P_n$ is the part of the boundary of P_n corresponding to the blow-up operation together with one point on the disc's boundary. $\tilde{\partial}$ denotes the lift of ∂ to \tilde{P}_n . As a $\mathbb{Z}[t^\pm]$ module this is free of rank n . Note that, by the homology long exact sequence of a pair, the Burau representations fit into a short exact sequence

$$0 \rightarrow V_r \rightarrow V_u \rightarrow D \oplus \mathbb{Z}[t^\pm] \rightarrow 0,$$

where V_r and V_u are reduced and unreduced Burau B_n -modules respectively and $D \subset \mathbb{Z}^n$ is the complement to the diagonal subspace (i.e.,

$$D = \{(x_1, x_2, \dots, x_n) \in \mathbb{Z}^n: x_1 + x_2 + \dots + x_n = 0\},$$

and B_n acts on \mathbb{Z}^n by the permutation representation. Reduced Burau representation is related with Alexander polynomial, as follows. *If a knot K is the closure of a braid f , then the Alexander polynomial $\Delta_K(t)$ is given by*

$$\Delta_K(t) = \det(I - f_*)$$

where f_* is the reduced Burau representation of the braid f .

The first nonfaithful Burau representations are found without the use of computer, using a notion of winding number or contour integration ([37]). Now, Burau's representation is known not to be faithful for $n \geq 5$ but faithful for $n \leq 3$ ([5], [34], [36], [41], [46] and [48]). At this time, the case $n = 4$ remains open.

Open Question 5.2 Investigate the faithfulness of the Burau representation when $n = 4$.

Remark 5.3 More generally, it was a major open problem whether braid groups were linear. In 1990, Ruth Lawrence described a family of more general "Lawrence representations" depending on several parameters. Around 2001 Stephen Bigelow and Daan Krammer independently proved that *all braid groups are linear*. Their work used the Lawrence-Krammer representation of dimension $n(n-1)/2$ depending on two variables q and t . By suitably specialising these variables, the braid group B_n may be realized as a subgroup of the general linear group over the complex numbers. Remind that the general linear group of degree n is the set of $n \times n$ invertible matrices, together with the operation of ordinary matrix multiplication. \square

5.2. Representations of B_n from R –matrices

5.2.1. Notation

Let V and W be two vector spaces of dimensions n and m , respectively. Given two bases

$$\{v^{(i)} = (v_1^{(i)}, \dots, v_n^{(i)})^T \in V; i = 1, \dots, n\}$$

and

$$\{w^{(j)} = (w_1^{(j)}, \dots, w_m^{(j)})^T \in W; j = 1, \dots, m\}$$

for V and W , respectively, the tensors

$$\begin{aligned} \{v^{(i)} \otimes w^{(j)} &= (v_1^{(i)} w_1^{(j)}, \dots, v_1^{(i)} w_m^{(j)}, \dots, v_n^{(i)} w_1^{(j)}, \dots, v_n^{(i)} w_m^{(j)})^T; i = 1, 2, \dots, n, j \\ &= 1, 2, \dots, m\} \end{aligned}$$

form a basis for $V \otimes W$ (generally ordered so that $v^{(i)} \otimes w^{(j+1)}$ comes before $v^{(i+1)} \otimes w^{(j)}$).

The dimension $\dim(V \otimes W)$ of the tensor product $V \otimes W$ therefore is the product of dimensions of the original spaces; for instance $\mathbb{R}^m \otimes \mathbb{R}^n$ will have dimension mn . The k^{th} tensor power of the vector space V is the k –fold tensor product of V with itself. That is

$$V^{\otimes k} = \underbrace{V \otimes \dots \otimes V}_{k\text{-times}}$$

A *tensor* on V is an element of a vector space of the form

$$T_r^s(V) = V^{\otimes r} \otimes V^* \otimes^s, \text{ for non-negative integers } r \text{ and } s.$$

The tensor product of the n –matrix

$$A = (a_{i,j})_{i,j=1,2,\dots,n}$$

by the m –matrix

$$B = (b_{i,j})_{i,j=1,2,\dots,m}$$

is the $m \times n$ –matrix given by

$$A \otimes B = \begin{pmatrix} a_{1,1}B & a_{1,2}B & \dots & a_{1,n}B \\ a_{2,1}B & a_{2,2}B & \dots & a_{2,n}B \\ \vdots & \vdots & \vdots & \vdots \\ a_{n,1}B & a_{n,2}B & \dots & a_{n,n}B \end{pmatrix}.$$

If, for instance,

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \text{ and } B = \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix},$$

then

$$A \otimes B = \begin{pmatrix} a_{1,1}b_{1,1} & a_{1,1}b_{1,2} & a_{1,1}b_{1,3} & a_{1,2}b_{1,1} & a_{1,2}b_{1,2} & a_{1,2}b_{1,3} \\ a_{1,1}b_{2,1} & a_{1,1}b_{2,2} & a_{1,1}b_{2,3} & a_{1,2}b_{2,1} & a_{1,2}b_{2,2} & a_{1,2}b_{2,3} \\ a_{1,1}b_{3,1} & a_{1,1}b_{3,2} & a_{1,1}b_{3,3} & a_{1,2}b_{3,1} & a_{1,2}b_{3,2} & a_{1,2}b_{3,3} \\ a_{2,1}b_{1,1} & a_{2,1}b_{1,2} & a_{2,1}b_{1,3} & a_{2,2}b_{1,1} & a_{2,2}b_{1,2} & a_{2,2}b_{1,3} \\ a_{2,1}b_{2,1} & a_{2,1}b_{2,2} & a_{2,1}b_{2,3} & a_{2,2}b_{2,1} & a_{2,2}b_{2,2} & a_{2,2}b_{2,3} \\ a_{2,1}b_{3,1} & a_{2,1}b_{3,2} & a_{2,1}b_{3,3} & a_{2,2}b_{3,1} & a_{2,2}b_{3,2} & a_{2,2}b_{3,3} \end{pmatrix}.$$

In particular, for $B = Id_{\mathbb{R}^3}$, we have

$$A \otimes Id_{\mathbb{R}^3} = \begin{pmatrix} a_{1,1} & 0 & 0 & a_{1,2} & 0 & 0 \\ 0 & a_{1,1} & 0 & 0 & a_{1,2} & 0 \\ 0 & 0 & a_{1,1} & 0 & 0 & a_{1,2} \\ a_{2,1} & 0 & 0 & a_{2,2} & 0 & 0 \\ 0 & a_{2,1} & 0 & 0 & a_{2,2} & 0 \\ 0 & 0 & a_{2,1} & 0 & 0 & a_{2,2} \end{pmatrix};$$

and, for $A = Id_{\mathbb{R}^2}$, we have

$$Id_{\mathbb{R}^2} \otimes B = \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} & 0 & 0 & 0 \\ b_{2,1} & b_{2,2} & b_{2,3} & 0 & 0 & 0 \\ b_{3,1} & b_{3,2} & b_{3,3} & 0 & 0 & 0 \\ 0 & 0 & 0 & b_{1,1} & b_{1,2} & b_{1,3} \\ 0 & 0 & 0 & b_{2,1} & b_{2,2} & b_{2,3} \\ 0 & 0 & 0 & b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix}.$$

Finally, the tensor product of two multilinear maps $f(x_1, x_2, \dots, x_n)$ and $g(x_1, x_2, \dots, x_m)$ is the multilinear function given by

$$(f \otimes g)(y_1, \dots, y_n, y_{n+1}, \dots, y_{n+m}) = f(y_1, \dots, y_n) \Big|_{y_1=x_1, \dots, y_n=x_n} g(y_{n+1}, \dots, y_{n+m}) \Big|_{y_{n+1}=x_1, \dots, y_{n+m}=x_m}.$$

5.2.2. Representations of the braid group and solution of the Yang-Baxter equation

Let V be a vector space over a field \mathbb{F} .

We want to get a representation of B_n on the n -fold tensor product $V^{\otimes n}$ (n^{th} tensor power of V with itself). Here's a simple way. Just map the i^{th} elementary braid generator σ_i to the linear map taking

$$v^{(1)} \otimes v^{(2)} \otimes \dots \otimes v^{(n)}$$

to

$$v^{(1)} \otimes v^{(2)} \otimes \dots \otimes R(v^{(i)} \otimes v^{(i+1)}) \otimes \dots \otimes v^{(n)}$$

where $R \in \text{Aut}_{\mathbb{F}}(V \otimes V)$ is an invertible linear transformation of $V^{\otimes 2}$. In other words, we use R to "switch" the i^{th} and $(i+1)^{\text{st}}$ factors, and leave the rest alone.

It's easy to see that this mapping defines a representation of the braid group B_n as long as we can get the following equation to hold, and this is equivalent to having

$$(R \otimes Id_V)(Id_V \otimes R)(R \otimes Id_V) = (Id_V \otimes R)(R \otimes Id_V)(Id_V \otimes R)$$

where Id_V is the identity on V . We are thus

Definition 5.4 ([39], [50]) *Let $R \in Aut_{\mathbb{F}}(V \otimes V)$. The Yang-Baxter equation is the following equation in the group $Aut_{\mathbb{F}}(V \otimes V \otimes V)$*

$$(R \otimes Id_V)(Id_V \otimes R)(R \otimes Id_V) = (Id_V \otimes R)(R \otimes Id_V)(Id_V \otimes R).$$

Solutions of this equation are called R –matrices.

More rigorously, the same representation procedure can be described as follows. Let V be a vector space. Let $R \in Aut_{\mathbb{F}}(V \otimes V)$. Let also $n > 1$ be an integer. For $1 \leq i \leq n - 1$, define a $R_i \in Aut_{\mathbb{F}}(V^{\otimes n})$ by

$$R_i = Id_{V^{\otimes(i-1)}} \otimes R \otimes Id_{V^{\otimes(n-i-1)}}.$$

Clearly, if $|i - j| > 1$, then $R_i R_j = R_j R_i$. It is easy to prove the following.

Lemma 5.5 ([39], [50]) *Under the above assumptions, we have in the group $Aut_{\mathbb{F}}(V^{\otimes n})$*

$$R_i R_{i+1} R_i = R_{i+1} R_i R_{i+1}$$

for all i if and only if R is a solution of the Yang-Baxter equation.

Thus, we have

Corollary 5.6 ([39], [50]) *Let $R \in Aut_{\mathbb{F}}(V \otimes V)$ be a solution of the Yang-Baxter equation. Then, for all $n > 1$, there exists a unique group homomorphism*

$$\rho_n^c : B_n \rightarrow Aut_{\mathbb{F}}(V^{\otimes n})$$

given by

$$\rho_n^c(\sigma_i) = R_i \text{ for any } 1 \leq i \leq n - 1.$$

In other words, the R –matrices give Braid group representations.

Remark 5.7 From the point of view of topology, the matrix R is regarded as representing an elementary bit of braiding represented by one string crossing over

another. In Figure 5 we have illustrated the braiding identity that corresponds to the Yang-Baxter equation. Each braiding picture with its three input lines (below) and output lines (above) corresponds to a mapping of the three fold tensor product of the vector space V to itself, as required by the algebraic equation quoted above. The pattern of placement of the crossings in the diagram corresponds to the factors $R \otimes Id_V$ and $Id_V \otimes R$: This crucial topological move has an algebraic expression in terms of such a matrix R : Our approach in this section to relate topology, quantum computing, and quantum entanglement is through the use of the Yang-Baxter equation. In order to accomplish this aim, we need to study solutions of the Yang-Baxter equation that are unitary. Then the R matrix can be seen either as a braiding matrix or as a quantum gate in a quantum computer.

