Quantum Information Science Notes

Andy Lutomirski

2008-05-13 12:33:37 -0400 (Tue, 13 May 2008)

The class is done, so there won't be any more regular updates. If you find an error, though, feel free to email me at luto@mit.edu, and I'll fix it.

Contents

1 Intro to the class

This is 6.443J, 8.371J, 18.409, and MAS.865.

We'll discuss quantum algorithms (Shor's, Grover's, etc.), crypto (key distribution and signatures), quantum communication (superdense coding, etc.), error correction, alternative models of computation (quantum computation by teleportation or measurement), etc. We will not cover implementations (these are covered more in 8.422, 6.73? (Orlando's), Dave Cory's). We will also not cover complexity theory (covered in the fall in Scott Aaronson's class).

There will be a project (20% presentation, 40% paper) as well as four problem sets (40%).

2 Quantum Error Correction

We will cover quantum operations, the criteria for quantum error correction, the CSS codes, the stabilizer formalism, and non-additive codes using the CWS (codeword-stabilized codes) framework.

2.1 Quantum operations

2.1.1 Density matrices

This will provide to background to understand errors. We need to understand what happens when we take a quantum state, put it into a black box, and receive output. This system could be a measurement, an operation, something that throws out the state and replaces it entirely, for example. It could be an adversary or a friend. We will only describe a discrete-time model.

Using pure states is insufficient to describe such systems because they could add randomness.

Pure states are of the form $a|0\rangle + b|1\rangle$. Unitary operations act by $U|\psi\rangle = a'|0\rangle + b'|1\rangle$. If, on the other hand, we throw away qubits, we might have $|\psi_{AB}\rangle = a|00\rangle + b|01\rangle + c|10\rangle$ and throw away B's state. This results in the partial trace over B. One way to think of it is to imagine B measuring the bit.

In a simple example, if we start with $\sqrt{\frac{3}{4}}|00\rangle + \sqrt{\frac{1}{4}}|11\rangle$, then A gets $\sqrt{\frac{3}{4}}|0\rangle \overset{\text{"or"}}{\oplus} \sqrt{\frac{1}{4}}|1\rangle$. If B instead applies a Hadamard gate and measures, then the (pre-measurement) state is $\left(\sqrt{\frac{3}{8}}|0\rangle + \sqrt{\frac{1}{8}}|1\rangle\right)|0\rangle + \left(\sqrt{\frac{3}{8}}|0\rangle - \sqrt{\frac{1}{8}}|1\rangle\right)|1\rangle$. Then A has $\left(\sqrt{\frac{3}{8}}|0\rangle + \sqrt{\frac{1}{8}}|1\rangle\right) \oplus \left(\sqrt{\frac{3}{8}}|0\rangle - \sqrt{\frac{1}{8}}|1\rangle\right)$. A must not be able to tell these mixed states apart (by any means at all).

The density matrix is a tool for tracking statistical mixtures, defined by

$$\rho = \sum_k |\psi_k\rangle\langle\psi_k| \tag{1}$$

where the $|\psi_k\rangle$are unnormalized. For example,

$$|0\rangle\langle 1| = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

and both mixtures give $\rho = \frac{1}{4}\begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}$.

A matrix ρ is a density matrix iff $Tr(\rho) = 1$ and $\rho \succeq 0$. The recipe $\rho = \sum_k p_k|\psi_k\rangle\langle\psi_k|$ always gives a density matrix (for normalized states and a probability distribution). Also, $\forall$ density matrixes $\rho.\rho = \sum_k p_k|\psi_k\rangle\langle\psi_k|$ for some $\{p_k\}, \{|\psi_k\rangle\}$ (the spectral decomposition).

Unraveling lemma Let $\rho = \sum_k p_k|\psi_k\rangle\langle\psi_k|$. Then $\rho = \rho' = \sum_k q_k|\phi_k\rangle\langle\phi_k|$ if $\sqrt{p_k}|\psi_k\rangle = \sum_j u_{kj}q_j|\phi_k\rangle$ with u unitary.

2.1.2 System-environment model

Suppose we have a system ρ and an environment $|e\rangle$. We have a black box which acts by a unitary matrix u on $|\psi\rangle$ and $|e\rangle$, producing ρ_{out} and some other output which is measured and then forgotten. Then the output can be written in terms of partial projections $\rho = \oplus_k\langle e_k|u|e\rangle|\psi\rangle$ where the $|e_k\rangle$are the possible outputs of the environment.

(In full generality,

$$\rho = \sum_k E_k\rho_{in}E_k^\dagger, \tag{2}$$

where, in this case $E_k = \langle e_k|u|e\rangle$.)

As an example, suppose the input state is controlles a rotation of the environment (with the environment written first). Then $U|00\rangle = |00\rangle$ and $U|01\rangle = \left[\cos\frac{\theta}{2}|0\rangle + \sin\frac{\theta}{2}|1\rangle\right]|1\rangle_{sys}$. $E_0 = |0\rangle\langle 0| + \cos\frac{\theta}{2}|1\rangle\langle 1|$ and $E_1 = \sin\frac{\theta}{2}|1\rangle\langle 1|$. Using eq. 2, $E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{p} \end{bmatrix}$ and $E_1 = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{1-} \end{bmatrix}$, the operation elements. The final result is $\mathcal{E}(\begin{bmatrix} a & b \\ c & d \end{bmatrix}) = \begin{bmatrix} a & \sqrt{p}b \\ \sqrt{p}c & d \end{bmatrix}$, which is phase damping (and is dephasing as $p \to 0$).

As another example, we could put a CNOT at the end (environment controlling input), which allows energy exchange as well. The result is $\mathcal{E}(\begin{bmatrix} a & b \\ c & d \end{bmatrix}) = \begin{bmatrix} 1-pd & b\sqrt{p} \\ c\sqrt{p} & pd \end{bmatrix}$. This is called amplitude damping.

2.1.3 Quantum operations definition

A quantum operation takes a density matrix and does something to it. It must have several properties.

Trace preserving We need the output to be legitimate, so $Tr[\mathcal{E}(\rho)] = 1$.

Linear $\mathcal{E}(\sum_k p_k\rho_k) = \sum_k p_k\mathcal{E}(\rho_k)$

Completely positive $\rho \geq 0 \implies \mathcal{E}(\rho) \geq 0$ and $(I \otimes \mathcal{E})\rho \geq 0$

As an example, transposition ($\mathcal{E}(\rho) = \rho^T$) meets the first two requirements. But if we apply it to the second half of $|00\rangle + |11\rangle$ then we get $\mathcal{E}(\rho) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \not\geq 0$. This map is a test for entanglement: if you apply to a state and get a negative eigenvalue, then the state is entangled. This is not a necessary condition for more than two qubits. (We say that this state has a negative partial transpose.)

Question: how many parameters describe a one-qubit quantum operation? There are 12 real degrees of freedom.

2.1.4 Operator sum representation

Let $\mathcal{E}(\rho) = \sum_k E_k \rho E_k^\dagger$ where $\sum_k E_k^\dagger E_k = I$. Then $\mathcal{E}(\rho)$ is a TCP operation and all TCP operations can be written in this form. (In d dimensions – $d \times d$ density matrix – we need up to d^2 operation elements.)

3 Coding

Suppose we put a message through a channel and we want to recover the original message. The simplest classical code is to repeat the message three times. We encode (repeat the message), then send it, then measure the syndrome, and then perform recovery.

All of the cases we consider involve a memoryless channel. There is an analog to the nonwhite Gaussian noise channel, but we won't cover it in this class.

In quantum mechanics, we cannot add redundancy for two reasons. First, we can't measure the state it sent – it collapses. Second, errors are continuous. Third, there's the no-cloning theorem.

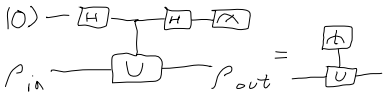
3.1 Classical coding

The binary symmetric channel sends one bit and flips it with probability p.

A classical $[n, k, d]$ code is a set of 2^k n-bit codes with minimum distance d. The distance we use in classical codes is the Hamming distance, which is the weight of the exclusive or of two strings.

3-way repetition with majority decoding has a probability of error $3p^2 - 2p^3 = O\left(p^2\right)$.

3.2 Operator measurements



Given unitary U with eigenvalues ± 1 and eigenstates $|u_\pm\rangle$, the act of "measuring U" is shown above. If $|\psi\rangle = c_0|u_+\rangle + c_1|u_-\rangle$, then the result is $|0\rangle c_0|u_+\rangle + |1\rangle c_1|u_-\rangle$.

3.3 Quantum coding

In quantum language, we can describe this channel by $\mathcal{E}(\rho) = (1-p)\,\rho + pX\rho X$.

An $[[n, k, d]]$ quantum code is a k-qubit sized subspace of an n-qubit sized Hilbert space with distance d (presently undefined). We could also talk about an $((n, K, d))$ code where the code space is K-dimensional.

The analogue of the repetition code is $|0\rangle \to |000\rangle$ and $|1\rangle \to |111\rangle$, and we can build it out of two CNOTs. This is *not* a repetition code (think about its action on other inputs). In the classical case, we did a majority vote to find errors, but we can't do that directly in quantum mechanics. Instead we use operator measurements. The possible outputs of the channel are $a|001\rangle + b|110\rangle$ w.p. $p\,(1-p)^2$. Consider measuring $U_1 = Z_1 Z_2 = ZZI = Z \otimes Z \otimes I$ and $U_2 = IZZ$. These give the syndrome, which we can then correct.

Claim: this corrects small rotations. Suppose $|\psi\rangle = (1-\epsilon)[a|000\rangle + b|111\rangle] + \epsilon\,[a|001\rangle + b|110\rangle]$. This operator measurement will project into a definite state of being rotated or not rotated.

To correct phase flips, we can use $|{+}{+}{+}\rangle$ and $|{-}{-}{-}\rangle$ and measure the syndrome with XXI and IXX.

Claim: correcting X, Z, and $ZX = iY$ is sufficient to correct all single-bit errors.

Argument: $\mathcal{E}(\rho) = \sum_k E_k \rho E_k^\dagger$ where $E_k = \sum_j c_{kj}\sigma_j$. Then $\mathcal{E}(\rho) = c_{00}\rho + c_{11}X\rho X + \cdots + c_{12}X\rho Y + \cdots$. The syndrome measurements kill the off-diagonal terms.

3.4 The Shor 9-qubit code

The channel acts on single qubits. The code is $|0_L\rangle = (000+111)^{\otimes 3}$ and $|1_L\rangle = (000-111)^{\otimes 3}$. The syndrome measurements are Z_1Z_2, Z_2Z_3, Z_4Z_5, Z_5Z_6, Z_7Z_8, Z_8Z_9, $X_1X_2X_3X_4X_5X_6$, and $X_4X_5X_6X_7X_8X_9$.

3.5 QEC conditions

The idea is to look for more systematic approaches to understanding and finding codes. It would also be useful to describe a code more efficiently than as a set of bitstrings. Define C to be the subspace spanned by the code words and define $P = |O_L\rangle\langle 0_L| + |1_L\rangle\langle 1_L|$ to be the projector into C.

We hope that different errors we need to distinguish map C into different mutually orthogonal subspaces. We also claim that measuing qubit 1 maps into a superposition of I, X, Y, and Z errors.

Given a channel $\mathcal{E}(\rho) = \sum_k E_k \rho E_k^\dagger$ we want $E_k|\psi_k\rangle$ to map into orthogonal spaces. We also require that the E_k be unable to distinguish different codeword states. Formally,

$$\langle\psi_l|E_k^\dagger E_j|\psi_l\rangle = 0 \,\forall l \,\forall k \neq j$$

and

$$\langle\psi_l|E_k^\dagger E_k|\psi_l\rangle = d_k \,\forall l \text{ where } d_l \text{ is a c-number}$$

Theorem: Let C be a quantum code defined by $P = \sum_l |\psi_l\rangle\langle\psi_l|$ (a basis). Then there exists a quantum operation R correcting $\mathcal{E}$ on C iff $PE_k^\dagger E_j P = \delta_{kj} d_k P$.