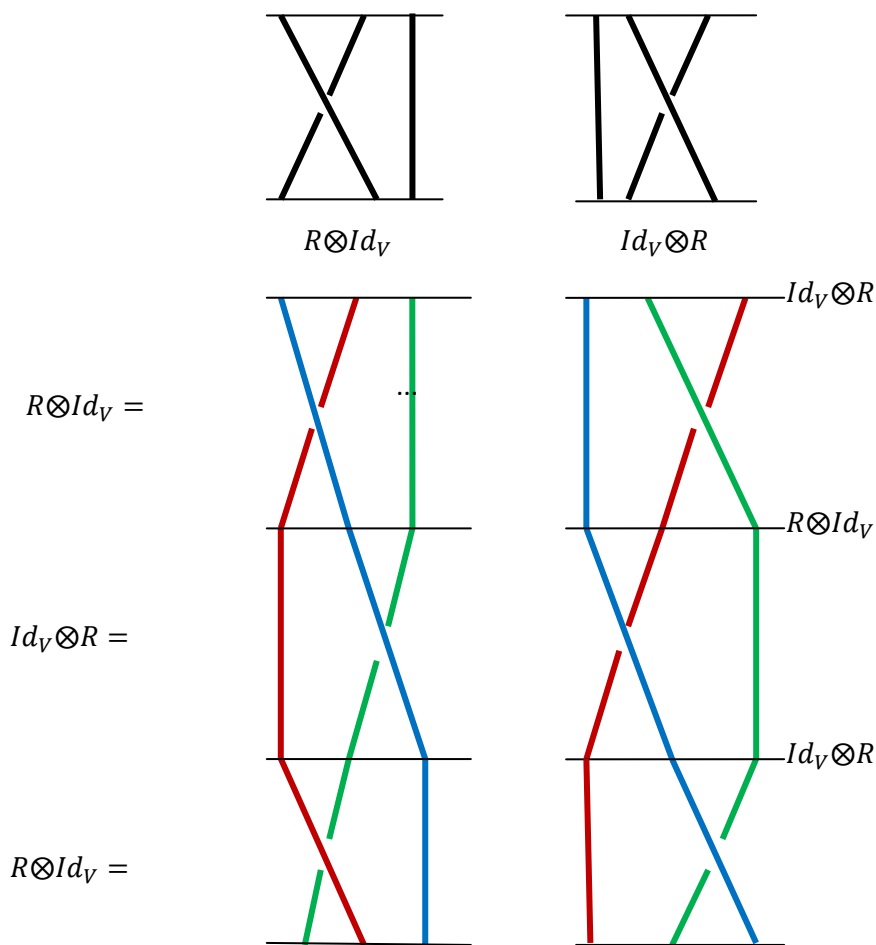


Figure 5: The Yang-Baxter equation

$$(R \otimes Id_V)(Id_V \otimes R)(R \otimes Id_V) = (Id_V \otimes R)(R \otimes Id_V)(Id_V \otimes R)$$

The solutions of the Yang-Baxter equation are usually very hard to find. But once found, they provide interesting representations of the braid group B_n . Here is given a solution of the Yang-Baxter equations for $n = 2$.

Lemma 5.8 (A solution of the Yang-Baxter equation) *Let V be spanned by two vectors X and Y , and define R by*

$$R(X \otimes X) = X \otimes X$$

$$R(Y \otimes Y) = Y \otimes Y$$

$$R(X \otimes Y) = q(Y \otimes X)$$

$$R(Y \otimes X) = q(X \otimes X) + (1 - q^2)(Y \otimes X)$$

Then, R satisfies the Yang-Baxter equation and

$$R^2 = (1 - q^2)R + q^2.$$

This quadratic equation is called a *Hecke relation*.

4.2.3. Relating Yang-Baxter equation with unitary R-matrices and universal gates

Relating topology, quantum computing (and quantum entanglement) is through the use of the Yang-Baxter equation. In order to accomplish this aim, we need to study solutions of the Yang-Baxter equation that are *unitary*. Then the R matrix can be seen either as a braiding matrix or as a quantum gate in a quantum computer.

The problem of finding solutions to the Yang-Baxter equation that are unitary turns out to be surprisingly difficult. In 2003, Dye has classified all such 4×4 matrices.

Theorem 5.9 ([24]) *All 4×4 unitary solutions to the Yang-Baxter equation are similar to one of the following types*

$$\begin{aligned}
 \text{i. } R &= \begin{pmatrix} 1/\sqrt{2} & 0 & 0 & 1/\sqrt{2} \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 0 & 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 0 & 0 & 1/\sqrt{2} \end{pmatrix} \\
 \text{ii. } R' &= \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & 0 & b & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & 0 & d \end{pmatrix} \\
 \text{iii. } R'' &= \begin{pmatrix} 0 & 0 & 0 & a \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ d & 0 & 0 & 0 \end{pmatrix}
 \end{aligned}$$

where a, b, c, d are unit complex numbers.

There is a remarkable correlation between unitary solutions to the Yang-Baxter and universality of (quantum) gates. Let V be a two complex dimensional vector space. Let also $G : V \otimes V \rightarrow V \otimes V$ be any unitary linear mapping. Recall that the mapping G is said to be a *two-qubit gate* and that the gate G is *universal for quantum computation (or just universal)* if G together with local unitary transformations (unitary transformations from V to V) generates all unitary transformations of the complex vector space of dimension 2^n to itself.

Definition 5.10 *The gate G is said to be entangling if there is a vector*

$$|\varphi\rangle \otimes |\psi\rangle \in V \otimes V$$

such that

$$G(|\varphi\rangle \otimes |\psi\rangle)$$

is not decomposable as a tensor product of two qubits. Under these circumstances, one says that $G(|\varphi\rangle \otimes |\psi\rangle)$ is entangled.

Remark 5.11 ([30]) A two-qubit pure state

$$|\varphi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$$

is entangled exactly when $(ad - bc) \neq 0$. It is easy to use this fact to check when a specific matrix is, or is not, entangling. \square

In 2002, J. L. Brylinski and R. Brylinski gave a general criterion of G to be universal.

Theorem 5.12 ([13]) *A two-qubit gate G is universal if and only if it is entangling.*

Let us give an indicant example.

Example 5.13([30])

i. *Let \mathcal{D} denote the phase gate shown below. \mathcal{D} is a solution to the algebraic Yang-Baxter equation Then \mathcal{D} is a universal gate.*

$$\mathcal{D} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

ii. *The matrix solution*

$$R = \begin{pmatrix} 1/\sqrt{2} & 0 & 0 & 1/\sqrt{2} \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 0 & 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 0 & 0 & 1/\sqrt{2} \end{pmatrix}$$

to the Yang-Baxter equation is a universal gate.

iii. *The matrix solutions*

$$R' = \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & 0 & b & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & 0 & d \end{pmatrix} \quad \text{and} \quad R'' = \begin{pmatrix} 0 & 0 & 0 & a \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ d & 0 & 0 & 0 \end{pmatrix}$$

to the Yang-Baxter equation are universal gates exactly when $ad - bc \neq 0$. \square

5.3. Hecke algebras representations of braid groups and polynomial invariants of knots

A simple calculation, together with the Cayley-Hamilton theorem, shows that the image of each of our braid group generators under the Burau representation, $\rho(\sigma_i)$, satisfies the characteristic equation

$$x^2 = (1 - t)x + t$$

and thus has two distinct eigenvalues (compare with Hecke relation, above). This prompted Jones to study all representations

$$\rho: B_n \rightarrow GL_n(\mathbb{C})$$

which have at most two distinct eigenvalues ([28]).

Let $x_i = \rho(\sigma_i)$. Then for all i , x_i must satisfy a quadratic equation of the form

$$x_i^2 + ax_i + b = 0.$$

By rescaling, we may assume that one of the eigenvalues is 1 and eliminate one of the variables, e.g.,

$$a = -(1 + b).$$

Note that by rewriting our quadratic equation and making the substitution

$$b = -t$$

we regain the characteristic equation from the Burau representation. However, the convention in the literature seems to be to rescale our representation by (-1) so that the equation takes the form

$$x_i^2 = (t - 1)x_i + t.$$

With this motivation, we define the *Hecke algebra* $H_n(t)$ to be the algebra with generators $1, x_1, \dots, x_{n-1}$ and defining relations as follows:

$$(4) \quad \begin{cases} x_i x_j = x_j x_i \text{ for } |i - j| \geq 2 \\ x_i x_{i+1} x_i = x_{i+1} x_i x_{i+1} \text{ for } 1 \leq i \leq n-1 \\ x_i^2 = (t - 1)x_i + t \text{ for } 1 \leq i \leq n \end{cases}$$

Comparing the relations in (1) and (4), we see that $H_n(1) \cong \mathbb{C}S_n$, the group algebra of the symmetric group. Hence we can think of $H_n(t)$ as a ‘‘deformation’’ of $\mathbb{C}S_n$.

The connection between $H_n(t)$ and $\mathbb{C}S_n$ is made even more transparent by noting that the vector space $H_n(t)$ is spanned by $n!$ lifts of a system of reduced words in the transpositions $s_i \in S_n$. For example, we can take as a spanning set

$$\{(x_{i_1}x_{i_1-1} \cdots x_{i_1-j_1})(x_{i_2}x_{i_2-1} \cdots x_{i_2-j_2}) \cdots (x_{i_r}x_{i_r-1} \cdots x_{i_r-j_r})\}$$

where $1 \leq i_1 < i_2 < \cdots < i_r \leq n - 1$ and $i_k - j_k \geq 1$ ([12], [28]).

Our main purpose in this section is to outline Jones' development in [28] of a two-variable polynomial knot invariant arising from representations of the Hecke algebras $H_n(t)$. This polynomial is essentially the well-known HOMFLY polynomial, and includes the Jones polynomial as a specialization.

We begin by defining a function $f: B_n \rightarrow H_n(t)$ by $f(\sigma_i) = x_i$. The function f is well defined on reduced words in the generators σ_i and commutes with the natural inclusions $B_{n-1} \subset B_n$ and $H_{n-1}(t) \subset H_n(t)$, although in general f fails to be a homomorphism. We can then apply the following result due to Adrian Ocneanu which appeared in [24] and was proved inductively in [28] using the $n!$ -element basis given above.

Theorem 5.14 ([24], [28]) *For each $z \in \mathbb{C}^*$ (and each $t \in \mathbb{C}^*$), there exists a unique trace function $tr: \bigcup_{n=1}^{\infty} H_n(t) \rightarrow \mathbb{C}$ such that*

1. $tr(1) = 1$
2. $tr(ab) = tr(ba)$
3. tr is \mathbb{C} -linear
4. $tr(ux_{n-1}v) = z tr(uv)$ for all $u, v \in H_{n-1}(t)$.

Theorem 5.14 gives us a one-parameter family of trace functions on a one-parameter family of algebras. In fact, using the properties of the trace function given in theorem it is possible to compute $tr(f(X))$ for all $X \in B_n$. (We note the fact that for any $w \in H_n(t)$ such that $w \notin H_{n-1}(t)$, there is a unique reduced word $w = x_{i_1} \cdots x_{i_r}$ in which x_{n-1} appears exactly once [28].) In practice, the third relation of $H_n(t)$ is quite useful for computing the trace function tr , both in its original form and in the following:

$$x_i^{-1} = t^{-1}x_i + (t^{-1} - 1).$$

Example 5.15 Let $x_1 = \sigma_1^3 \in B_2$, and let $x_2 = \sigma_1\sigma_2^{-1}\sigma_1\sigma_2^{-1} \in B_3$. Then

$$\text{tr}(f(x_1)) = (t^2 - t + 1)z + t(t - 1)$$

and

$$\begin{aligned} \text{tr}(f(x_2)) &= (3 - t^{-1} - t)t^{-1}z^2 + (3 - t^{-1} - t)(t^{-1} - 1)z \\ &\quad - (2 - t^{-1} - t). \square \end{aligned}$$

We also note that the second property of the trace function given in Theorem 5.14 implies that $\text{tr} \circ f$ is invariant on conjugacy classes in B_n . It remains to tweak the function a bit in order to obtain from a given braid a two-variable polynomial which is also invariant under stabilization and destabilization moves as defined below, such a polynomial will be an invariant of the knot type of the closed braid.