Proof: We'll use the polar decomposition $E_k P = U_k \sqrt{P E_k^\dagger E_k P} = d_k U_k P$ (a lot of magic happened here). (Imagine $d_k = \sqrt{E_k^\dagger E_k}$,which is conveniently scalar..) Then let $P_k = U_k P U_k^\dagger = \frac{E_k P U_k^\dagger}{\sqrt{d_k}} = \frac{U_k P E_k^\dagger}{\sqrt{d_k}}$. The P_k are orthogonal: $\forall k \neq j \; P_k P_j \propto U_k P E_k^\dagger E_j P U_j^\dagger = \delta_{jk} = 0$. Somehow we know that P_k are projectors, so we can do the projective measurements and find the label k of the error. Now we perform $U_k^\dagger$, giving $R(\rho) = \sum_k U_k^\dagger E_k^\dagger P E_k U_k$. The entire operation is now $R\left(\mathcal{E}(\rho)\right) = \sum_k U_k^\dagger E_k^\dagger \left(\sum_j E_j \rho E_j^\dagger\right) E_k U_k$. Note that for $|\psi\rangle \in C$,

$$\begin{aligned} U_k^\dagger P_k E_j|\psi\rangle &= \frac{U_k^\dagger U_k P E_k^\dagger}{\sqrt{d_k}} E_j|\psi\rangle \\ &= \frac{\delta_{jk} d_k}{\sqrt{d_k}} P|\psi\rangle \\ &= \sqrt{d_k}|\psi\rangle \end{aligned}$$

and, substituting back in,

$$R\left(\mathcal{E}\left(|\psi\rangle\langle\psi|\right)\right) = \sum_{k,j} U_k^\dagger P_k E_j |\psi\rangle\langle\psi| E_j^\dagger P_k U_k = \sum_k d_k |\psi\rangle\langle\psi|$$

This is trace-preserving, so $\sum_k d_k = 1$ and the recovery operator works. (It's insufficient to prove it for all codeword states, but the proof extends simply if we put density matrices in.)

3.6 CSS codes: the idea

Quantum codes can be constructed from classical codes.

Theorem: If a quantum code C corrects $\{E_k\}$ then C also corrects any linear superposition of errors from E_k, i. e. span $(\{E_k\})$. Focus on just X, Y, and Z errors. No proof.

If a classical code corrects bit flips (e.g. the triple-redundancy code), then $PX_jX_kP = \delta_{jk}$ (where $P = |000\rangle\langle 000| + |111\rangle\langle 111|$), so it's a perfectly good quantum code to correct bit flips. We can correct Z by applying a Hadamard gate to the code space, as $Z = HXH$.

3.7 Classical linear codes

A classical binary linear $[n, k]$ is a set of 2^k n-bit strings which are closed under binary addition. We can talk about a generator G which is an $n \times k$ matrix acting by $\vec{x} = G\vec{d}$ where $\vec{x}$ is a codeword and $\vec{d}$ is arbitrary data. The parity generator H is an $n - k \times n$ matrix s. t. $HG = 0$.

The Hamming code is parameterised by $r \geq 2$ and the columns of H are the numbers from $2^r - 1$ through 1. $r = 3$ gives a $[7, 4, 3]$.

$$G = \begin{bmatrix} 1 & \\ 1 & \\ 1 & \\ 1 & H^T \\ 1 & \\ 1 & \\ 1 & \end{bmatrix}$$

3.8 CSS Codes

For a linear code C, there is a dual code $C^\perp$ with generator H^T and parity check G^T. The codewords are orthogonal to the codewords in C. If $C^\perp \subseteq C$, then we say C is *weakly self-dual*. (C is over $\mathbb{Z}^2$, so this is self-consistent.)

Lemma: If $x \in C^\perp$, then $\sum_{y \in C} (-1)^{x \cdot y} = |C|$, else $\sum_{y \in C} (-1)^{x \cdot y} = 0$.

Sketch of the proof: If $x \notin C^\perp$, then $\exists z \in C s.t.\ x \cdot z = 1$. So, $\forall y \in C\ s.t.\ x \cdot y = 0$, choose $y' = y + z$. Then $x \cdot y' = 1$.

A CSS code is made from two linear codes C_1 and C_2, with $C_2 \subseteq C_1$.

$$CSS(C_1, C_2) = \left\{ |x + C_2\rangle = \frac{1}{\sqrt{|C_s|}} \sum_{y \in C_2} |x + y\rangle \ \forall x \in C_1 \right\}$$

The notation $|x+C_2\rangle$ refers to a coset, so there are $\frac{|C_1|}{|C_2|}$ codewords, and this should be a $[n, k_1 - k_2]$ quantum code.

Claim: $CSS(C_1, C_2)$ is a $[n, k_1 - k_2]$ code.

Proof sketch: Bit flip errors are corrected by C_1. Phase flip errors are corrected by something else. If there's a bit flip error, then each element in the sum has a (constant) error added to it: $\frac{1}{\sqrt{|C_s|}} \sum_{y \in C_2} |x+y+e\rangle$. Now we just measure the syndrome defined by H_1 and correct for it. (In the quantum world, $0 \to I$ and $1 \to Z$, and then we measure the operators, which obviously commute.)

We can correct phase flip errors independently of any bit flip errors. If we represent the location of the phase flips by a binary vector e_2, then the result of the error is

$$|\psi_0\rangle = \frac{1}{\sqrt{|C_s|}} \sum_{y \in C_2} (-1)^{(x+y) \cdot e_2} |x + y\rangle$$

To correct them, we'll move into the Hadamard basis. Now we have

$$\begin{aligned} H|\psi_0\rangle &= \sum_{\substack{y \in C_2 \\ z \in \{0,1\}^n}} (-1)^{(x+y) \cdot (e_2 + z)} |z\rangle \\ &= \sum_z \left[\sum_{y \in C_2} (-1)^{(x+y) \cdot (e_2 + z)} \right] |z\rangle \end{aligned}$$

Letting $z' = e_2 + z$ and applying the lemma,

$$\begin{aligned} H|\psi_0\rangle &= \sum_z \left[\sum_{y \in C_2} (-1)^{(x+y) \cdot z'} \right] |z' + e_2\rangle \\ &= \sum_{z \in C_2^\perp} \left[\sum_{y \in C_2} (-1)^{x \cdot z'} \right] |z' + e_2\rangle \end{aligned}$$

So we can use the parity check for $C_2^\perp$ to detect or correct phase flips. This is equivalent to measuring the syndrome using X instead of Z.

Using the Hamming code, we end up with a $[[7,1,3]]$ code. The quantum distance is still undefined. The logical 1 is $XXXXXXX|0_L\rangle$.

These codes have all kinds of nice properties.

Gilbert-Varshamov bound: Let the channel rate be $R = \frac{k}{n}$. Then, for large n, there is a CSS code with rate $R \sim 1 - 2H\left(\frac{d}{n}\right)$. This is very nice (for communication, anyway).

4 Stabilizers

This is motivated by a desire to correct phase and bit flip errors together. If we write out all the parity checks like $\begin{matrix} I & I & I & Z & Z & Z & Z \\ I & Z & Z & I & Z & Z & Z \\ Z & I & Z & I & Z & I & Z \end{matrix}$ and $\begin{matrix} I & I & I & X & X & X & X \\ I & X & X & I & I & X & X \\ X & I & X & I & X & I & X \end{matrix}$, then we notice that we have a set of mutually commuting matrices (6 of them), and our codewords are $+1$ eigenstates of them.

4.1 The basics

The Pauli group $G_n$on n qubits is generated by all n-fold tensor products of $\{X, Y, Z\}$ and $\{\pm 1, \pm i\}$. All pairs of elements in the Pauli group either commute or anticommute. A stabilizer for a subspace is the set $S = \{g \in G \,|\, g|\psi\rangle = |\psi\rangle \forall |\psi\rangle \in V_s\}$. By convention, if $V_s = \emptyset$, then we say $-I \notin S$. For example, if $V_s = \text{span}\{|00\rangle\}$, then S is generated by $\{II, ZZ, IZ, ZI\}$ (or just IZ, ZI). If $V_s = \text{span}\{00 + 11\}$, then S is generated by XX, ZZ. If $\dim V_s = 2^k$ then $k = n - (\min \#$ generators of S). The null set is a problem because of the $-I$ exclusion. (Why is it there?)

Another example is $V_s = \text{span}\{000, 111\}$. Then $S = \langle ZZI, IZZ\rangle$.

For $V_s = \text{span}\left\{(000 + 111)^{\otimes n}, (000 - 111)^{\otimes n}\right\}$, then $S = \langle\rangle$. There cannot be any more.

As another example, $S = \langle XX\rangle$ gives $00 + 11$ and $01 + 10$.

4.2 Stabilizer codes

An $[[n, k]]$ stabilizer code $C(S)$ is the vector space spanned by $S \subseteq G_n$ where $S = \langle g_1, g_2, \cdots, g_{n-k}\rangle$ is a valid stabilizer.

Theorem: Given errors $\{E_a\} \in G$, if $\forall a \neq b, E = E_a^\dagger E_b \,.\, \exists g \in S \,.\, Eg = -gE$, then $\{E_a\}$ can be corrected by $C(S)$.

Proof: $\forall |psi\rangle \in C(S)$, $\langle\psi|E|\psi\rangle = \langle\psi|Eg|\psi\rangle = -\langle\psi|gE|\psi\rangle = -\langle\psi|E|\psi\rangle = 0$.

Definition: For $S = \langle g_1, \cdots, g_n\rangle$ and error E, the SYNDROME is the bitstring

$$\left(\begin{cases} 0, & [g_l, E] = 0 \\ 1, & \text{otherwise} \end{cases}\right)$$

For example, with the repetition code $\langle IZZ, ZZI\rangle$, the syndromes are $XII \to 01$, $IXI \to 11$, $XXX \to 00$, and $III \to 00$. The XXX error cannot be detected, but it does something to the code space.

Another example: $S = \langle XX\rangle$ can detect ZI and IZ, but it cannot correct both, as the syndrome for ZZ is 0.

If we use $\langle XZIZ, ZXZI, IZXZ, ZIZX\rangle$, then we have no encoded qubits. This state turns out to be interesting for other reasons.

The 7 qubit Steane code is the $r = 3$ Hamming CSS code. It has stabilizers $\langle IIIXXXX, IXXIIXX, XIXIXIX, IIIZZZ$ We can inspect the Pauli errors and find that each one anticommutes with something unique. There are 64 possible syndromes and 22 possible single-qubit errors (counting the identity). Each such error gives a unique syndrome. It turns out that it can correct some multibit errors, but, if we don't care about those, we can get a tighter code.

The tighter code does not fit into the CSS formalism. It has a stabilizer group generated by $XZZXI$ and its cyclic permutations. We can construct the syndromes for each Pauli error, and we get unique bitstrings for each one. There are 16 possible syndromes and 16 possible errors.

There are codes that are not described by stabilizers, for example $|0_L\rangle = 01 + 10$ and $|1_L\rangle = 11$. It's a code but it has no nontrivial stabilizers. This is a major research problem today.

4.3 The Normalizer

If we apply a unitary operation U then the corresponding operation on the stabilizers is $USU^\dagger$. This is because

$$\begin{aligned} U|\psi\rangle &= Ug|\psi\rangle \\ &= UgU^\dagger U|\psi\rangle \\ &= \left(UgU^\dagger\right) U|\psi\rangle \end{aligned}$$

Definition: The NORMALIZER of a S is $N(S) = \left\{g \in G \,|\, ghg^\dagger \in S \,\forall h \in S\right\}$, the set of Pauli transformations that map the stabilizer onto itself.