Algebraically, stabilization and destabilization each take the form

$$X \rightarrow X\sigma_n^{\pm 1},$$

the only difference being appropriate conditions on the braid X . We would like to rescale our representation f in such a way that both versions of stabilization (resp. destabilization) have the same effect on the trace function. Suppose there exists a complex number k such that

$$\text{tr}(kx_i) = \text{tr}((kx_i)^{-1}).$$

Then we can find a ‘formula’ for k as follows:

$$\begin{aligned} k^2 \text{tr}(x_i) = \text{tr}(x_i^{-1}) &\Rightarrow k^2 z = \text{tr}(t^{-1}x_i + t^{-1} - 1) \Rightarrow k^2 \\ &= \{[t^{-1}z + t^{-1} - 1z]/z\} \Rightarrow k^2 = \{1 + z - t\}/tz \end{aligned}$$

Solving this for z , we obtain

$$z = -(1 - t)/(1 - k^2t).$$

We set $\kappa = k^2$, and define $f_\kappa: B_n \rightarrow H_n(t)$ by $f_\kappa(\sigma_i) = \sqrt{\kappa}\sigma_i$. Now we have

$$\text{tr}(f_\kappa(\sigma_n)) = \sqrt{\kappa}z = -\sqrt{\kappa}[(1 - t)/(1 - \kappa t)].$$

and

$$\begin{aligned} \text{tr}(f(w\sqrt{\kappa}\sigma_n)) &= -\sqrt{\kappa}[(1 - t)/(1 - \kappa t)] \text{tr}(f(w)) \\ &= \text{tr}(f(w \cdot [1/\sqrt{\kappa}]\sigma_n^{-1})) \end{aligned}$$

for any $w \in B_n$.

Now we simply define

$$\begin{aligned} F(X) \equiv F_X(t, \kappa) &= \left(-\frac{1}{\sqrt{\kappa}} \frac{1 - \kappa t}{1 - t} \right)^{n-1} \text{tr}(f_\kappa(X)) \\ &= \left(-\frac{1}{\sqrt{\kappa}} \frac{1 - \kappa t}{1 - t} \right)^{n-1} (\sqrt{\kappa})^E \text{tr}(f(X)) \end{aligned}$$

for $X \in B_n$, where E is the exponent sum of X as a word in $\sigma_1, \dots, \sigma_{n-1}$. It is clear that $F(X)$ depends only on the knot type of $b(X)$.

We now reparametrize one last time, setting

$$l = \sqrt{\kappa}\sqrt{t} \text{ and } m = \sqrt{t} - \frac{1}{\sqrt{t}}.$$

With this substitution, we obtain a Laurent polynomial in two variables l and m , which we denote

$$P_{b(X)}(l, m) = P_K(l, m),$$

where K is the (oriented) knot or link type of $b(X)$. Furthermore, $P_K(l, m)$ satisfies the skein relation

$$mP_{K_0} = l^{-1}P_{K_+} - lP_{K_-}$$

where K_0, K_+ , and K_- are oriented knots with identical diagrams except in a neighborhood of one crossing. Thus by beginning with $P_U = 1$, where U denotes the unknot, it is possible to calculate P_K for any knot or link K using only the skein relation, which is often simpler than using the trace function.

Definition 5.16 *The polynomial*

$$P_K(l, m)$$

obtained in this way is essentially the same as the two-variable polynomial known as the HOMFLY polynomial ([24]) which is usually reparametrized as

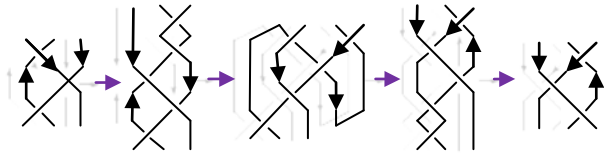
$$P_K(il^{-1}, i m).$$

Example 5.17 Let X_1, X_2 be the braids defined in Example 4.2 whose respective closures are

$K_1 =$
the right –
*handed trefoil knot*²



$K_2 =$ *the right Figure knot*



It is easy to check that

$$\begin{aligned} F_{X_1}(t, \kappa) &= \kappa(1 + t^2 - \kappa t^2) = \kappa t(t + t^{-1} - \kappa t) \\ &= \kappa t(2 - \kappa t + t + t^{-1} - 2) \\ &= \kappa t \left(2 - \kappa t + \left(\sqrt{t} - \frac{1}{\sqrt{t}} \right)^2 \right) \end{aligned}$$

and hence we have

$$P_{K_1}(l, m) = 2l^2 - l^4 + l^2m^2.$$

Similarly, one can check that

$$\begin{aligned} F_{X_2}(t, \kappa) &= \frac{1 - \kappa(1 - t + t^2) + \kappa^2 t^2}{t \kappa} \\ P_{K_2}(l, m) &= l^{-2} - m^2 - 1 + l^2. \end{aligned}$$

For explicit calculations of F_{X_2} and P_{K_2} using the trace function, see p. 350 of [28].

□

There is also a 1-variable knot polynomial, the Jones polynomial, associated to the algebra

$$J_n(t)$$

generated by $1, g_1, \dots, g_{n-1}$ with defining relations

$$\begin{aligned} g_i g_k &= g_k g_i \text{ if } |i - k| \geq 2, \\ g_i g_{i+1} g_i &= g_{i+1} g_i g_{i+1}, \\ g_i^2 &= (t - 1)g_i + t, \end{aligned}$$

²The left-handed trefoil knot is 

$$1 + g_i + g_{i+1} + g_i g_{i+1} + g_{i+1} g_i + g_i g_{i+1} g_i = 0.$$

In the situation of the *Jones algebra* the trace is unique, whereas in the situation of the Hecke algebra, as we presented it here, there is a 1-parameter family of traces. The 1-variable Jones polynomial was discovered before the 2-variable HOMFLY polynomial.

This two-variable knot polynomial has been much studied and reviewed in the literature. For the sake of completeness we list here a few of its noteworthy properties and applications.

1. Connect sums

$$P_{K_1 \# K_2} = P_{K_1} \cdot P_{K_2}.$$

2. Disjoint unions

$$P_{K_1 \sqcup K_2} = \binom{l^{-1}-l}{m} P_{K_1} \cdot P_{K_2}.$$

3. Orientation

$$P_{\bar{K}} = P_K,$$

where \bar{K} denotes the link obtained by reversing the orientation of every component of the link K .

4. Chirality

$$P_{\tilde{K}}(l, m) = P_K(l - 1, -m),$$

where \tilde{K} denotes the mirror image of the link K .

5. Alexander polynomial: Note that $F_X(1, \kappa)$ is not defined. It comes as something of surprise, then, that the specialization $l = 1, m = \sqrt{t} - (1/\sqrt{t})$ gives the *Alexander polynomial*

$$\Delta_K(t) = P_K(1, \sqrt{t} - \frac{1}{\sqrt{t}}).$$

Jones shows how to avoid the singularity by exploiting an alternate method of calculating the trace function using weighted sums of traces (see [51] as well as [24] and [28]). A by-product of this alternate method is another derivation of the Equation

$$\Delta_{b(X)}(t) = \frac{\det(\tilde{\rho}(X) - \mathbb{I}_{n-1})}{1 + t + \dots + t_{n-1}}$$

showing how to calculate $\Delta_K(t)$ from the Burau representation. Here $\tilde{\rho}$ denotes the reduced Bureau representation and \mathbb{I}_{n-1} is the $(n - 1) \times (n - 1)$ identity matrix. Thus the Alexander

polynomial of the closed braid associated to the open braid X , i.e. $\Delta_{b(X)}(t)$, is a rescaling of the characteristic polynomial of the image of X in the reduced representation.

6. **Jones polynomial:** The famous *Jones polynomial* can be obtained from the two-variable polynomial by setting

$$V_K(t) = P_K(t, \sqrt{t} - \frac{1}{\sqrt{t}}).$$

Note that we are abusing notation by reusing the variable t here and above in $\Delta_K(t)$.

7. **Bracket polynomial.** A totally different way of defining Jones' polynomial can be derived from the bracket polynomial derived from any non-oriented knot diagram. The bracket polynomial, $\langle K \rangle = \langle K \rangle(A)$, assigns to each unoriented link diagram K a Laurent polynomial in the variable A , such that

1. If K and K' are regularly isotopic diagrams, then

$$\langle K \rangle = \langle K' \rangle.$$

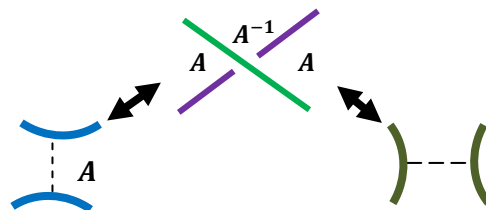
2. If $K \sqcup O$ denotes the disjoint union of K with an extra unknotted and unlinked component O , then

$$\langle K \sqcup O \rangle = (-A^2 - A^{-2}) \langle K \rangle.$$

3. $\langle K \rangle$ satisfies the following formulas

$$\langle \begin{array}{c} \diagup \\ \diagdown \end{array} \rangle = A \langle \begin{array}{c} \diagdown \\ \diagup \end{array} \rangle + A^{-1} \langle \begin{array}{c} \text{---} \\ \text{---} \end{array} \rangle \text{ and } \langle \begin{array}{c} \diagdown \\ \diagup \end{array} \rangle = A^{-1} \langle \begin{array}{c} \diagup \\ \diagdown \end{array} \rangle + A \langle \begin{array}{c} \text{---} \\ \text{---} \end{array} \rangle$$

where the small diagrams represent parts of larger diagrams that are identical except at the site indicated in the bracket. We take the convention that the notation $\begin{array}{c} \diagup \\ \diagdown \end{array}$ denotes a crossing where the curved line is crossing over the straight segment. The notation $\begin{array}{c} \diagdown \\ \diagup \end{array}$ denotes the switch of this crossing, where the curved line is undercrossing the straight segment. See Figure 6 for a graphic illustration of this relation, and an indication of the convention for choosing the labels A and A^{-1} at a given crossing.



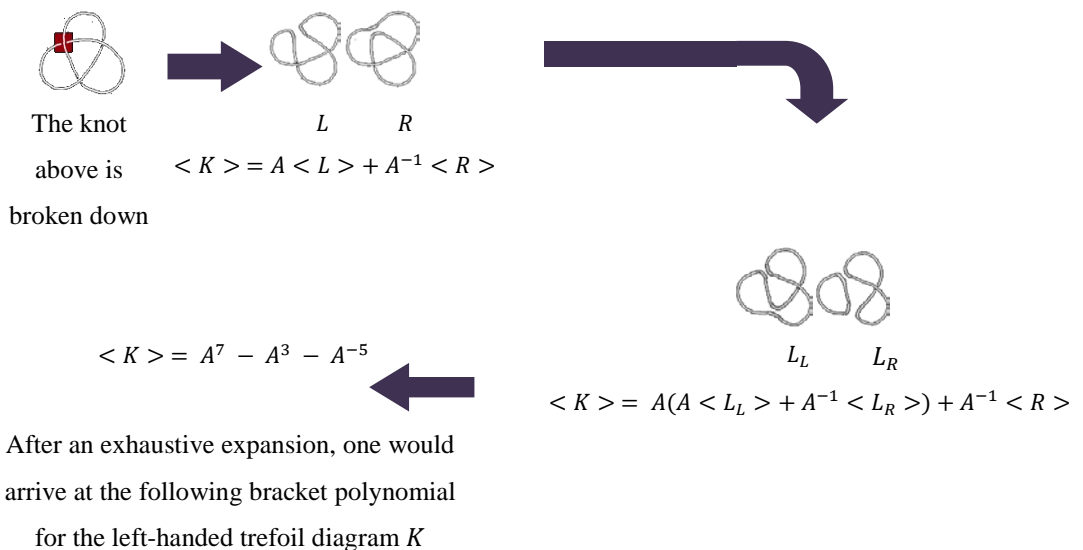
$$A^{-1}$$

$$\begin{aligned} \langle \diagup \rangle &= A \langle \diagdown \rangle + A^{-1} \langle \rangle \text{ and} \\ \langle \diagdown \rangle &= A^{-1} \langle \diagup \rangle + A \langle \rangle \end{aligned}$$

Figure 6: Bracket Smoothing

It is easy to see that Properties 2 and 3 define the calculation of the bracket on arbitrary link diagrams. The choices of coefficients (A and A^{-1}) and the value of $(-A^2 - A^{-2})$ make the bracket invariant under the types II and III Reidemeister moves (but not invariant under a type I Reidmeister move). Thus Property 1 is a consequence of the other two properties.

The idea behind the bracket polynomial is to break down a knot into a trivial link of unknots:



The bracket is invariant under regular isotopy and can be normalized to an invariant of ambient isotopy by the definition

$$(5) f_K(A) = (-A)^{-3 \cdot w(D)} \cdot \langle K \rangle (A);$$

where we chose an orientation for K , and where $w(K)$ is the sum of the crossing signs of the oriented link K . $w(K)$ is called the *writhe* of K . The convention for crossing signs is shown in Figure 7. By a change of variables one obtains the original Jones polynomial $V_K(t)$ for oriented knots and links from the normalized bracket:

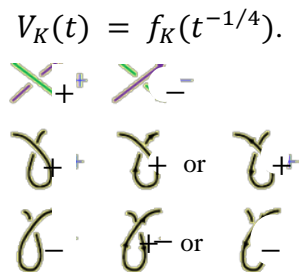


Figure 7: Crossing Signs and Curls

One useful consequence of these formulas is the following *switching formula*

$$A \langle \text{crossing} \rangle = -A^{-1} \langle \text{switched crossing} \rangle = (A^2 - A^{-2}) \langle \text{smoothed crossings} \rangle.$$

Note that in these conventions the A -smoothing of is ; while the A smoothing of is . Properly interpreted, the switching formula above says that you can switch a crossing and smooth it either way and obtain a three diagram relation. This is useful since some computations will simplify quite quickly with the proper choices of switching and smoothing. Remember that it is necessary to keep track of the diagrams up to regular isotopy (the equivalence relation generated by the second and third Reidemeister moves). Here is an example. Figure 8 shows a left-handed trefoil diagram K , an unknot diagram U and another unknot diagram U' :

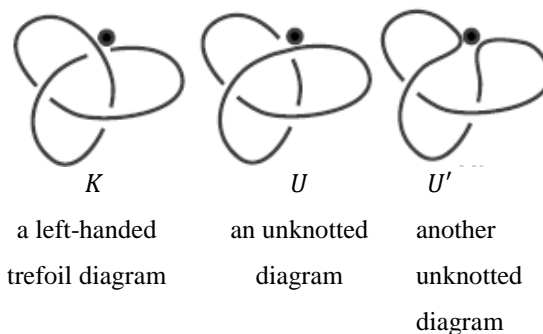


Figure 8: *Left-handed trefoil and two relatives*

Applying the switching formula, we have

$$A^{-1} \langle K \rangle - A \langle U \rangle = (A^{-2} - A^2) \langle U' \rangle$$

where $\langle U \rangle = -A^3$ and $\langle U' \rangle = (-A^{-3})^2 = A^{-6}$. It follows that

$$A^{-1} \langle K \rangle - A(-A^3) = (A^{-2} - A^2)(A^{-6})$$

which implies

$$(6) \langle K \rangle = -A^5 - A^{-3} + A^{-7}$$

This is the bracket polynomial of the trefoil diagram K . Since $w(K) = 3$, substitution of (6) into equation (5) gives the normalized polynomial

$$f_K(A) = (-A^3)^{-3} \langle K \rangle = -A^9(-A^5 - A^{-3} + A^{-7}) = A^{-4} + A^{-12} - A^{-16}.$$

In particular, we obtain $f_K(A) \neq f_K(A^{-1}) = f_{-K}(A)$. This shows that *the trefoil is not ambient isotopic to its mirror image*, a fact that is much harder to prove by classical methods.

- A lower bound for braid index.** In §4.1 we remarked that it is an open problem to determine the braid index of a knot algorithmically. However, the HOMFLY polynomial does give a remarkably useful lower bound, via a famous inequality which is known as the *Morton-Franks-Williams inequality*. It was proved

simultaneously and independently by Hugh Morton in [38] and by John Franks and Robert Williams in [23].

5.4. The Lawrence-Krammer representation of braid groups and polynomial invariants of knots

Let D be the unit disk centered at the origin in the complex plane. Fix arbitrary real numbers $-1 < p_1 < \dots < p_n < 1$. Let

$$D_n = D \setminus \{p_1, \dots, p_n\}$$

be the n -times punctured disk. The braid group B_n is the mapping class group of D_n , that is, the set of homeomorphisms from D_n to itself that act as the identity on ∂D , taken up to isotopy relative to ∂D . Let also

$$C_2 D_n$$

be the space of all unordered pairs of distinct points in D_n .

Suppose x is a point in D_n , and a is a simple closed curve in D_n enclosing one puncture point and not enclosing x . Let

$$\gamma: I \rightarrow C_2 D_n$$

be the loop in $C_2 D_n$ given by

$$\gamma(s) = \{x, a(s)\}.$$

Further, suppose τ_1 and τ_2 are paths in D_n such that $\tau_1 \tau_2$ is a simple closed curve that does not enclose any puncture points p_1, \dots, p_n . Let

$$\tau: I \rightarrow C_2 D_n$$

be the loop in $C_2 D_n$ given by

$$\tau(s) = \{\tau_1(s), \tau_2(s)\}.$$

Let

$$\Phi: \pi_1(C_2 D_n) \rightarrow \mathbb{Z}^2[q, t] = \langle q \rangle \oplus \langle t \rangle$$

be the unique homomorphism such that

$$\Phi(\gamma) = q \text{ and } \Phi(\tau) = t$$

for any γ and τ defined as above. (For a proof of the existence and uniqueness of such a homomorphism see [41]). The second homology

$$H_2(C_2D_n)$$

is a module over $\mathbb{Z}[q^{\pm 1}, t^{\pm 1}]$, where q and t act by covering transformations. Let now

$$\overline{C_2D_n}$$

be the *Lawrence–Krammer cover*, that is the connected covering space of C_2D_n whose fundamental group is the kernel of the projection map Φ . The second homology

$$H_2(\overline{C_2D_n})$$

is known to be a free $\mathbb{Z}[q^{\pm 1}, t^{\pm 1}]$ -module, of rank $\binom{n}{2}$.

Definition 5.18 ([7], [8]) *The Lawrence-Krammer representation of B_n is the induced action*

$$B_n \times H_2(\overline{C_2D_n}) \rightarrow H_2(\overline{C_2D_n})$$

of B_n on $H_2(\overline{C_2D_n})$ by $\mathbb{Z}[q^{\pm 1}, t^{\pm 1}]$ -module automorphisms. More precisely, given an element of B_n represented by a homeomorphism

$$\sigma: D_n \rightarrow D_n,$$

consider the induced action

$$\sigma: C_2D_n \rightarrow C_2D_n.$$

There is a unique lift

$$\bar{\sigma}: \overline{C_2D_n} \rightarrow \overline{C_2D_n}$$

that acts as the identity on $\partial\overline{C_2D_n}$. This induces an automorphism of $H_2(\overline{C_2D_n})$, which can be shown to respect the $\mathbb{Z}[q^{\pm 1}, t^{\pm 1}]$ -module structure.

Using Bigelow's conventions for the Lawrence–Krammer representation, one can demonstrate the following result.

Theorem 5.19 ([7], [8]) *Let $v_{i,j}$ ($1 \leq i < j \leq n$) be the generators for $H_2(\overline{C_2D_n})$. If σ_i denote the standard Artin generators of the braid group, then we get the following expression for the generator Lawrence-Krammer representation:*

$$\sigma_i \cdot v_{j,k} = \begin{cases} v_{j,k}, & \text{if } i \notin \{j-1, j, k-1, k\} \\ qv_{i,k} + (q^2 - q)v_{i,j} + (1 - q)v_{j,k}, & \text{if } i = j - 1 \\ v_{j,k}, & \text{if } i = j \neq k - 1 \\ qv_{j,i} + (1 - q)v_{j,k} - (q^2 - q)tv_{i,k}, & \text{if } i = j - 1 \\ v_{j,k+1}, & \text{if } i = k \\ -tq^2v_{j,k}, & \text{if } i = j \neq k - 1 \end{cases}.$$

Finally, Stephen Bigelow and Daan Krammer have independent proofs that

Theorem 5.20 ([7], [8], [33]) *The Lawrence–Krammer representation is faithful.*

6 Approximating Qubit Gates with Fibonacci Braiding Generators

The infinite braid group can have both one-dimensional and higher-dimensional representations. Abelian anyons correspond to the one dimensional case ($\sigma_i = e^{ia}$). Non-abelian anyons correspond to higher dimensional representations. The non-abelian anyons are characterized by D -dimensional Hilbert spaces, so that every set of N non-abelian anyons can be found in D^N orthogonal quantum states. It follows that every set of N non-abelian anyons can encode

$$N \log_2 D \text{ qubits } (D > 1).$$

Therefore, the non-abelian anyons are of particular interest for our purposes. Of these, the Fibonacci anyons satisfy the simplest rule fusion, meaning that fusion of two Fibonacci anyons delivers a single Fibonacci anyon together after a quantum state that is trivial statistical.

Recall that fusion rules are rules that determine the exact decomposition of the tensor product of two representations of a group into a direct sum of irreducible

representations. The aim of this section is to encode quantum information using only *Fibonacci anyons* (called Fibonacci because N anyons span a Hilbert space of dimension D equal to the $N + 1$ Fibonacci number).

Firstly, in order to manage quantum information, one must define the operators describing the operators “sigma” describing the Fibonacci braidings (trajectories of Fibonacci anyons). To be more specific, we will restrict ourselves to the study of cases

$$N = 3 \text{ and } N = 4.$$

Example 6.1(: $N = 3$) The Hilbert space of three Fibonacci anyons is three dimensional and spanned by the states

$$\begin{aligned} |0\rangle &= ((\bullet \bullet)_0, \bullet)_1, \\ |1\rangle &= ((\bullet \bullet)_1, \bullet)_1 \text{ and} \\ |NC\rangle &= ((\bullet \bullet)_1, \bullet)_0 \text{ (: the non-computational state).} \end{aligned}$$

Figure 9 shows the elementary braid operations σ_1 and σ_2 that can be applied to three quasiparticles.

Associated with each of these braid operations there is a matrix M which acts on the three-dimensional Hilbert space of the three Fibonacci anyons. These matrices are

$$\sigma_1 = \left(\begin{array}{cc|c} e^{-i4\pi/5} & 0 & \\ 0 & e^{i3\pi/5} & \\ \hline & & e^{i3\pi/5} \end{array} \right)$$

$$\sigma_2 = \left(\begin{array}{cc|c} \tau e^{-i\pi/5} & \sqrt{\tau} e^{-i3\pi/5} & \\ \sqrt{\tau} e^{-i3\pi/5} & -\tau & \\ \hline & & e^{i3\pi/5} \end{array} \right)$$

where the upper left 2×2 blocks of these matrices act on the computational qubit space ($|0\rangle$ and $|1\rangle$) while the lower right matrix element is the phase factor which

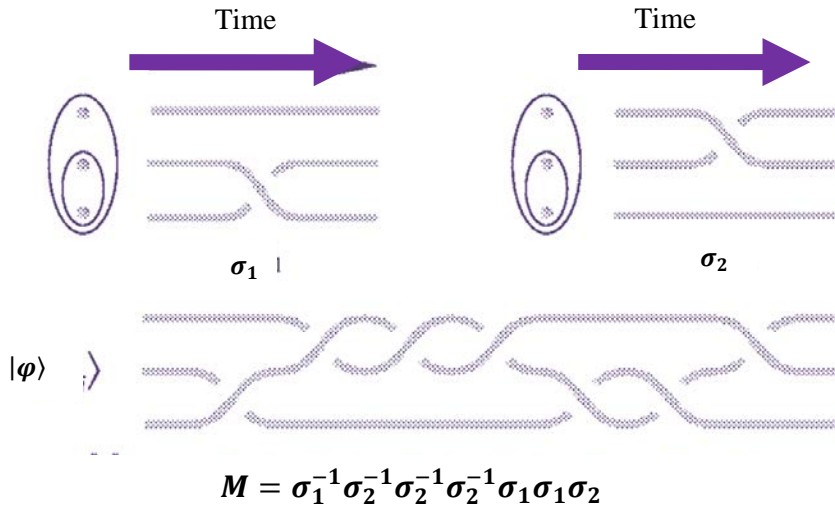


Figure 9: Elementary braid operations acting on three quasiparticles and the evaluation of a general braid in terms of these elementary operations. The solid dots represent Fibonacci anyons and the oval enclosing these dots play the same role as the parenthesis in the notation $((\bullet\bullet), \bullet)$ used in the text.

is applied to the state $|NC\rangle$. The form of these matrices is essentially fixed by certain consistency conditions dictated by fusion rules. To compute the unitary operation produced by an arbitrary braid involving three strands one then simply expresses the braid as a sequence of elementary braid operations and multiplies the corresponding matrices (σ_1 , σ_2 and their inverses) to obtain the net transformation as shown in Figure 9.