Lemma: Given that $-I \notin S$, $N(S) = \{g \in G \,|\, [g,h] = 0 \,\forall h \in S\}$. The proof follows from the fact that Pauli group elements always either commute or anticommute.

The normalizer is a (non-Abelian) group.

Example: $S = \langle XX\rangle$ gives $N(S) = \langle IX, XI, XX, ZZ\rangle$.

Example: $S = \langle IXX, IZZ\rangle$ gives $N(S) = \langle ZZZ, XXX, S\rangle$.

The picture here is that normalizer elements do something interesting to the codewords.

Definition: $Wt(g) = \#$ of things that aren't I.

Definition: A code $C(S)$has distance d if $N(S) - S$ has no elements of weight $< d$. (Do we require that d be maximal?) (The usual classical results about error correcting properties of these apply.)

Definition: If S has elements of $Wt < d$ (except the identity), then $C(S)$ is degenerate. Otherwise $C(S)$ is nondegenerate. There's nothing inherently wrong with degenerate codes.

4.4 The Clifford Group

What is the "generalized" normalizer of the Pauli group G?

In the following table, the contents are the top operator acting on the left:

I	X	Y	Z	H	S
X	X	-X	-X	Z	Y
Y	-Y	Y	-Y	-Y	-X
Z	-Z	-Z	Z	X	Z

These form a group and only H and S are needed (e.g. $S^2 = Z$, $HZH = X$, $XZ = -iY$).

Definition: $\langle H, S\rangle$ is the Clifford group on one qubit.

On two qubits there are more. To test an operation, we can check $Ug \stackrel{?}{=} GU$. We can compare a one-qubit Pauli operation before CNOT to CNOT followed by Pauli operations and it always works. We have (remembering that $(HH)(CNOT)(HH) = CNOT$):

I	CNOT
XI	XX
IX	IX
XX	XI
IZ	ZZ
ZI	ZI
XZ	-YY
ZZ	IZ

Definition: $C_2 \equiv$ Clifford group $= \langle H, S, CNOT\rangle$. It's the "generalized" normalizer of G. The 2 has nothing to the number of qubits.

Theorem (Gottesman-Knill): Suppose $UgU^\dagger \in G_n \,\forall g \in G_n$. Then U can be constructed from $O(n^2)$ H, S, and $CNOT$ gates, up to $e^{i\theta}$. Also, any quantum circuit composed of these elements acting on input $|0\rangle^{\otimes n}$, plus measurement in the computational basis, plus classical control, can be efficiently simulated on a classical computer! No proof given.

The Clifford group is missing the controlled-controlled-not, for example. It turns out that we can break this and do universal quantum computation by addition of a few extra gates or extra input states.

There are tons of open questions related to this.

The T gate is $\begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}$. It's also called the $\frac{\pi}{8}$ gate, which causes confusion.

Solovay-Kitaev Theorem: An arbitrary single qubit op U can be approximated to accuracy ϵ (by pretty much any distance measure) using $O(\log^3 \frac{1}{\epsilon})$ gates from $\{H, T\}$. Not only that, but the accuracy can be refined by extending the string of gates (by a constant number).

Open problem: There are algorithms to refine these, but there is no known canonical decomposition. The 3 in the bound is also not known to be tight.

4.5 Computing with stabilizer codes

We can use stabilizer codes to protect against both adversaries and errors. We can also compute on encoded data in many cases. (There are some negative theorems about computing on encrypted data.)

As an example, consider the 5-qubit code $\langle XZZXI_c \rangle$. $\bar{X} = XXXXX$ and $\bar{Z} = ZZZZZ$ are in the stabilizer, but $HHHHH$ is not. But perhaps we can make a logical $\bar{H}$ by applying permutations or other operations.

For the 7-qubit code, we can again use $\bar{X} = XXXXXXX$ and $\bar{Z} = ZZZZZZZ$. This time, $\bar{H} = H^7$ and $\bar{S} = S^7$ work. $CNOT$ is also in the normalizer. T^7 does not work (but if it did, it would not make $P = BQP$). As it happens, this code has been searched exhaustively, and there is nothing outside the Clifford group in the normalizer of this code.

On the problem set, we'll look at the 15-qubit CSS Reed-Muller code, and it has T^{15} in the normalizer but it does not have H. It was recently proved that, under somewhat reasonable assumptions, no stabilizer code has transversality for H and T.

As an example of following a computation in the stabilizer framework, we can prepare the EPR state $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$ with $(CNOT)(HI)$. We start with $\langle IZ, ZI \rangle$. After the first gate, we have $\langle IZ, XI \rangle$. After the second, we have $\langle ZZ, XX \rangle$.

We can take this farther. Suppose $S = \langle g_1, \cdots g_n \rangle$ and we measure m, where $m^2 = I$. WLOG, assume $\{m, g_1\} = 0$ and $[m, g_k] = 0 \forall k > 1$ (if there are two things it anticommutes with, we can use their product as a new generator). If the measurement results in a, then we get $S = \langle (-1)^a m, g_2, \cdots, g_n \rangle$. We could then try to "fix" the result of the measurement by applying a classically controlled g_1 to force into the $+1$ eigenstate. We can produce a stabilizer codeword by doing this one-at-a-time with each stabilizer.

5 Fault tolerant computing

In the mid 1900s, a lot of models of computation came out, and then people realized that they were all equivalent in the presence of noise. As an example, an analog computer might be given by $\vec{x}(0)$ and $\frac{d\vec{x}}{dt} = f(\vec{x})$ with a measurement at $\vec{x}(T)$. This is a very powerful model of computation. In the absence of noise, it is far more powerful than a Turing machine. In the presence of noise, it is exactly equivalent.

Noiseless quantum computation can obviously do the same thing. We need a theorem that this works with finite noise (also due to Peter Shor).

5.1 The classical case

5.1.1 The idea

We'd like to turn noisy classical gates into robust classical computation. Shannon showed in the 1940s that for a communication channel, the probability of failure was a step function of the rate, which means that a noisy channel can be used to send a (limited) amount of essentially noiseless data.

Von Neumann "Probabilistic logics and the synthesis of reliable organisms from unreliable components" in the 1950s based on the observation that, even though large vacuum tube computers seemed unreliable, people were rather reliable. He showed that there is a threshold probability of per-gate error below which perfect computation is possible.

Observation: A circuit containing N error-free gates can be simulated with probability of error $< \epsilon$ using $O(\frac{n}{\epsilon})$ gates each of which fail with probability at most p so long as $p < p_{th}$. This result is trivial if $p < \frac{\epsilon}{N}$.

We would like a circuit that computes something with probability of error p. We could try to repeat the circuit 3 times. If we had a perfect majority gate, then we could take a vote and get probability of error $\sim 3p^2$. This is good for $p < \frac{1}{3}$. This has problems:

1. Single point of failure.
2. Inefficient (the number of gates goes at $\frac{1}{\epsilon}$). We want $O\left(poly\left(\log\left(\frac{1}{\epsilon}\right)\right)\right)$.
3. The value p_{th} is not a real "threshold" in that it depends on $\frac{\epsilon}{N}$.

5.1.2 The real theorem

This is close to a proper mathematical theorem. A circuit composed of N error-free gates with no feedback (i.e. a dag) can be simulated with probability of error $< \epsilon$ using $O\left(poly\left(\log\left(\frac{N}{\epsilon}\right)\right) * N\right)$ gates, each of which fails with probability $< p_{th}$, so long as $p < p_{th}$, with p_{th} independent of N and ϵ. p_{th} will depend on things such as the kinds of gates available and the ways in which we're allowed to connect them.

Proof sketch: We compute on encoded data which we never decode. To produce gates that fail with asymptotically low probability, we can, for example, use a majority code. We can apply NAND gates on the code, but then we need to perform an error correction on result (duplicated, of course), and we claim that the probability of error is very good.

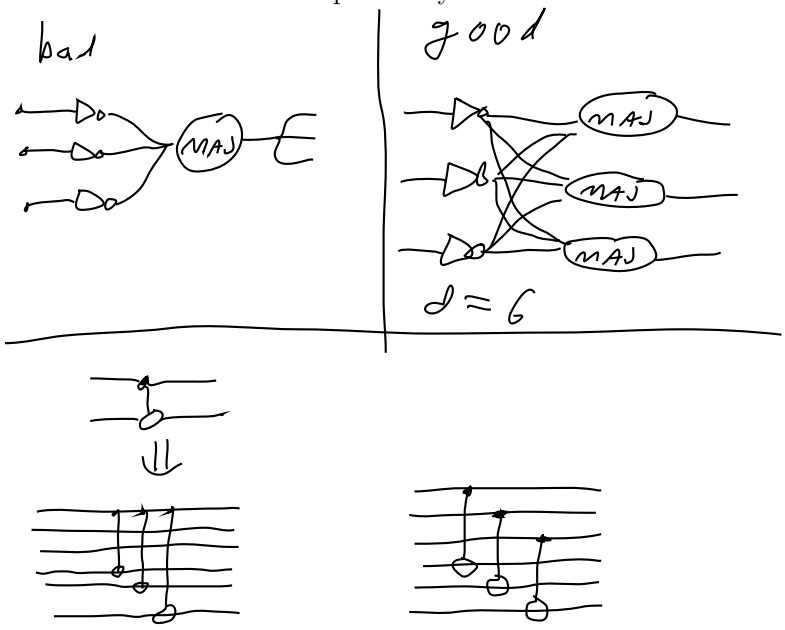
The output is incorrect only if two or more errors occur, which has probability at most $15p^2 + O(p^3)$. So we've improved our gate iff $p < \frac{1}{15}$. To get to zero, we replicate the replicated bits again.

If k is the number of recursive encodings. If C is the number of fault paths, then we want $cp_{err} = (cp)^{2^k} < p$, which is true if $p < \frac{1}{c}$ for rather small k. Hajek and Weller have a threshold of $\frac{1}{6}$. There's an upper bound of $\frac{1}{2} - \frac{1}{2\sqrt{k}}$ for k-input gates (Evans and Schulman 2003). For 2-input NAND gates, $p_{th} = \frac{3-\sqrt{7}}{4}$, which is tight (Evans and Schulman 2003).

Unfortunately our knowledge of thresholds is much more limited for quantum computation than classical computation.

5.2 Principles of fault tolerance

Definition: A procedure is fault tolerant if a single component failure causes at most one error in each encoded block of bits in the output. They look like this:



We have an equation (what exactly does it mean)?

$$p_{th} = \frac{1}{\#\text{ of fault paths}}$$

In quantum computation, we need to control eror propagation. Transversal gates ($U_{gate} = U_1 \otimes \cdots \otimes U_n$) are sufficient. We don't know whether they are necessary.

For a recursive code ("multiply concatenated"), if the length of the base code is n, c is the number of fault paths (i.e. the number of ways that two errors can destroy the computation), p is the probability of error per gate, and k is the recursion level, then the total code size is n^k and $p_{fail} \approx \frac{(cp)^{2^k}}{c}$ (for a single gate), so $p_{th} \approx \frac{1}{c}$. This can be rewritten as

$$\frac{p_{fail}}{p_{th}} = \left(\frac{p}{p_{th}}\right)^{2^k}, \tag{3}$$

which is easy to remember and universal to (all?) fault tolerance schemes.