Example 6.2 (The case $N = 4$) Our aim now is to encode qubits with four Fibonacci anyons. To this end, we consider a system of four anyons whose total charge is trivial. There are two possible states (see Figure 10. a) and we associate them to the logic 0 and the logic 1. *Three Fibonacci anyons with total charge equal to 1 are enough to encode a qubit* (see Figure 10.b).

Finally, with different total charges we obtain non computational states that must be avoided (see Figure 10.c). To process a single qubit we must find the operators σ that define the braidings. In Figure 11, we give the elementary braid

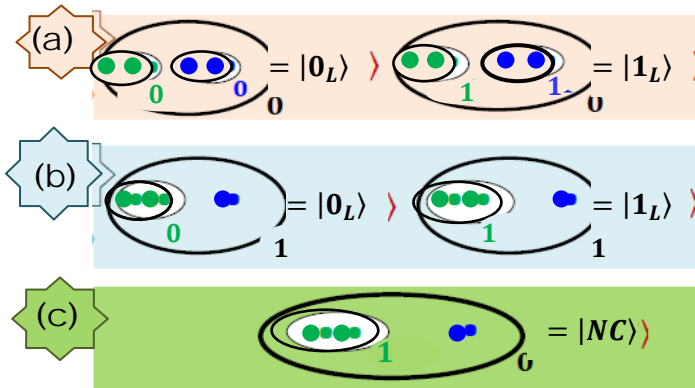


Figure 10: Coding qubits with four Fibonacci anyons

operations $\sigma_1, \sigma_2, \sigma_3$ that can be applied to four quasiparticles. The solid dots represent Fibonacci anyons and the ovals enclosing these dots play the same role as the parentheses in the notation

$$((\bullet\bullet), (\bullet\bullet))$$

used in the text. The basic braid generators of Fibonacci anyons are given by the three tables below (Figure 11). The character φ represents the gold number ($\because \varphi = (1 + \sqrt{5})/2 = 1.6180339$).

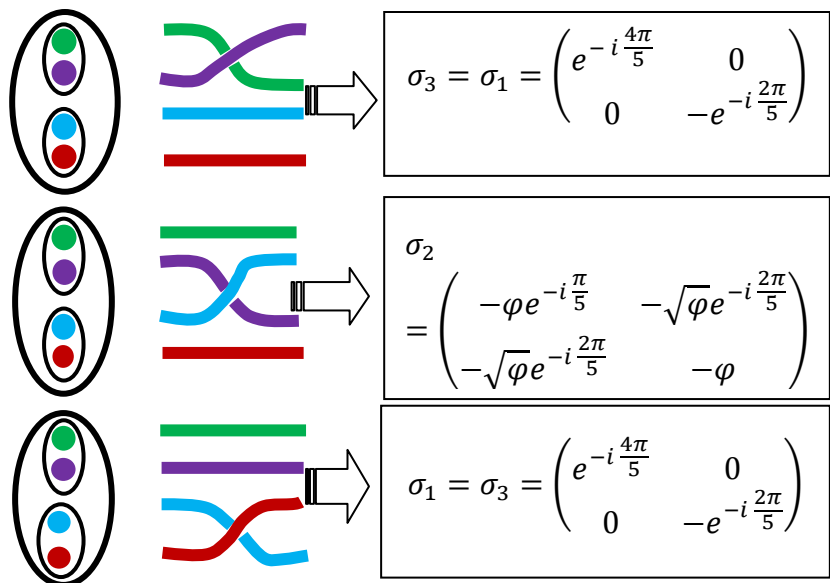


Figure 11: Fibonacci braidings

So, to compute the unitary operation produced by an arbitrary braid one then simply express the braid as a sequence of elementary braid operations and multiplies the corresponding matrices ($\sigma_1 = \sigma_3$, σ_2 and their inverses) to obtain the net transformation.

Figure 12 shows the manner in which the Fibonacci braiding shape is analyzed in such a product. Note that, for Fibonacci anyons, the elementary braidings generate an infinite group, dense in $SU(2)$. Specifically, Figure 12 shows the elementary braid operations σ_1 , σ_2 and σ_3 that can be applied to four quasiparticles. Associated with each of these braid operations there is a matrix which acts on the four dimensional Hilbert space of the four Fibonacci anyons. The form of these matrices is essentially fixed by certain consistency conditions dictated by fusion rules. To compute the unitary operation produced by an arbitrary braid involving four strands one simply expresses the braid as a sequence of elementary braid operations and multiplies the corresponding matrices (σ_1 , σ_2 , σ_3 and their inverses) to obtain the net transformation, as shown in Figure 12.

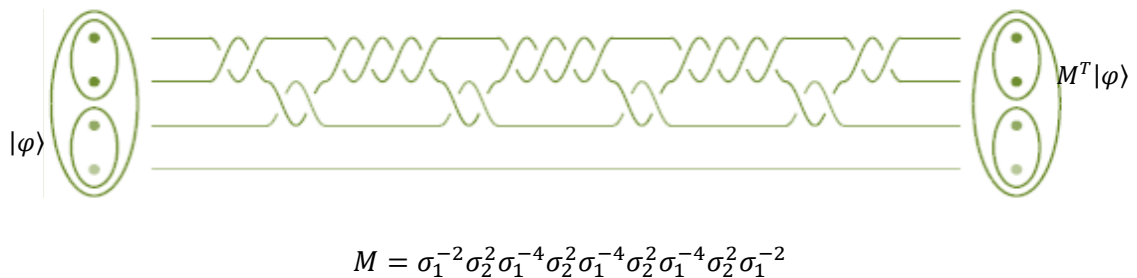


Figure 12: Elementary braid operations acting on four quasiparticles and evaluation of a general braid in terms of these elementary operations

To the purpose of universal quantum computation, we want now to approximate, at any given accuracy, any single-qubit gate using as *generators the braidings* $\sigma_1, \sigma_2, \dots, \sigma_N$. To simplify the situation, we will restrict ourselves to the case $N = 3$. As we envision carrying out a single-qubit operation on one of these encoded qubits it is convenient to consider a restricted class of braids known as weaves –

braids in which only one quasiparticle moves. It is known that any operation which can be carried out by a braid can also be carried out by a weave. Figure 13 shows the rotation vectors \vec{a} corresponding to single-qubit rotations

$$U_{\vec{a}} = \exp(i\vec{a} \cdot \vec{\sigma}/2)$$

generated by four elementary weaving operators

$$\sigma_1^2, \sigma_2^2, \sigma_1^{-2}, \sigma_2^{-2}.$$

These squared braid matrices describing weave operations in which the middle quasiparticle is woven once around either the top or bottom quasiparticle in either a clockwise or counterclockwise sense. Because the number of topologically distinct braids grows exponentially with braid length, and because the operators

$$\sigma_1^{\pm 2} \text{ and } \sigma_2^{\pm 2}$$

generate a group which is dense in $SU(2)$, the set of distinct operations which can be carried out by braids rapidly fills the space of all single-qubit rotations, as is also shown below in Figure 13.

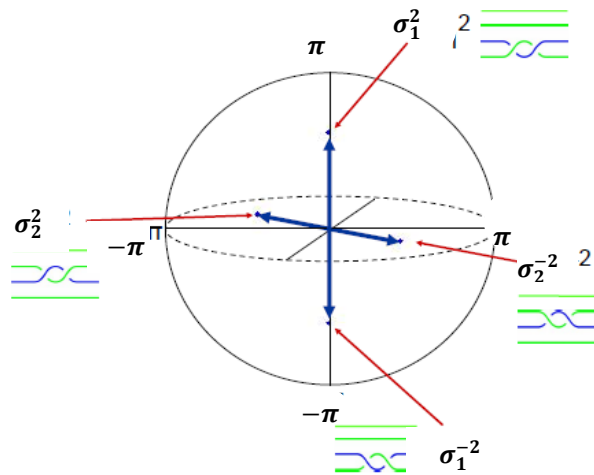
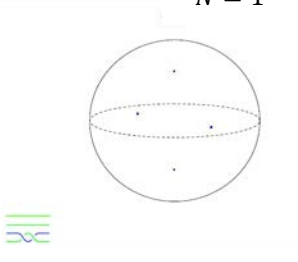
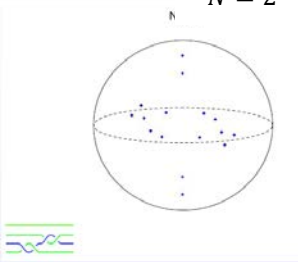
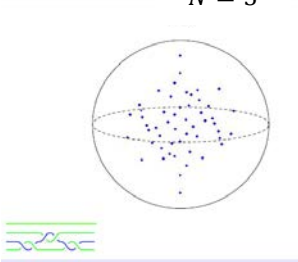
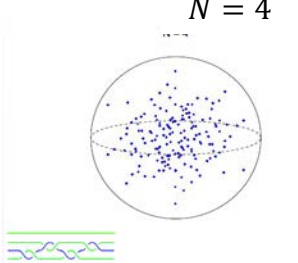
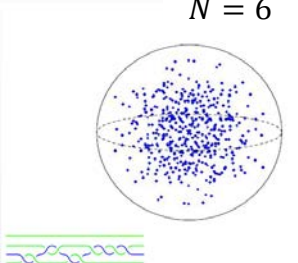
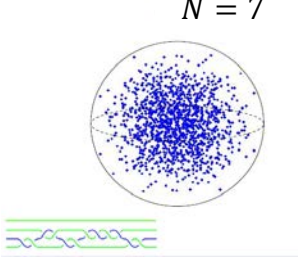
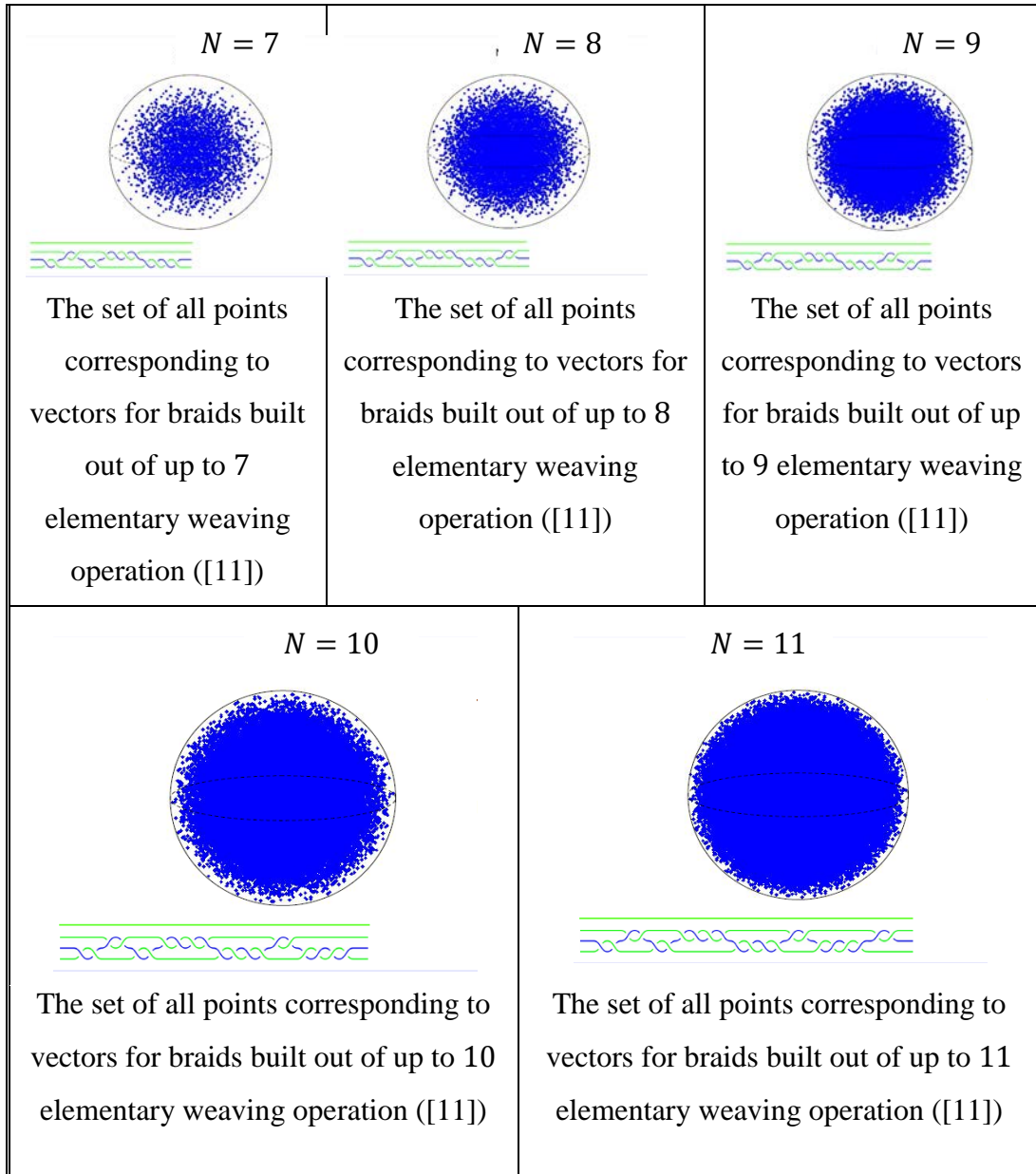