If our goal is to simulate an N-gate circuit with probability ϵ, our errors don't expand, and we can compute on encoded data, then each FT procedure for a gate should have error $\frac{\epsilon}{n}$. If the FT procedure for a gate has size d, then:

$$\begin{aligned}
p_{fail} &= p_{th}\left(\frac{p}{p_{th}}\right)^{2^k} < \frac{\epsilon}{N} \\
2^k &= \frac{\log \frac{\epsilon}{np_{th}}}{\log \frac{p}{p_{th}}} \\
Nd^k &= \left(\frac{\log \frac{\epsilon}{np_{th}}}{\log \frac{p}{p_{th}}}\right)^{\log d} N \\
\text{curcuit size} &= Poly\left(\log\left(\frac{N}{\epsilon}\right)\right) \cdot N,
\end{aligned}$$

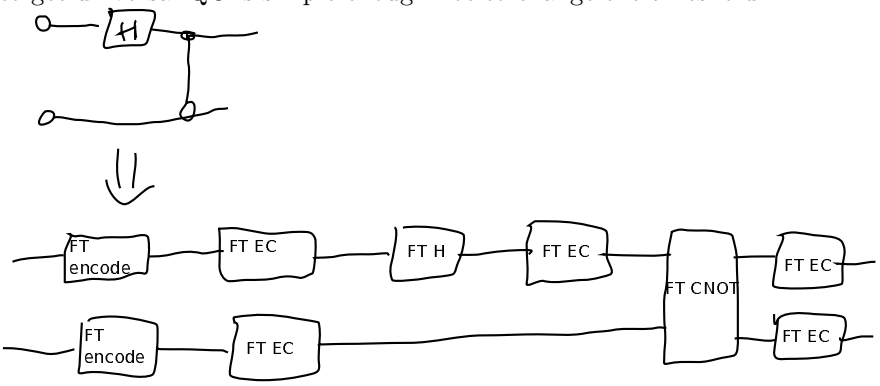
which is the desired bound.

5.3 Thresholds for FTQC

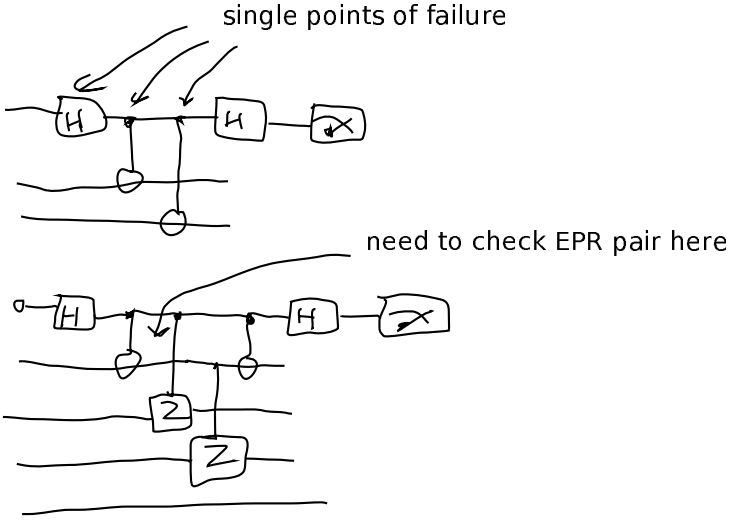
For QC, we need to consider measurement and state preparation. We also no longer have fanout for free.

Stabilizer codes are very useful because $\langle H, S, CNOT\rangle$ is the Clifford group. It seems that $CNOT$ is the most complicated thing we need to do.

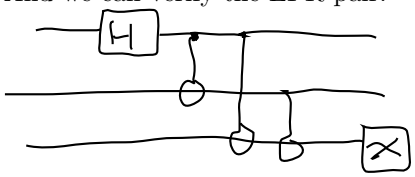
We will begin with a threshold for just the Clifford group gates and we'll hope that whatever we need to do to get universal QC is simple enough not to change the threshold.



To do fault tolerant error correction, we need to measure a Pauli group operator. We can do it with EPR pairs:



And we can verify the EPR pair:



(The above EPR verification needs to be repeated. It's also designed so that most errors in the verification won't propagate errors. If the EPR pair has the wrong phase, it won't destroy the data.)

Threshold estimate We assume that the CNOT is the limiting procedure. We use the $[[7, 1, 3]]$ Steane code. There are 6 stabilizers and CNOT traversal. We need 7 gates for the CNOT. For the syndrome measurement, we need 4 gates for the measurement and 3 repeated syndrome measurements (for example). This gives us ~ 79 gates. The number of fault paths is the number of ways in which two gates can fail, which is ~ 3081. So $p_{th} \approx \frac{1}{3081} \approx 3 \cdot 10^{-4}$.

This is a conservative estimate because many errors cancel, so we're overcounting the number of fault paths. This is an optimistic estimate because we undercounted gates and because the most limiting procedure might be Toffoli or $\pi/8$.

We can probably do better. There are funny classical gates like XAND (partially-error-correcting AND gates). Classical computing has free fanout – QC does not. Classical computing has no measurement error. But QC has entanglement, which could help.

[I will now miss a discussion of implementing Toffoli and $\pi/8$.]

6 Alternative QC models

The previous circuit model of QC involved a sequence of unitary gates. Classical computation models include:

1. Circuits (i.e. AND, OR, NOT) – not reversible
2. Turing machines
3. Memory-based computers (close to what we actually use)
4. Wolfram's cellular automata
5. Strange things such as Conway's fraction-based machine.

These are mostly equivalent up to polynomial factors.

In QC, there are bunch of such things as well.

1. Quantum Turing machines (QTM)
2. Quantum RAM (QRAM) – iffy to implement
3. Adiabatic QC
4. Approximating Jones polynomials at 5th root of unity (i.e. $p\left(e^{\frac{2i\pi}{5}}\right)$) (BCQ-complete, but the polynomial order is huge).
5. Topological QFT (TQFT) – computing with anyons (to be discussed in the next two lectures)
6. Probably more

BQP is the class of problems solvable on a QC with bounded error in polynomial time. Scott Aaronson will teach more about that later.

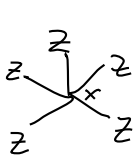
6.1 Measurement-based computation

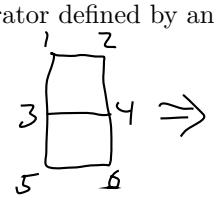
Otherwise known as one-way QC or cluster state QC.

We start with a bunch of qubits and we measure them one at a time in bases determined by program and/or previous measurements. (Question: how powerful is the classical computer? It turns out that if you're willing to restrict yourself to Clifford states you get a Clifford QC and can do it entirely in parallel. But how much post-processing is needed and are the classical problems parallelizable?)

6.1.1 Cluster states

A cluster state is a stabilizer state defined on any graph G. There is qubit on each vertex and each vertex gets a stabilizer generator defined by an X on that vertex and a Z on adjacent vertices.



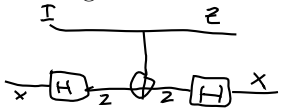


1: XZZIII
2: ZXIXII
3: ZIXZZI
4: IZZXIZ
5: IIZIXZ
6: IIIZZX

They commute because for two non-adjacent vertices, there can't be any overlap except for two X's in the same place, and, for two adjacent vertices, there are two XZ pairs which contribute $(-1)^2$. They are independent because each has an X in one unique spot.

To generate these states, we measure all the generators of the stabilizer group. Then apply Z to each vertex with the wrong sign. The single-vertex Z operators commute with the rest of the vertices, so all of the generator operators are forced simultaneously into the $+1$ eigenstate.

Alternatively, start in the state $|+\rangle^{\otimes|G|}$ (eigenstate of X) and apply $\begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & -1 \end{pmatrix}$ (controlled-Z) to each edge.



This Clifford group circuit works because $|0?\rangle \to |0?\rangle$ trivially, $|10\rangle \to |1\rangle\,(|0\rangle + |1\rangle) \to |10\rangle$, and $|11\rangle$ is the same except that the CNOT generates a sign change. Applied to a stabilizer $IIIIXI$ (with the X at a vertex), it turns the adjacent vertices into Z's. These all compute because they are all diagonal.

Yet another way is based on

$$\begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & -1 \end{pmatrix} = \exp\left[i\pi \begin{pmatrix} 0 & & & \\ & 0 & & \\ & & 0 & \\ & & & 1 \end{pmatrix} \right]$$

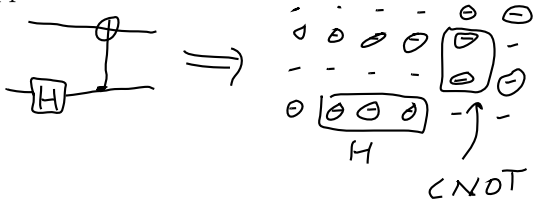
and using the appropriate Hamiltonian to apply these operators.

If we measure a cluster state vertex in the Z basis, then we anticommute with exactly one stabilizer generator (the one belonging to that vertex) and we end up with a cluster state for the graph minus that vertex.

6.1.2 Computing with cluster states

We assume that all qubits start in $|0\rangle$.

The frist step is to lay out "qubit paths" such that the paths only touch where we want operations to happen. We need three vertices in a row to do a one-qubit gate.



(We're drawing these on lattices, but that's not a requirement. It just happens to be handy because we might be able to build these on optical lattices.)

The next step is to measure all the qubits outside the logical path in the Z basis. All that's left now is a cluster state on the qubit paths, with possibly some signs wrong, so we'll need to remember that and apply virtual Pauli gates at the end of the computation to correct it.

If we were to start with the left side of a wire in $|\psi\rangle$ and the rest of the wire vertices in $|+\rangle$ (this is *not* a cluster state) and apply CZ to each edge (in any order), then we end up with what we call the "logical state $|\psi\rangle$" or $|\psi_L\rangle$. (This is not how we'll actually generate $|\psi_L\rangle$.

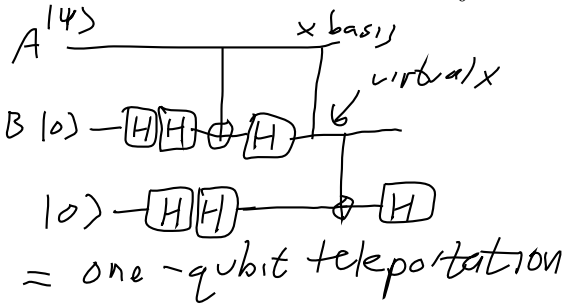
To "move the logical state" means to move the position of the $|\psi\rangle$ (prior to the CZ's) one to the right and the remove the previous position from the cluster by measuring it. We measure the vertex with the $|\psi\rangle$ in the X basis. This only fails to commute with the single closest CZ, so we can consider only the two-qubit case. Algebraically:

$$(\alpha|0\rangle + \beta|1\rangle)(|0\rangle + |1\rangle) \xrightarrow[CZ]{} (\alpha|00\rangle + \beta|10\rangle + \alpha|01\rangle - \beta|11\rangle)$$

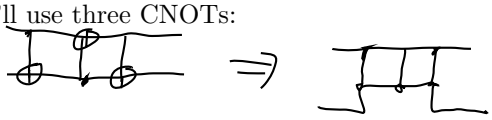
$$\xrightarrow{\text{measure } X_1} \begin{cases} (\alpha+\beta)|0\rangle + (\alpha-\beta)|1\rangle, & \text{if } 0 \\ (\alpha-\beta)|0\rangle + (\alpha+\beta)|1\rangle, & \text{if } 1 \end{cases}$$

Now, if you get a + outcome, then you've applied H. If you get a - outcome, then you've applied XH. We can't undo the X, but we can keep track of it and, in the future, apply GX instead of G. We end up accumulating a Pauli error, but we can keep track of it.

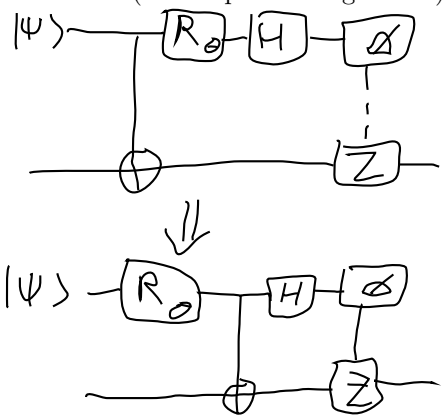
Another way to see this is to imagine we've started with all bits in $|0\rangle$ and that we've taken advantage of the fact that CZ commutes with non-adjacent measurements to reorder everything:



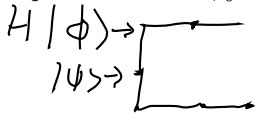
The gates $R_\theta = \begin{pmatrix} 1 & \\ & e^{-i\theta} \end{pmatrix}$, CNOT, and $HR_\theta H$ are universal for QC. If we imagine the grid to be checkerboard-colored, we'll put R_θ on white squares and $HR_{theta}H$ on black squares. To get wires to cross, we'll use three CNOTs:

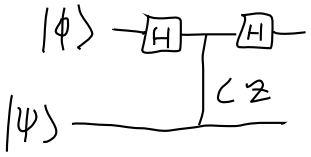


To get R_θ gates, we'll measure in the $|0\rangle \pm e^{i\theta}|1\rangle$ basis, which is equivalent to applying R_θ and measuring in the x basis (with a possible sign error).



To implement CNOT, just let two qubits intersect. The whole process is automatic.





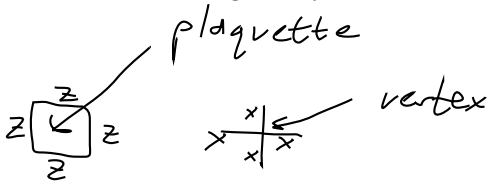
To commute all the errors to the end, we may need change signs on the thetas as we go through, because $R_\theta Z = ZR_\theta$ and $R_\theta X = XR_{-\theta}$. (Open problem: if we only have $\theta = \frac{\pi}{2}$, can we measure everything at once and then just do classical computation. Can we parallelize the classical part as well? Can we do anything that's not in NC1 (???) this way.)

6.2 Anyons and toric codes

6.2.1 Toric codes (and their extensions to other graphs)

Toric codes are also referred to as Abelian anyons. They live on grids where the qubits are edges. The squares in the grid are known as plaquettes, presumably because John Preskill used to work on lattice gauge theory. The lattice points are called vertices.

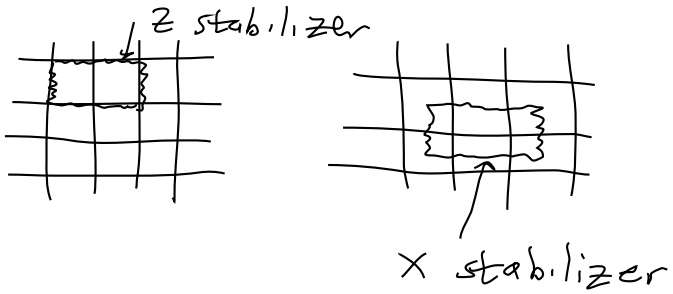
The stabilizers are given by:



On a $k \times l$ grid, there $(k+1)\,l + (l+1)\,k = 2kl + k + l$ qubits, with $(k+1)\,(l+1) - 1$ independent generators (the product of all the vertex generators is $I^{\otimes ?}$), so we have dimension 0.

On a $k \times l$ torus, there are extra edges wrapping around each side. Now there are $(k+1)\,(l+1)$ edges in each direction, for a total of $2\,(k+1)\,(l+1)$ qubits. There are $(k+1)\,(l+1) - 1$ independent plaquettes and the same number of vertex generators, for a total of $2\,(k+1)\,(l+1) - 2$ generators and two qubits left.

The normalizers in the Pauli group The stabilizer contains all unions of simple closed curves of Z's on the torus, by multiplying all the plaquettes contained inside. For the X elements of the stabilizer, look at the dual lattice (turning vertices into faces and vice versa – in this case, the same lattice offset by $\left(\frac{1}{2}, \frac{1}{2}\right)$).



The normalizer contains sets of X's that have 0, 2, or 4 edges incoming on any plaquette. Euler's theorem says that these decompose into closed curves, so the X normalizer elements outside the stabilizer are closed curves of odd winding number in either direction. The Z normalizer elements are the same things on vertices. In other words, the normalizer is the same thing as the stabilizer with the requirement that the curves have 0 winding number removed. This gives four extra generators ($\bar{X}_1$ is X winding horizontally acting on vertical qubits, $\bar{X}_2$ is X winding vertically acting on horizontal qubits, $\bar{Z}_1$ is Z winding vertically acting on vertical qubits, and $\bar{Z}_2$ is Z winding horizontally acting on horizontal qubits). It's easy to check the (anti-)commutation relations.

To correct errors, we measure the syndromes at each vertex and each plaquette. If we get -1 for a pair of "stars", then the error is some path between them. It doesn't matter which path we use to correct it so long as they are topologically equivalent. Pick the shortest. If we have 4 -1's, then we want the shortest matching (this is min-cost matching, which can be solved in polynomial time, although the proof is a little bit nontrivial).

We know that this is a CSS code b/c the generators are all X's or all Z's. If we defined it on an arbitrary graph on a manifold (i.e. a graph plus a definition of what a face is), then #generators = #faces $-$ 1 + #vertices $-$ 1. The #encoded qubits $= E - F - V + 2 = 2 \cdot$ Genus. The genus is the number of holes, so it is 1 for a torus. (Another way to see it is that there are 2^{genus} ways to thread through the holes, and there are both X lines and Z lines.

6.2.2 The Hamiltonian version and Abelian anyons

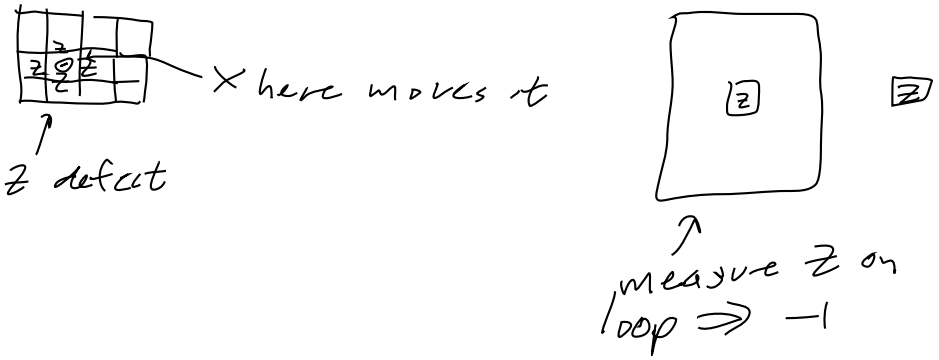
Start with the Hamiltonian

$$H = \sum_i \left(\frac{I - g_i}{2} \right).$$

The ground state is then the state where all of the stabilizers are in the +1 eigenstate. Then the ground state is the code space.

There can't be just one -1 eigenvalue in a different state because multiplying all the other stabilizers recovers that one. But there can be 2. As before, though, if we have two -1 eigenstates (say on the X stabilizer elements, i.e. vertices), then the error could be corrected by applying Z over the shortest path.

The Z operator moves one of these "X_{-1} particles, " and, if two collide, they annihilate. Similarly, X moves a "Z_{-1} particle."

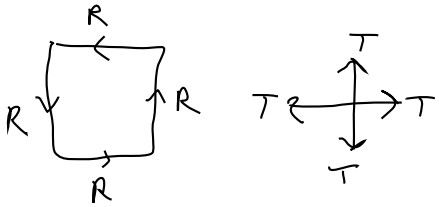


If we move a Z defect around the X defect, then we'll cross the "string of $X's$" coming out of the Z defect once, and we'll pick up a phase of -1. This is a representation of $\mathbb{Z}_2$.

Qutrits and $\mathbb{Z}_3$ anyons Introduce operators

$$\begin{aligned} T &= \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \\ R &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \omega & 0 \\ 0 & 0 & \omega^2 \end{pmatrix} \\ \omega &= e^{2\pi i/3} \end{aligned}$$

Now we have operators $\overset{T}{\rightarrow}=\overset{T^2}{\leftarrow}$ and $\overset{R}{\rightarrow}=\overset{R^2}{\leftarrow}$. Flipping a qubit is just a basis change $|1\rangle \leftrightarrow |2\rangle$. The stabilizer generators are



To check that the generators commute, there's only one case (say the T is at the top-right corner of the R):

$$R_1 R_2 T_1 T_2^2 = \omega^3 T_1 T_2^2 R_1 R_2.$$

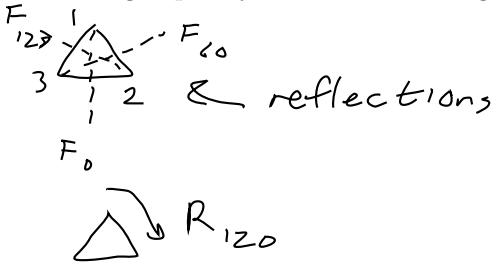
Moving a T defect around an R defect gives a phase of ω or ω^2 depending on direction.

6.2.3 The group S_3

S_3 is the group of permutations of three elements a, b, and c. The group elements can be written with a cycle representation, where, for example, (12) means interchanging the first two elements. So (23)(12) = (132) (you can draw a picture, but it's not very important for what we're currently doing).

6.2.4 Conjugacy clases of finite groups

Consider the group of symmetries of a triangle (the dihedral group).



If we rotate, flip, and unrotate, then we get a different flip ($RFR^{-1} = F'$). Specifically $R_{120} F_0 R_{123}^{-1} = F_{120}$.

A conjugacy class is a set of group elements that can be obtained by conjugating one element like this. An element with a conjugacy class size of 1 commutes with everything.

The number of conjugacy classes in a finite group is the number of such classes of elements. In this case, there are three: $\{F_0, F_{60}, F_{120}\}$, $\left\{R_{120}, R_{120}^{-1}\right\}$, and $\{I\}$.

A representation is a map from $G \rightarrow M_{k\times k}$ which preserves the group structure, so $R(g_1 g_2) = R(g_1) R(g_2)$. In this group, $R : g \rightarrow I$ is a representation. $R : g \rightarrow$ permutations is another. -1 for flips and 1 for non-flips is another. An irreducible representation (irrep) is one which cannot be diagonalized so that all of its matrices have the same block diagonal form.

Every representation has an associated character $\chi_R(g) = Tr(R(g))$. Characters are constant on conjugacy classes. (Proof: If $g_1 = h^{-1} g_2 h$, then $Tr(R(g_1)) = Tr\left(R(h)^{-1} R(g_2) R(h)\right) = Tr(R(g2)))$.

Two representations are the same if they one can be converted into the other by a similarity transformation.

There's a theorem that the number of conjugacy classes is equal to the number of irreps.

If we include complex matrices, then the character table is a square table with conjugacy classes in columns and irreps in rows. When the columns are weighted by the size of the conjugacy class, then the rows are orthogonal. The columns (still weighted) are orthonormal in the sense that the sum of the squares times the weight equals the size of the group.

	I	R	F
irrep 1	1	1	-1
irrep 2	1	1	-1
irrep 3	2	-1	0

Now we can find the last irrep. In the basis $\begin{pmatrix} 1 & 1 & 1 \end{pmatrix}^T$, $\begin{pmatrix} 1 & \omega & \omega^2 \end{pmatrix}^T$, $\begin{pmatrix} 1 & \omega^2 & \omega \end{pmatrix}^T$, then the rotations and flips reduce into I in the top-left and the third irrep in the other block.

This means that two representations are the same iff they result in the same characters.