Figure 13: Weaving Operators Representation Sphere: the vectors corresponding to the four elementary weaving operators $\sigma_1^2, \sigma_2^2, \sigma_1^{-2}, \sigma_2^{-2}$ can be represented points inside a solid sphere of radius 2π

So, one can now see the set of all points of the ball corresponding to \vec{a} vectors generated by basic braiding.

<p style="text-align: center;">$N = 1$</p>  <p>The set of all points corresponding to vectors for braids built out of up to 1 elementary weaving operation ([11])</p>	<p style="text-align: center;">$N = 2$</p>  <p>The set of all points corresponding to vectors for braids built out of up to 2 elementary weaving operation ([11])</p>	<p style="text-align: center;">$N = 3$</p>  <p>The set of all points corresponding to vectors for braids built out of up to 3 elementary weaving operation ([11])</p>
<p style="text-align: center;">$N = 4$</p>  <p>The set of all points corresponding to vectors for braids built out of up to 4 elementary weaving operation ([11])</p>	<p style="text-align: center;">$N = 6$</p>  <p>The set of all points corresponding to vectors for braids built out of up to 5 elementary weaving operation ([11])</p>	<p style="text-align: center;">$N = 7$</p>  <p>The set of all points corresponding to vectors for braids built out of up to 6 elementary weaving operation ([11])</p>



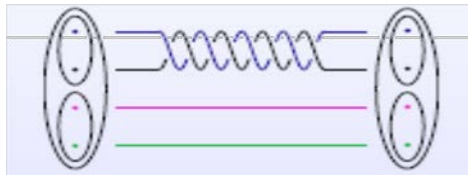
By carrying out *Bruce Force (BF) searches* over braids with up to 46 elementary braid operations we typically find braids which approximate a desired target gate to a distance of $\sim 10^{-3}$. Recall that a BF search allows to find the best weave (: braid in which only one quasiparticle moves) of a given length L to approximate every target gate. Any operation which can be carried out by a braid can also be carried out by a weave. The number of possible braids grows

exponentially as $3^{L/2}$, while the time required is exponential in the length and calculations become cumbersome for $L > 40$. Although the accuracy grows exponentially as $3^{-L/6}$, the BF approach optimal solution is reached very slowly.

Using weaves we can approximate every single-qubit gate choosing the best braid among the $3^{L/2}$ possibilities.

Example 6.3

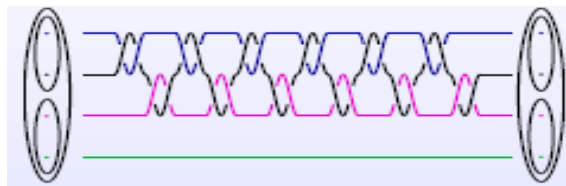
i. Target Gate: $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$



$$L = 8,$$

$$\tilde{Z}_8 \cong \begin{pmatrix} 0.31 + 0.95 i & 0 \\ 0 & 0.31 - 0.95 i \end{pmatrix}$$

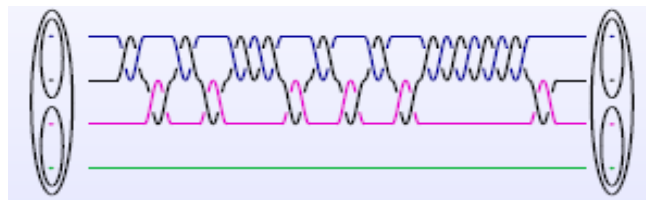
ii. Target Gate: $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$



$$L = 24,$$

$$\tilde{Z}_{24} \cong \begin{pmatrix} 0.0234 - 0.9997 i & 0.0060 + 0.0020 i \\ -0.006 + 0.002 i & 0.0234 + 0.9997 i \end{pmatrix}$$

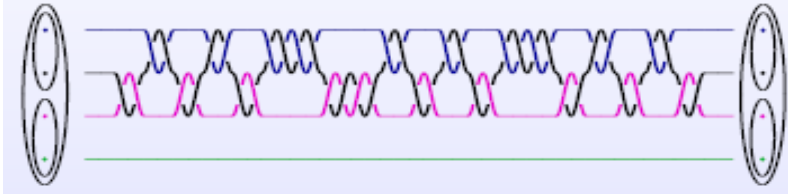
iii. Target Gate: $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$



$$L = 32,$$

$$\tilde{Z}_{32} \cong \begin{pmatrix} 0.004 + 0.99997 i & 0.004 - 0.003 i \\ -0.004 - 0.003 i & 0.004 + 0.99997 i \end{pmatrix}$$

iv. Target Gate: $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$



$$L = 44,$$

$$\tilde{Z}_{44} \cong \begin{pmatrix} -i & o(10^{-3}) \\ o(10^{-3}) & i \end{pmatrix}$$

7 Quantum Hashing with the Icosahedral Group

The brute force search is inefficient for long braids because it samples the whole $SU(2)$ space with almost equal weight. To get a faster algorithm we must enhance the sampling near the target gate we want to approximate. In this way we can get a much faster algorithm that finds good approximations for arbitrary $SU(2)$ gates (but in general not the optimal one).

The question is thus the following.

Open Question 7.1 *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 Section, we explore topological quantum hashing with the finite **icosahedral group \mathcal{T}** 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 (Figure 14).

The icosahedral rotation group \mathcal{T} 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

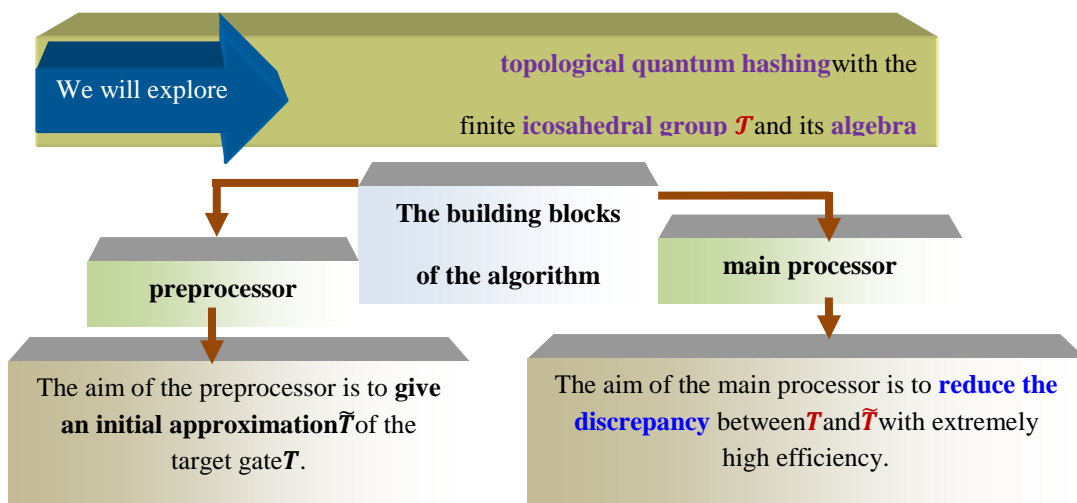


Figure 14: Topological Quantum Hashing and Icosahedral Group

studies of $SU(2)$ lattice gauge theories, and this motivated us to apply the icosahedral group representation in the braid construction. The icosahedral rotation group \mathcal{T} 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, and this motivated us to apply the icosahedral group representation in the braid construction. \mathcal{T} 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

$$\mathcal{T} = \{g_0, g_1, \dots, g_{59}\},$$

where

$$g_0 = e$$

is the identity element. Thanks to the homomorphism between $SU(2)$ and $SO(3)$, we start by associating a 2×2 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 and neglecting an overall phase. In this way, we obtain an approximate representation in $SU(2)$ of the icosahedral group

$$\tilde{\mathcal{T}}(N) = \{\tilde{g}_0(N), \tilde{g}_1(N), \dots, \tilde{g}_{59}(N)\}.$$

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.15 for an example).

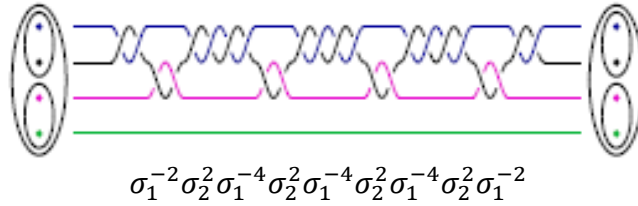


Figure 15: (color online) Approximation to the $-iX$ gate (an element of the icosahedral group) in terms of braids of the Fibonacci anyons of length $L = 24$ in the graphic representation. In this example the error is 0.0031

We point out that the 60 elements of $\tilde{\mathcal{T}}(N)$ (for any finite N) do not close any longer the composition laws of \mathcal{T} ; in fact, they form a *pseudo-group*, not a group, isomorphic to \mathcal{T} only in the limit $N \rightarrow \infty$ (Figure 16). 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{T} can be linked to the Wigner-Dyson distribution, which we will discuss later.

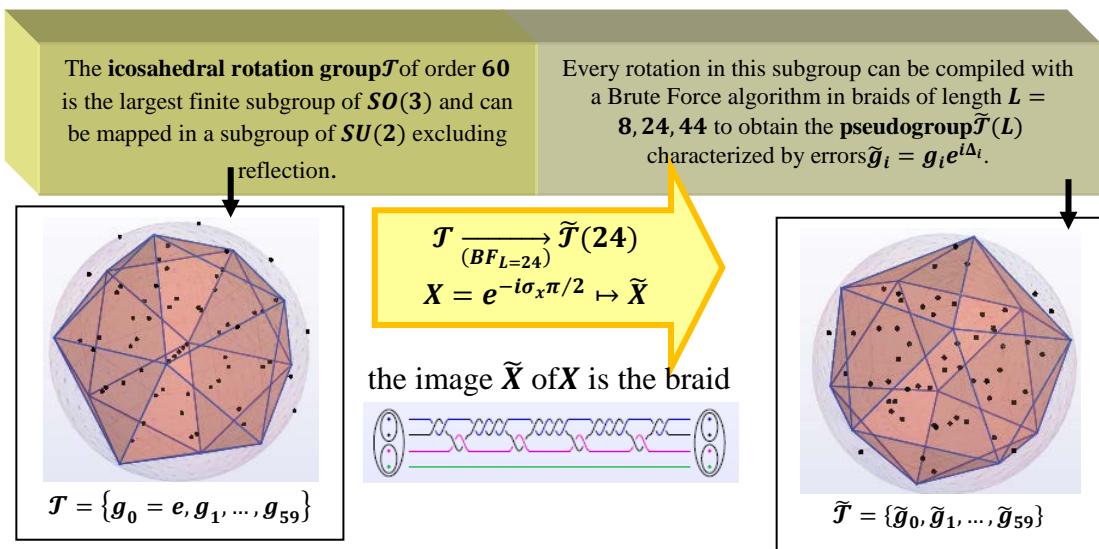


Figure 16: Icosahedral Group and Icosahedral Pseudogroup (Burrello et al.)