6.2.5 Non-ableian anyons

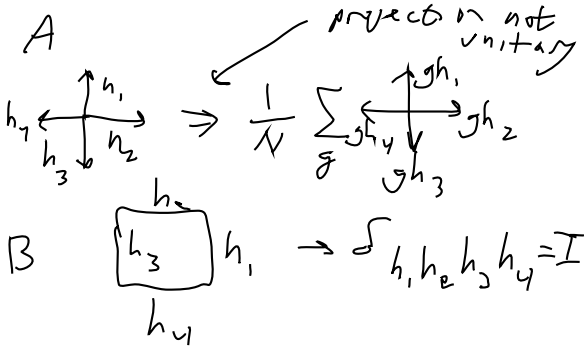
Label the basis states of our Hilbert space by group elements (e.g. $|g_1\rangle, \cdots, |g_m\rangle$). (So, if we used S_3, we'd have 6-ary qudits, with basis states $|(12)\rangle$, $|(13)\rangle$, $|(23)\rangle$, $|(123)\rangle$, $|(132)\rangle$, $|I\rangle$.)

We'll define directions on edges, where $\overset{|z\rangle}{\rightarrow} = \overset{|z^{-1}\rangle}{\leftarrow}$. For each group element g, we have two unitary operators $L_+^g|z\rangle = |gz\rangle$ and $L_-^g|z\rangle = |zg^{-1}\rangle$. This way, L_- on a left-facing qudit looks like L_+ on a right-facing qudit. We have nonunitary operators $T_+^h|z\rangle = \delta_{h,z}|z\rangle$ and $T_-^h|z\rangle = \delta_{h,z^{-1}}|z\rangle$. Any operator at all can be constructed from these operators and scalars. The commutation relations are

$$\begin{aligned} L_+^g T_+^h &= T_+^{gh} L_+^g \\ L_-^g T_+^h &= T_+^{hg^{-1}} L_-^g. \end{aligned}$$

The star relation gives $A_g(s) = \prod_{j \in s} L_{j,s}^g(j)$, where we use + if j points out of s and - otherwise. The plaquette relations gives $B_h(p,s) = \sum_{h_1 \cdots h_4 = h} \prod_{i<4} T^{h_i}(j_i)$, where s is the starting point. This means that $\sum_{h \in G} B_h(p,s) = I$ because you select every possibility.

The stabilizer generators are $A(s) = \frac{1}{N} \sum_{g \in G} S_g(s)$ and $B(p) = B_I(s,p)$ (the starting vertex doesn't matter because if $h_1 h_2 h_3 h_4 = I$ then $h_2 h_3 h_4 h_1 = I$). These are both projectors and we want the null space. (To prove that they A is a projector, we note that this sort of symmetrization over group elements has no effect if it's already been done, so $A^2 = A$. B is straightforward.)



Considering a star operator on a vertex of a plaquette, there are two edges of overlap. That piece of the

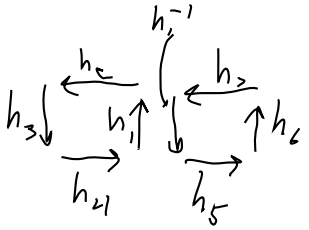
B operator looks like $\sum_h \sum_{h_1 h_2 = h} T^{h_1}(j_1) T^{h_2}(j_2)$. That piece of the A operator looks like

$$\begin{aligned} A &= \frac{1}{N} \sum_g L^g(j_1) L^g(j_2) \\ &= \sum_{h = h_1 h_2} \sum_g L^g_+(j_1) L^g_-(j_2) T^{h_1}_+(j_1) T^{h_2}_+(j_2) \\ &= \sum_{h_1 h_2 = h} \sum_g T^{g h_1}_+(j_1) T^{h_2 g^{-1}}_+(j_2) L^g_+(j_1) L^g_-(j_2) \end{aligned}$$

We'd like to show that these commute independently of h. Something is backwards in this proof.

On a torus we get a degenerate ground state space.

With the Hamiltonian $\sum_s (I - A(s)) + \sum_p (I - B(p))$, excitations are quasiparticles:



Define $g_1 = h_1 h_2 h_3 h_4$ (a characterization of the quasiparticle). (Starting at a different corner just conjugates it with something, e.g. $h_1^{-1} g_1 h_1$.) Then $g_2 = h_6 h_7 h_1^{-1} h_5$. The product around the whole loop is $h_5 h_6 h_7 (h_1 h_1^{-1}) h_2 h_3 h_4 = \left(h_6 g_2 h_5^{-1}\right) g_1$. If g_1 were the identity, then we're back to where we started, up to a conjugacy.

If $(I - B) = 1$ we have a magnetic quasiparticle, and, if $(I - A) = 1$ then we have an electric quasiparticle. If they are touching (the electric vertex is on the magnetic plaquette), then the resulting quasiparticle carries both electric and magnetic charge.

The number of magnetic quasiparticle types is equal to the number of conjugacy classes, because, if we draw a big loop around an excitation with magnetic charge g, we get hgh^{-1} (due to where we start on the ring), so we can only tell up to a conjugacy class what type of particle is there. These excited states are highly degenerate, depending on where the particles are, which element in the conjugacy class they are, etc.

If there are two quasiparticles in the ring of charge g and h, then the resulting magnetic charge is found by moving them together, obtaining $\sigma g \sigma^{-1} \sigma' h \sigma'^{-1}$.

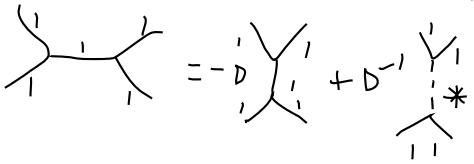
In the $\mathbb{S}_3$ case, the combination of a rotation and a flip must give some kind of flip. But the combination of two rotations gives either a rotation or the identity, as does the combination of two flips.

The overall excited state has to have total something equal to I. This is subtle because R charges can reproduce, but flips are conserved.

If there is an electric charge but no magnetic charge, then there is an associated irrep. To see this, we can write $L_g(s) = \prod_{j \in \text{star}(s)} L_g(j)$. It can be shown that, for any basis state on each edge, $\forall g$, $\exists$some family of conjugates of g dependent on s such that $\prod_s L_{g(s)} = I$. The rest lives in some paper of Kitaev's, presumably.

6.2.6 Anyons in general

There is some number of particle types. With Fibonacci anyons, there are two: 1 and $*$. Any set of anyons can be combined into a new new one, following the rule that $1 + 1 = *$ or $1 + 1 = 1$. The original state is a superposition of these diagrams. A TQFT (topological QFT) is a set of rules for rewriting diagrams:



where $\zeta = e^{2\pi i/5}$ and $D = \frac{\sqrt{5}-1}{2}$

7 Hidden subgroup problem

[missed part]

7.1 Discrete log problem

We have a prime p and a generator g such that $g^{p-1} \equiv 1 \pmod{p}$ and $g^x \neq 1 \pmod{p}$.

Given $x \bmod p$, find r so that $g^r \equiv x$ (i.e. $r = \log_g x \pmod{p}$). Classically, the best known algorithm takes $2^{\epsilon(\log p)^{1/3}(\log\log p)^{2/3}}$.

To formulate it as a hidden subgroup problem, take the group $G = \mathbb{Z}_{p-1} \times \mathbb{Z}_{p-1}$ with the function $f(s,t) = g^t x^{-s} \pmod{p}$. If $g^t x^{-s} = g^{t'} x^{-s'}$, then $g^{t-t'} = x^{s-s'} = g^{r(s-s')}$. This means that $f(t,s) = f(t',s')$ iff $r(s-s') = t-t'$, so f defines the hidden subgroup (t,s) such that $t = rs$.

If we can solve the hidden subgroup problem, then we usually find $t = rs$ with s and $p-1$ relatively prime, and we can find r by division.

7.2 Fourier transforms on a cyclic group

7.2.1 Definition

The cyclic group $\mathbb{Z}_n$ is the additive group of integers mod n. The FT is

$$|j\rangle \to \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} e^{-2\pi i jk/n} |k\rangle.$$

On a product of cyclic groups, the FT is

$$|j_1\rangle|j_2\rangle \to \sum_{k_1=0}^{n_1-1} \sum_{k_2=0}^{n_2-1} \exp\left[-2\pi i \left(\frac{j_1 k_1}{n_2} + \frac{j_2 k_2}{n_2}\right)\right] |k_1\rangle|k_2\rangle.$$

The Hadamard gate is the FT over $\mathbb{Z}_2$.

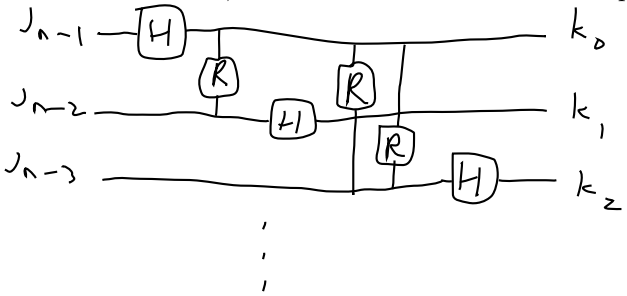
We can do this efficiently for groups whose size factor into small primes. For large primes, we can approximate them by using even larger powers of 2.

7.2.2 The quantum algorithm

For the FT mod 2^n, we need

$$|j\rangle \to \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{-2\pi i jk/2^n} |k\rangle.$$

We write the numbers j and k in binary (i.e. $j = \sum_{\alpha=0}^{n-1} j_\alpha 2^\alpha$). Looking at $|j_{n-1}\rangle$, the contribution to the phase is $(-1)^{j_{n-1}k_0}$. This means that we can just put a Hadamard between the input $|j_{n-1}\rangle$ and the output $|k_0\rangle$. Trying the same thing with $|j_{n-2}\rangle$ and $|k_1\rangle$ requires an addition correction for $\exp\left[\frac{-2\pi i j_{n-2}k_0}{4}\right]$, which gives a circuit like this, where the rotations are chosen appropriately:



We can ignore controlled rotations that go a long distance (more than $\log n$ wires), because they're tiny, which I think gives an asymptotic speedup.

7.3 Quantum algorithm for hidden subgroup problem

For a non-abelian group, all irreducible representations are maps $\rho \to \mathbb{C}$ such that $\rho(g_1)\rho(g_2) = \rho(g_1 g_2)$. For $\mathbb{Z}_6$, the maps are $\rho_k(j) = e^{-2\pi i jk/6}$ and the character table looks like a Fourier transform. Let $\hat{G}$ be the set of irreducible representations of G, which, in the abelian case, is just another copy of G.

We want $f(g)$ to be different on different cosets.

With a group G and $n = |G|$, where H is the hidden subgroup:

1. Start with $\frac{1}{\sqrt{n}} \sum_{g \in G} |g\rangle|0\rangle$.
2. Compute $\frac{1}{\sqrt{n}} \sum_{g \in G} |g\rangle|f(g)\rangle$.
3. Take FT: $\frac{1}{n} \sum_{g \in G} \sum_{\rho \in \hat{G}} \rho(g)|\rho\rangle|f(g)\rangle$ (this is just the usual Fourier transform). Looking at a specific $|f(g)\rangle$ (after measurement), we have
$$N \sum_{\rho \in \hat{G}} \sum_{h \in H} \rho(ch)|\rho\rangle = \sum_{\rho \in \hat{G}} \rho(c) \sum_{h \in H} \rho(h)|\rho\rangle.$$
4. Measure $|\rho\rangle$. Now we have, where N is the same normalization as above,
$$N\rho(c) \sum_{h \in H} \rho(h). \tag{4}$$

Lemma: $\sum_{h \in H} \rho(h) = 0$ unless $\rho(h) = 1 \, \forall h \in H$.

Proof: Choose $h_1 \in H$. Then $\sum_{h \in H} \rho(h_1 h) = \rho(h_1) \sum_{h \in H} \rho(h)$, and the result follows.

This means that we've just found a character that is constant on H (because these are 1D irreps). We're equally likely to get any of the irreps because they all have the same weight in eq. 4.