Using the pseudo-group structure of $\tilde{\mathcal{T}}$, 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 $\tilde{g}_{i_1}(L)\tilde{g}_{i_2}(L)\dots\tilde{g}_{i_n}(L)$ of $n \geq 2$ elements of $\tilde{\mathcal{T}}(L)$ of length L and multiplying them by the matrix $\tilde{g}_{i_{n+1}}(L) \in \tilde{\mathcal{T}}(L)$ such that $g_{i_{n+1}} = g_{i_n}^{-1}\dots g_{i_2}^{-1}g_{i_1}^{-1}$. In this way we generate all the possible combinations of $n + 1$ elements of \mathcal{T} whose result is the identity, but, thanks to the errors that characterize the braid representation $\tilde{\mathcal{T}}$, we obtain 60^n small rotations in $SU(2)$, corresponding to braids of length $(n + 1)L$.

7.1. 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 $[\tilde{\mathcal{T}}(l)]^m$ (of length $m \times l$) that best approximates it. Thus we obtain a starting braid

$$\tilde{\mathcal{T}}_0^{l,m} = \tilde{g}_{j_1}(l) \tilde{g}_{j_2}(l) \dots \tilde{g}_{j_m}(l)$$

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 \tilde{g} . Due to these random errors the set $[\tilde{\mathcal{T}}(l)]^m$ of all the products $\tilde{g}_{j_1}\tilde{g}_{j_2}\dots\tilde{g}_{j_m}$ 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 $\mathcal{S}(L, n)$ to efficiently reduce the error in $\tilde{\mathcal{T}}_0^{l,m}$. Multiplying $\tilde{\mathcal{T}}_0^{l,m}$ by all the elements of $\mathcal{S}(L, n)$, we generate 60^n possible braid representations of T :

$$\tilde{\mathcal{T}}_0^{l,m} = \tilde{g}_{i_1} \tilde{g}_{i_2} \cdots \tilde{g}_{i_n} \tilde{g}_{i_{n+1}}.$$

Among these braids of length $(n + 1)L + ml$, we search the one which minimizes the distance with the target gate T . This braid, $\tilde{\mathcal{T}}_{L,n}^{l,m}$, is the result of our algorithm.

Figure 17 shows the distribution of final errors for 10000 randomly selected target gates obtained with a preprocessor of $l = 8, m = 3$ and a main processor of $L = 24$ and $n = 3$.

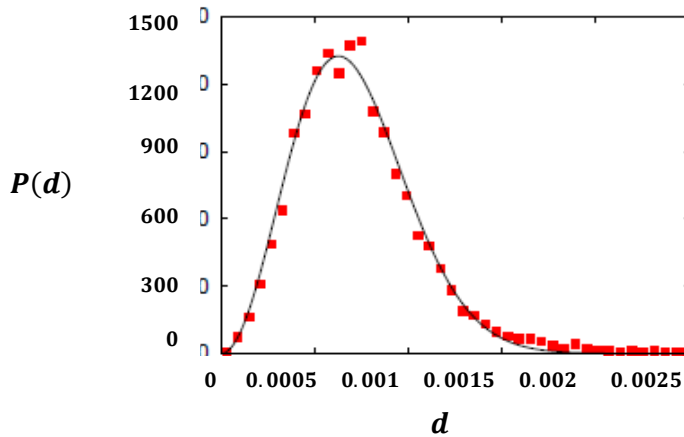


Figure 17: Probability distribution of d in 10000 random tests using the icosahedral group approach with a preprocessor of $l = 8$ and $m = 3$ and a main processor of $L = 24$ and $n = 3$.

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 $[\tilde{\mathcal{T}}(8)]^3$, the preprocessor selects a

$$\tilde{\mathcal{T}}_0^{8,3} = \tilde{g}_{p_1}(8) \tilde{g}_{p_2}(8) \tilde{g}_{p_3}(8),$$

which minimizes the distance to T to 0.038. Applying now the main processor, the best rotation in $\mathcal{S}(24, 3)$, that corrects $\tilde{\mathcal{T}}_0^{8,3}$ is given by a

$$\tilde{g}_{q_1}(24) \tilde{g}_{q_2}(24) \tilde{g}_{q_3}(24) \tilde{g}_{q_4}(24),$$

where $g_{q_4} = g_{q_3}^{-1} g_{q_2}^{-1} g_{q_1}^{-1}$. The resulting braid is then represented by

$$\begin{aligned} \tilde{\mathcal{T}}_{24,3}^{8,3} &= \tilde{g}_{p_1}(8) \tilde{g}_{p_2}(8) \tilde{g}_{p_3}(8) \tilde{g}_{q_1}(24) \tilde{g}_{q_2}(24) \tilde{g}_{q_3}(24) \tilde{g}_{q_4}(24) \\ &= \begin{pmatrix} -0.0004 + 1.0000i & -0.0007 - 0.0005i \\ 0.0007 - 0.0005i & -0.0004 - 1.0000i \end{pmatrix} e^{4\pi i/5} \end{aligned}$$

for the special set of p 's and q 's and, apart from an overall phase, the final distance is reduced to 0.00099 (see Figure 15).

7.2. 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 $\tilde{\mathcal{T}}(N)$ for braids of length N . One can write $\tilde{g}_i = g_i e^{i\Delta_i}$, where Δ_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 \tilde{g}_i that approximate $g_i g_j \cdots g_{n+1} = e$, one has

$$\tilde{g}_i \tilde{g}_j \cdots \tilde{g}_{n+1} = g_i e^{i\Delta_i} g_j e^{i\Delta_j} \cdots g_{n+1} e^{i\Delta_{n+1}} = e^{iH_n},$$

where H_n , related to the accumulated deviation, is

$$H_n = g_i \Delta_i g_i^{-1} + g_i g_j \Delta_j g_j^{-1} g_i^{-1} + \cdots + g_i g_j \cdots g_n \Delta_n g_n^{-1} \cdots g_j^{-1} g_i^{-1} + \Delta_{n+1} + O(\Delta^2).$$

The natural conjecture is that, *for a long enough sequence of matrix 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.

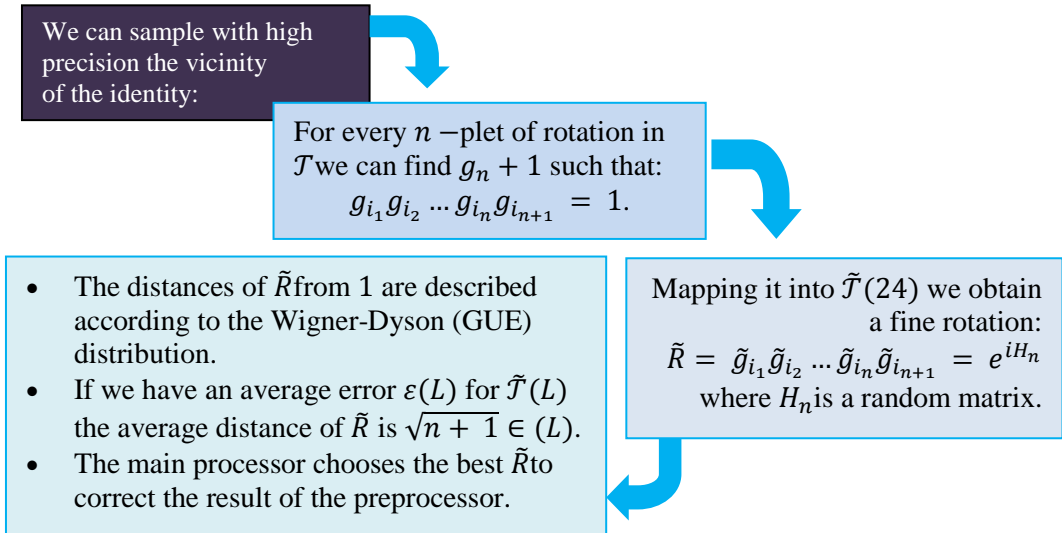


Figure 18: The main processor (Burrello et al.)

A direct consequence is that the distribution of the eigenvalue spacing s obeys the *Wigner-Dyson form* [35],

$$P(s) = \frac{32}{\pi^2 s_0} \left(\frac{s}{s_0}\right)^2 e^{-(4/\pi)(s/s_0)^2},$$

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 Figure 19).

One can show that the final error of $\tilde{\mathcal{T}}_{L,n}^{l,m}$ also follows the Wigner-Dyson distribution (as illustrated in Figure 17) with an average final distance $f \sim 60^{n/3} / \sqrt{n+1}$ times smaller than the average error of $\tilde{\mathcal{T}}_0^{l,m}$, 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.

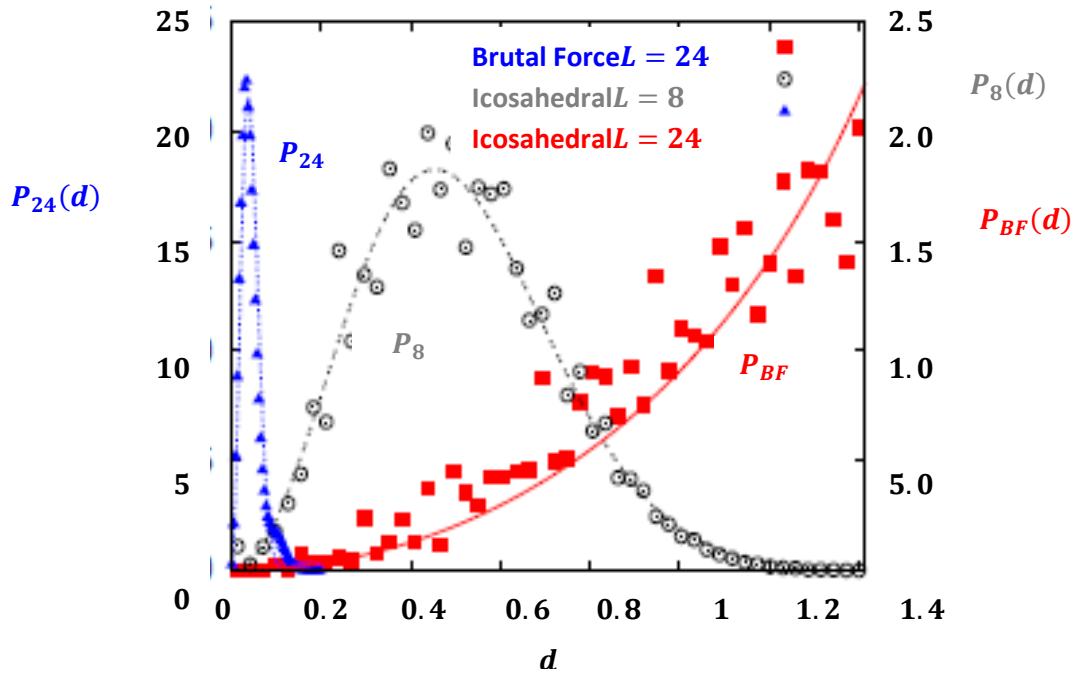


Figure 19: Probability distribution of the distance d to the targeted identity matrix in the set of nontrivial braids that one samples in different algorithms. $P_{BF}(d)$ of the brute-force search (red solid squares) roughly follows $(4/\pi)d^2/\sqrt{1 - (d/2)^2}$, reflecting the three-sphere nature of the unitary matrix space (three independent parameters apart from an unimportant phase). In the pseudo icosahedral group approach ($n = 4$), distributions for $L = 8$ (P_8 , black empty circles) and $L = 24$ (P_{24} , blue solid triangles) agree very well with the energy-level-spacing distribution of the unitary Wigner-Dyson ensemble of random matrices, $P_L(d) = (32/\pi^2)(d^2/d_L^3)\exp[-(4/\pi)(d/d_L)^2]$. $P_L(d)$ differ only by their corresponding average d_L (not a fitting parameter), which decays exponentially as L increases. Note that $P_{24}(d)$ is roughly ten-times sharper and narrower than $P_8(d)$.