There are $\frac{n}{|H|}$ of these irreps. Given a few, we can take find more (they form a group), and, if we have enough elements of the group of irreps that are constant over H, we can find generators for the entire group. (The idea behind the algorithm is to break into products of prime powers and then do something resembling Gaussian elimination to produce elements that are zero on more and more of the pieces.)

7.3.1 Hidden subgroup on dihedral groups

On the non-abelian dihedral group D_3, the irreps are (where the bottom four rows are matrix elements of a 2×2):

I	R_1	R_2	F_1	F_2	F_3
1	1	1	1	1	1
1	1	1	-1	-1	-1
1	ω	ω^2	0	0	0
1	ω^2	ω	0	0	0
0	0	0	1	ω	ω^2
0	0	0	1	ω^2	ω

The bottom four rows are only defined up to linear combinations.

The matrix above is unitary (with some normalization corrections) and defines some sort of mapping between group elements and representations. There is a compact definition in the appendix of Nielsen and Chuang.

If we start with the dihedral group on $N = 2^n$, every element can be represented as $y^k x^j$, where $0 \le j < N$ and $0 \le k \le 1$ (x is a rotation by 1 and y is a flip). Then $x^N = y^2 = 1$.

If we have a hidden subgroup $H = \{1, x^s y\}$ and an oracle f satisfying $f(x^s y g) = f(g) \; \forall g$, we would like to find s.

1. Start with $\sum_g |g\rangle$.

2. Compute

$$\begin{aligned} & \sum_{g=x^k} \left(|x^s y\rangle + |g\rangle\right) |f\left(g\right)\rangle \\ = & \sum_k \left(|x^s y x^k\rangle + |x^k\rangle\right) |f\left(x^k\right)\rangle \\ = & \sum_k \left(|y x^{k-s}\rangle + |x^k\rangle\right) |f\left(x^k\right)\rangle \end{aligned}$$

3. Assume f is different for different k, measure the rightmost register, and take FT over x:

$$\begin{aligned} & |y x^t\rangle + |x^{t+s}\rangle \\ \rightarrow & \sum_r \left(\omega^{-rt} |y x^r\rangle + \omega^{-r(t+s)} |x^r\rangle\right) \end{aligned}$$

4. Measure r, giving

$$\begin{aligned} & \omega^{-rt} |y\rangle + \omega^{-r(t+s)} |1\rangle \\ = & \alpha \left(|y\rangle + \omega^{-rs} |1\rangle\right) \end{aligned}$$

It took several years to figure out how to get s from here.

7.4 Hidden shift problem

7.4.1 The Legendre symbol

The Legendre symbol is

$$\left[\frac{a}{p}\right] = \begin{cases} 1, & \exists x \, a \equiv x^2 (\text{mod } p) \\ -1, & \nexists x \, a \equiv x^2 (\text{mod } p) \\ 0, & a = 0 \end{cases}$$

and can be computed efficiently $O(poly(log(p)))$ using quadratic reciprocity. We could also generate the whole sequence of Legendre symbols for a given p in $O(n)$ by finding the 1's, but this isn't very helpful.

7.4.2 The hidden shift problem for the Legendre symbol

We are given $f\left(x\right) = \left[\frac{x+s}{p}\right]$ as an oracle. We would like to find s, which, classically, ought to take about $O(\log p)$ queries (information-theoretically). The Legendre symbol is a character of the multiplicative group mod p, because $\left[\frac{a}{p}\right]\left[\frac{b}{p}\right] = \left[\frac{ab}{p}\right]$.

The quantum algorithm is:

1. Start with $\sum_{x \in \mathbb{F}_p} |x\rangle$.

2. Compute $\sum_{x \in \mathbb{F}_p} \left[\frac{x+s}{p}\right] |x\rangle$. This is not quite reversable, but turns out to be close enough.

3. Take FT. Define $\omega = \exp \frac{2\pi i}{p}$. This gives

$$\begin{aligned}
& \sum_{x,y\in\mathbb{F}_p} \left[\begin{array}{c} x+s \\ p \end{array}\right] \omega^{-xy}|y\rangle \\
= & \sum_{x,y\in\mathbb{F}_p} \left[\begin{array}{c} x+s \\ p \end{array}\right] \omega^{-(xy+sy)}\omega^{sy}|y\rangle \\
\approx & \sum_{y\neq 0}\sum_{x\in\mathbb{F}_p} \left[\begin{array}{c} (x+s)\,y \\ p \end{array}\right] \left[\begin{array}{c} y^{-1} \\ p \end{array}\right] \omega^{-(x+s)y}\omega^{sy}|y\rangle \\
= & \sum_{y\neq 0}\sum_{z\in\mathbb{F}_p} \left[\begin{array}{c} z \\ p \end{array}\right] \omega^{-z} \left[\begin{array}{c} y^{-1} \\ p \end{array}\right] \omega^{sy}|y\rangle \\
= & \alpha \sum_{y\neq 0} \left[\begin{array}{c} y^{-1} \\ p \end{array}\right] \omega^{sy}|y\rangle
\end{aligned}$$

4. Compute

$$\begin{aligned}
& \alpha \sum_{y\neq 0} \left[\begin{array}{c} y^{-1} \\ p \end{array}\right] \omega^{sy} \left[\begin{array}{c} y \\ p \end{array}\right] |y\rangle \\
= & \alpha \sum_{y\neq 0} \omega^{sy}|y\rangle
\end{aligned}$$

5. Take the inverse FT: $\alpha|s\rangle$.

This clearly works for any multiplicative character on a finite field.

For any irrep of a finite field $\mathbb{F}_q$, we have $\hat{\rho}\,(y) = \overline{\rho\,(y)}\hat{\rho}\,(1)$.

8 Entanglement

A pure state $|\psi_{AB}\rangle$ on two systems $A \otimes B$ is SEPARABLE iff $|\psi_{AB}\rangle = |\psi_A\rangle|\psi_B\rangle$. Otherwise it is ENTANGLED. Mixed state entanglement is a good deal messier, and we may get to it at the end.

8.1 Schmidt decomposition

8.1.1 Decomposition and rank

The Schmidt decomposition of $|\psi\rangle$ is

$$|\psi\rangle = \sum \sqrt{\lambda_i}|v_i\rangle_A \otimes |w_i\rangle_B$$

where

$$\begin{aligned}
\langle v_i|v_j\rangle &= \delta_{ij} \\
\langle w_i|w_j\rangle &= \delta_{ij}.
\end{aligned}$$

Proof: Look at $\mathrm{Tr}_B\,|\psi\rangle\langle\psi| = \sum \lambda_i|v_i\rangle\langle v_i|$. Then

$$|\psi\rangle = \sum \alpha_i|v_i\rangle \otimes |w_i\rangle. \tag{5}$$

We need $\langle w_i|w_j\rangle = \delta_{ij}$ and $|\alpha_i|^2 = \lambda_i$. The $|v_i\rangle\langle v_j|$ term in (5) is $\alpha_i\alpha_j^*\,\mathrm{Tr}\,|w_i\rangle\langle w_j|$.

The Schmidt rank is the number of terms in the Schmidt decomposition.

Any mixed state can be written as $\sum_{ijkl} \alpha_{ijkl}|v_i\rangle|w_j\rangle\langle v_k|\langle w_l|$. $\mathrm{Tr}_A\,|\psi\rangle\langle\psi|$ and $\mathrm{Tr}_B\,|\psi\rangle\langle\psi|$ have the same eigenvalues.

The Schmidt rank of a separable state is 1 and the the Schmidt rank of an entangled state is >1.

8.1.2 POVMs

A POVM is a set of positive operators $\{E_1, \cdots, E_k\}$ with $\sum E_i = I$. The probability of outcome i is $\operatorname{Tr} E_i\rho$.

We can assume (sort of) that the final state after outcome i is $\sqrt{E_i}\rho\sqrt{E_i}$. This follows from a description using general quantum operations are $\rho \to \sum_{k=1}^{l} A_k\rho A_k^\dagger$, where $\sum A_k^\dagger A_k = I$. If you somehow kep track of which was performed, you set $|v\rangle \to A_k|v\rangle$ w.p. $\operatorname{Tr} A_k^\dagger A_k|v\rangle\langle v|$. The corresponding POVM is $A_k^\dagger A_k = E_k$. By the polar decomposition theorem, $A_k = U\sqrt{A_k^\dagger A_k}$

8.1.3 Local operations and classical communication

Simple example As an example, Alice and Bob have the state $\frac{1}{\sqrt{3}}|0\rangle|0\rangle + \sqrt{\frac{2}{3}}|1\rangle|1\rangle$. Can they get an EPR pair? Alice performs POVM $E_1 = \begin{pmatrix} 1 & \\ & \frac{1}{2} \end{pmatrix}$, $E_2 = \begin{pmatrix} 0 & \\ & \frac{1}{2} \end{pmatrix}$. With probability $\frac{2}{3}$, they get an EPR pair and, with probability $\frac{1}{3}$ they get something separable.

The reverse can be done with probability 1 with the measurements $\begin{pmatrix} \frac{2}{3} & \\ & \frac{1}{3} \end{pmatrix}$ and $\begin{pmatrix} \frac{1}{3} & \\ & \frac{2}{3} \end{pmatrix}$.

When can you go from one state to another? Nielsen's theorem says that, given $|\psi\rangle$ and $|\phi\rangle$ (pure states) with $|\psi\rangle = \sum \sqrt{\lambda_i}|v_k\rangle|w_k\rangle$ and $|\phi\rangle = \sum \sqrt{\mu_i}|\tilde{v_k}\rangle|\tilde{w_k}\rangle$. The different bases don't matter because local unitary operations can change the bases.

Then you can go from $|\psi\rangle \to |\phi\rangle$ w.p. 1 iff $\{\mu_i\}$ majorizes $\{\lambda_i\}$. This means that $\mu_i \geq \lambda_i$, $\mu_i+\mu_j \geq \lambda_i+\lambda_j$, etc.

Fact (unproved): If μ majorizes λ, then you can go from μ to λ by transformations of the form $\mu_i' = \mu_i - \epsilon_i$, $\mu_j' = \mu_j + \epsilon_i$ $(i < j)$. It may take a lot of steps, though.

We can use measurements to go from $|\psi\rangle \to |\phi\rangle$. Using the fact above, we only need to worry about 2D systems.

Suppose we start with $\sqrt{\alpha}|00\rangle + \sqrt{\beta}|11\rangle$. Let Alice apply the POVM $\sqrt{E_1} = \frac{1}{\sqrt{2}}\begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$, $\sqrt{E_2} = \frac{1}{\sqrt{2}}\begin{pmatrix} \cos\theta & -\sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$. If we cat 1, then $\sqrt{E_1}|\psi\rangle = \cos\theta\sqrt{\alpha}|00\rangle + \sin\theta\sqrt{a}|10\rangle + \sin\theta\sqrt{\beta}|01\rangle + \cos\theta\sqrt{\beta}|11\rangle$. Taking the partial trace to get the Schmidt eigenvalues gives

$$\begin{aligned} \operatorname*{Tr}_A \sqrt{E_1}|\psi\rangle\langle\psi| &= \begin{pmatrix} \left(\cos^2\right)\alpha & (\sin\cos)\sqrt{\alpha\beta} \\ (\sin\cos)\sqrt{\alpha\beta} & \left(\sin^2\right)\beta \end{pmatrix} + \begin{pmatrix} \left(\sin^2\right)\alpha & (\sin\cos)\sqrt{\alpha\beta} \\ (\sin\cos)\sqrt{\alpha\beta} & \left(\cos^2\right)\beta \end{pmatrix} \\ &= \begin{pmatrix} \alpha & (2\sin\cos)\sqrt{\alpha\beta} \\ (2\sin\cos)\sqrt{\alpha\beta} & \beta \end{pmatrix}, \end{aligned}$$

so we can continuously bring the eigenvalues together.