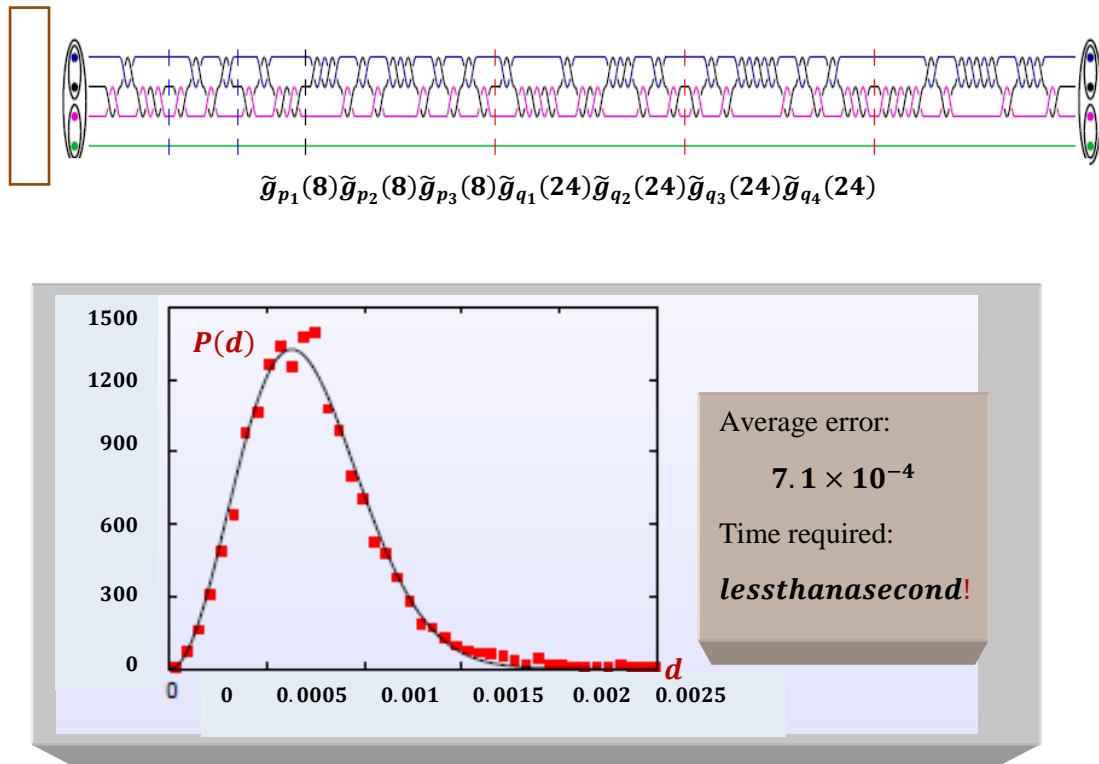


Figure 20: Results (maximal length= 120) ([15])

References

- [1] J.W. Alexander, Topological invariants of knots and links, *Trans.Amer.Math.Soc.*, **20**, (1923), 275-306.
- [2] J.W. Alexander, A lemma on a system of knotted curves, *Proc. Nat. Acad. Sci. USA*, **9**, (1923), 93-95.
- [3] D. Aharonov and M. Ben-Or, Fault-Tolerant Quantum Computation With Constant Error, *STOC*, (1997), 176-188.
- [4] D.J. Bernstein, *Introduction to Post-Quantum Cryptography*, Introduction to Daniel J. Bernstein, Johannes Buchmann, Erik Dahmen (editors), *Post-quantum cryptography*, Springer, Berlin, 2009.

- [5] S. Bigelow, The Burau representation is not faithful for $n = 5$, *Geom. Topol.*, **3**, (1999), 397-404.
- [6] S. Bigelow, Braid groups are linear, *J. Amer. Math. Soc.*, **14**, (2001), 471-486.
- [7] S. Bigelow, Representations of Braid Groups, *ICM 2002*, **II**, 37-45. Available online at <http://www.mathunion.org/ICM/ICM2002.2/Main/icm2002.2.0037.0046.ocr.pdf>
- [8] S. Bigelow, The Lawrence–Krammer representation, Topology and geometry of manifolds, (Athens, GA, 2001), *Proc. Sympos. Pure Math.* (AMS, Providence, RI), **71**, (2003), 51-68.
- [9] J.S. Birman and T.E. Brendle, *Braids: A survey*. Available online at <http://www.math.columbia.edu/~jb/Handbook-21.pdf>
- [10] N.E. Bonesteel, L. Hormozi and G. Zikos, Braid Topologies for Quantum Computation, *Phys. Rev. Lett.* **95**, (2005), 140503.
- [11] N.E. Bonesteel, L. Hormozi and G. Zikos, Quantum computing with non-Abelian quaziparticles, *International Journal of Modern Physics*, **B, 21**, (8-9), (2007), 1372-1378. Available online at <http://magnet.fsu.edu/~bonestee/papers/wurzburg.pdf>
- [12] N. Bourbaki, *Groups et algèbres de Lie*, Chapitres 4, 5, 6, Hermann, Paris, 1968.
- [13] J.L. Brylinski and R. Brylinski, Universal quantum gates, in *Mathematics of Quantum Computation*, Chapman & Hall/CRC Press, Boca Raton, Florida, (edited by R. Brylinski and G. Chen), (2002).
- [14] W. Burau, Über Zopfgruppen und gleichsinnig verdrillte Verkettungen, *Abh. Math. Sem. Ham.*, **II**, (1936), 171-178.
- [15] M. Burrello, H. Xu, G. Mussardo and X. Wan, Quantum Hashing with the Icosahedral Group, arXiv:0903.1497v2 [quant-ph], (23 Mar. 2010).
- [16] A.R. Calderbank and P.W. Shor, Good quantum error-correcting codes exist, *Phys. Rev.A*, **54**, (1996), 1098-1105.

- [17] G. Chen, L. Kauffman and S. Lomonaco, *Mathematics in Quantum Computation and Quantum Technology*, Chapman & Hall/CRC, 2007.
- [18] D.P. DiVincenzo, Quantum Computation, *Science*, **270**, (5234), (1995), 255–261. Available online at
Bibcode1995Sci...270..255D. doi:10.1126/science.270.5234.255
- [19] M. I. Dyakonov, Is Fault-Tolerant Quantum Computation Really Possible? *in: Future Trends in Microelectronics. Up the Nano Creek*, Université Montpellier (2006-10-14), Wiley, pp. 4–18 (edited by S. Luryi, J. Xu, and A. Zaslavsky). Available on line at arXiv:quant-ph/0610117
- [20] H. Dye, Unitary solutions to the Yang-Baxter equation in dimension four, arXiv:quant-ph/0211050 v2 22 January 2003.
- [21] R.J. McEliece, *A public-key cryptosystem based on algebraic coding theory*, *Jet Propulsion Laboratory DSN Progress Report* **42–44**, 114–116.
- [22] M. Freedman, A. Kitaev, M. Larsen and Z. Wang, Topological Quantum Computation, *Bull. Amer. Math. Soc. (N.S.)*, **40**(1), (2002), 31-38. Available online at doi:10.1090/S0273-0979-02-00964-3
- [23] J. Franks and B. Williams, Braids and the Jones-Conway polynomial, *Trans. AMS*, **303**(1), (1987), 97-108.
- [24] P. Freyd, D. Yetter, J. Hoste, W. Lickorish, K. Millett and A. Ocneanu, A new polynomial invariant of knots and links, *Bull. Amer. Math. Soc. (NS)*, **12**(2), (1985), 183-312.
- [25] D. Gottesman, The Heisenberg Representation of Quantum Computers, arXiv:quant-ph/9807006v1 (1 Jul, 1998).
- [26] http://en.wikipedia.org/wiki/Density_matrix
- [27] http://en.wikipedia.org/wiki/Lawrence%E2%80%93Krammer_representation
- [28] V. Jones, Hecke algebra representations of braid groups and link polynomials, *Annals of Math.*, **126**, (1987), 335-388.
- [29] L.H. Kauffman, State models and the Jones polynomial, *Topology*, **26**(3), (1987), 395-407.

- [30] L. Kauffman and S. Lomonaco, *Topological Quantum Information Theory*. Available online at <http://homepages.math.uic.edu/~kauffman/Quanta.pdf>
- [31] E. Knill, R. Laflamme and W.H. Zurek: Resilient quantum computation, *Science* **279**, (5349), (1998), 342-345.
- [32] H. Kobayashi and F.L. Gall, *Dihedral Hidden Subgroup Problem: A Survey*, *Information and Media Technologies*, **1**(1), (2006), 178-185. Available online at http://www.jstage.jst.go.jp/article/imt/1/1/1_178/_article
- [33] D. Krammer, Braid groups are linear, *Ann. Math.*, **155**, (2002), 131-156.
- [34] D. D. Long, and M. Paton, The Burau representation is not faithful for $n \geq 6$, *Topology*, **32**(2), (1993), 439-447.
- [35] M. Mehta, *Random Matrices*, 2nd ed., Academic, San Diego, 1991.
- [36] J. A. Moody, The Burau representation of the braid group B_n is unfaithful for large n , *Bull. Amer. Math. Soc. (N.S.)*, **25**(2), (1991), 379-384.
- [37] J.A. Moody, The faithfulness question for the Burau representation, *Proc. AMS* **32**, (1993), 439-447.
- [38] H. Morton, Seifert circles and knot polynomials, *Math. Proc. Cambr. Phil. Soc.*, **99**, (1986), 107-109.
- [39] M.A. Nielsen and I.L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, 2000.
- [40] J. Park, A personal note on the Yang-Baxter equation and braided bialgebra. Available online at http://mathsci.kaist.ac.kr/~jinhyun/note/yang-baxter/yb_braided_bialgebra.pdf
- [41] L. Paoluzzi and L. Paris, A note on the Lawrence-Krammer-Bigelow representation, *Algebr. Geom. Topol.* **2**, (2002), 499-518.
- [42] J. Preskill, Quantum Computation, *Lecture Notes for Physics*, **219**, 2004. Available online at www.theory.caltech.edu/~preskill/ph219/topological.pdf
- [43] K. Reidmeister, Elementare Begründung der Knotentheorie, *Abh. Math. Sem. Univ. Hamburg*, **5**, (1926), 24-32.

- [44] P.W. Shor: Scheme for reducing decoherence in quantum computer memory, *Phys. Rev.A*, **52**,(1995),2493-2496.
- [45] P.W. Shor, Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, *SIAM Journal of Computing*, **26**, (1997), 1484-1509.
- [46] C.C. Squier, The Burau representation is unitary, *Proc. AMS*, **90**(2), (1984), 199-202.
- [47] A. Steane, Error Correcting Codes in Quantum Theory, *Phys. Rev. Lett.*, **77**, (1996), 793-797.
- [48] V. Turaev, Faithful representations of the braid groups, Séminaire Bourbaki 1999-2000, *Astérisque*, **276**, (2002), 389-409.
- [49] P. Vogel, Representation of links by braids: A new algorithm, *Comment. Math. Helvetici*, **65**(1), (1990), 104-113.
- [50] N.R. Wallach, *A variety of solutions to the Yang-Baxter equation*, in *Advances in Geometry*, Progress in mathematics 172, Boston, Birkhauser 1999, pp.391-399 (edited by J.L Brylinski et al). Available online at <http://math.ucsd.edu/~nwallach/YANGBAXF.pdf>
- [51] H. Wenzl, *Representations of Hecke algebras and subfactors*, PhD thesis, Univ. of Pennsylvania, 1985.
- [52] S. Yamada, The minimal number of Seifert circles equals the braid index of a link, *Invent. Math.*, **89**(2), (1987), 347-356.