The other way: if $|\psi\rangle \to |\phi\rangle$ by LOCC, then the Schmidt eigenvalues of $|\phi\rangle$ majorize $|\phi\rangle$. The first step is for Alice to make measurements and send results, then Bob does the same thing, etc. But it doesn't matter who makes the measurements, because Alice and Bob can do local unitaries to fix up the bases (they have the same effect on the eigenvalues). So it could be done as one giant measurement by Alice.

If Alice takes $|\psi\rangle \to M_i|\psi\rangle$, then Alice's final matrix is $\sum M_i\rho_\phi M_i^\dagger = \rho_\phi$. The final state is $|\phi\rangle\langle\phi| = \sum \varepsilon^B\left(M_i|\psi\rangle\langle\psi|M_i^\dagger\right)$. The LHS is a pure state, so the RHS terms are all the same pure state (to get Schmidt rank 1). Then $M_i\rho_\psi M_i^\dagger = \rho_\phi$. Now take the polar decomposition: $M_i\sqrt{\rho_\psi} = \sqrt{M_i\rho_\psi M_i^\dagger}U_i$. This means that

$$\begin{aligned} \rho_\psi &= \sum_i \sqrt{\rho_\psi}M_i^\dagger M_i\sqrt{\rho_\psi} \\ &= \sum U_i^\dagger\sqrt{M_i\rho_\psi M_i^\dagger}\sqrt{M_i^\dagger\rho_\psi M_i}U_i \\ &= \sum_i U_i^\dagger M_i\rho_\psi M_i U_i = \sum U_i^\dagger\rho_\phi U_i, \end{aligned}$$

so the eigenvalues of ψ majorize those of ϕ. This can be obtained from the fact, with Π_k being a k-dim projector, $\operatorname{Tr}(\Pi_k \rho_\phi \Pi_k) \leq \mu_1 + \mu_2 + \cdots + \mu_k$.

We define the entanglement $\operatorname{Ent}(|\psi\rangle) = \sum -\lambda_i \log \lambda_i = H(\operatorname{Tr}_B |\psi\rangle\langle\psi|)$ where the λ_i are the Schmidt eigenvalues. There's a theorem that, if Alice and Bob have n copies of $|\psi\rangle$ then they can obtain a state within distance ϵ of $n \operatorname{Ent}(|\psi\rangle) - \epsilon$ EPR pairs where $\epsilon \to 0$ as $n \to \infty$.

The proof uses the Schmidt decomposition of the entire state $|\psi\rangle^{\otimes n} = \sum \sqrt{\lambda_{i1}\lambda_{i2}\cdots}|v_{i1}, \cdots\rangle_A |w_{i1}w_{i2}\cdots\rangle_B$ where most of the mass of the state has amplitude where $\lambda_{i1}\cdots\lambda_{in} \approx 2^{-H(\rho)n \pm \epsilon n}$, so we ignore eigenvalues much larger or smaller than that.

9 Quantum protocols

9.1 Perspective

There are a bunch of different communication channels.

Informaion	Channel	
Classical	Classical	Shannon
Classical	Quantum	Holevo, HSW
Quantum	Noiseless quantum	Schumacher coding
Quantum	Noisy quantum	coherent information
Classical	Noise Q + entanglement	Ask Peter

There are also distributed algorithms (e.g. Alice and Bob share a quantum channel and want to compute something).

There are also interesting applications in cryptography (due to no-cloning, etc.). There are also complexity things like RSA (broken by QC) and public key encryption.

The idea we'll use is that coherent information can be a quantitative measure of trust.

9.2 Classical communication complexity

Given a function $f : \{0,1\}^n \otimes \{0,1\}^n \to \{0,1\}$ where Alice owns the first set of bits x and Bob owns the second set y, we would like to know how much communication is required to compute $f(x,y)$. The problem depends on what kind of communication is available, how reliable the result must be (exact, bounded-error, one-sided error, etc) and other resources (shared randomness, entanglement). (Most of this stuff is due to Andy Yao.)

As a concrete example, consider $f(x,y) = (x = y)$, the equality function. In the literature, $D(EQ)$ is the exact answer complexity. $D(EQ) = n$.

For bounded error, $R(EQ) = O(\log n)$ (with public shared information). A&B agree on $p \geq \frac{n}{\epsilon}$ where $\epsilon < 1$ is the probability of error. Define polynomials $A(z) = \sum x_i z^{i-1}$, $B(z) = \sum y_i z^{i-1}$ and $C = A - B$, all over $\mathbb{F}_p$, where x_i and y_i are the bits of the message. We have $x = y \implies C(z) = 0$. If $x \neq y$, then there are at most n roots of $C(z)$. (We could use a higher base as well, reducing n by a factor of $\log p$.) The protocol is that Alice chooses a random $z \in \mathbb{F}_p$ and sends z, $A(z)$ to Bob. Bob computes $C(z)$ and announces EQUAL iff $C(z) = 0$. The probability of error is bounded above by $\frac{n}{p} \leq \epsilon$ and A sends B $2 \log p = O(\log n)$ bits.

Some results for other problems are:

Problem	Cl. exact	R	Q	Q exact
EQ	n	$\log n$	$\log n$	n
Parity	n	n	n	n
DISJ (disjointness)	n	n	$\sqrt{n}$	?
Deutsch-Jozsa (as of 2000)	n	$\log n$	$\log n$	$\log n$?
RAZ (Ron Raz's problem)		$n^{1/4} \log^{-1} n$	$\log n$	

There are others such as matchings.

9.3 Fingerprinting

9.3.1 Classical

Yao (1979) introduced a simplified model in which A and B have a set of bits x and y and there's a trusted referee who computes $f(x,y)$. The constraints are that there is no communication between A and B, the referee cannot send backwards to A and B, and the referee must compute f with bounded error.

For this example, we'll use the equality function. We want functions on x and y with few collisions. Usually we'd use universal hashes, but it can be done with codes, too. There exists an $n \to m$ code with the following property: $\{E(x) \in \{0,1\}^m | x \in \{0,1\}^n, m = cn\}$ has minimum distance $\geq (1-\delta)m$. (The Justensen codes satisfy this property for any $c \geq 2$ and sufficiently large n with $\delta \leq \frac{9}{10} + \frac{1}{15c}$.) Let $E_i(x)$ by the i^{th} bit of $E(x)$.

Assume that A and B share a secret random bitstring $K \in \{0,1\}^{\log m}$. Then A and B send $E_k(x)$ and $E_k(y)$ respectively. We claim that $Prob[E_k(x) \neq E_k(y) | x \neq y] \geq 1-\delta$ (sending one bit). Now we use boosting by repeating it r times, giving $\Pr[\text{success}] \geq 1-\delta^r$.

The problem is we need a secret key. Andy Yao conjectured we need $\Omega(\sqrt{n})$ without a secret key, and it was proven in 1996 by Ambainis (motivated by the quantum case) and separately by Neuman and Szegedy.

(There's another variant in which the referee isn't allowed to figure out anything about x and/or y other than $f(x,y)$.)

9.3.2 Quantum

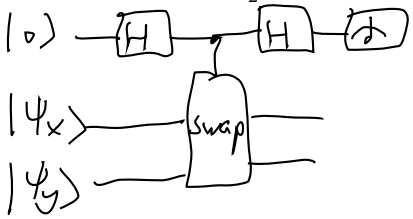
We want to replace k with $O(\log n)$ qubits.

The first theorem is that we can find enough states which are far apart by a bounded amount such that there are no (or few) collisions. Specifically, $\exists\ 2^{2^{O(m)}}$ states $|\psi_x\rangle$ on $O(m)$ qubits (specifially $cm+1$ qubits) such that $\langle\psi_x|\psi_{x'}\rangle \leq \delta$ for $x \neq x'$

Proof: Let $|\psi_x\rangle = \frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} |E_k(x)\rangle|k\rangle$. As a check, $\langle\psi_x|\psi_x\rangle = 1$ trivially. For $x \neq y$,

$$\begin{aligned} |\langle\psi_x|\psi_y\rangle| &= \frac{1}{m}\sum_{k,k'}\langle k'|k\rangle\,|\langle E_{k'}(y)\,|E_k(x)\rangle| \\ &= \frac{1}{m}\sum_{k,k'}|\langle E_k(y)\,|E_k(x)\rangle| \\ &\leq \frac{m\delta}{m} = \delta \end{aligned}$$

The second theorem is that we can do a good enough equality test between this limited set of quantum states. Given two states $|\psi_x\rangle$ and $|\psi_y\rangle$ such that either $|\psi_x\rangle = |\psi_y\rangle$ or $\langle\psi_x|\psi_y\rangle \leq \delta$, there is a test with probability of error $\leq \frac{1+\delta^2}{2}$.



The states are

$$\begin{aligned} |0,\psi_x,\psi_y\rangle &\to (0+1)\,\psi_x\psi_y \\ &\to 0\psi_x\psi_y + 1\psi_y\psi_x \\ &\to (0+1)\,\psi_x\psi_y + (0-1)\,\psi_y\psi_x \\ &\to 0\,(\psi_x\psi_y + \psi_y\psi_x) + 1\,(\psi_x\psi_y - \psi_y\psi_x) \end{aligned}$$

and $\Pr[\text{measure 1}] = \frac{1}{4}|\langle\psi_x\psi_y - \psi_y\psi_x|\psi_x\psi_y - \psi_y\psi_x\rangle| \geq \frac{1+\delta^2}{2}$.

These two theorems give the protocol directly. We now boost to get the fingerprinting algorithm.

9.4 Digital signature schemes

We have two parties, Alice and Bob. Alice has a message m. She would like to produce a signed message (m, s_m) such that B can verify that it came from Alice and so that Bob can send it to a third party who can also verify it. This is called "transferrable message authentication." This is done with a public/private key pair, where the public key is used to verify the signature but the private key is needed to produce the signature.

We need the following properties:

- Unforgeable (classically, we'll only ever get this for a computationally bounded adversary).
- Non-repudiatable – Alice can never convince anyone she didn't sign the message.
- Efficient – public keys should be reusable.

There's a simple scheme due to Leslie Lampart (1979) satisfying the first two properties. We have a public one-way function $f(x)$. The generator makes a pair (k_0, k_1) for the private key, and the public key is $(0, f(k_0))$ and $(1, f(k_1))$. The signature is (b, k_b).

For example, let $f([x, y]) = xy$. Then k_0 and k_1 are so large that no one can factor the products. (No one would ever do it this way.)

The quantum analogue is to use the states $|\psi_{k_0}\rangle$ and $|\psi_{k_1}\rangle$ instead of $f(k_0)$ and $f(k_1)$, and it is secure against a computationally unbounded adversary.

10 Quantum games

10.1 History

The field of game theory was started in the 1940s by von Neumann and Oskar Morgenstein.

Game theory is multiperson decision theory in which decision processes are analyzed assuming each player plays rationally to maximize rewards.

Albert Tucker (1950s) invented the Prisoner's Dilemma. Al and Bob allegedly committed a crime and the police interrogate them individually. We know that someone will be sent to jail. Al and Bob could each cooperate (deny the crime and not blame the other) or defect (admit the crime and blame the other). The police will assign sentences ($r = 3$ is reward, $t = 5$ is temptation, and $s = 0$ is the sucker's payoff):

Years in jail (Al, Bob)	Cooperate	Defect
Cooperate	3,3	0,5
Defect	5,0	1,1

(For some reason, we intentionally give a penalty for defecting.)

The dominant strategy is one which earns a player a larger payoff than any other, *regardless* of what the other player does. Which strategy is dominant depends on whether the payoffs are taken to be positive or negative. In any case, the dominant strategy is not optimal.

As a second example, suppose we have two competing companies: Shor's widgets and Chuang's gadgets. The pricing options are:

			Shor's price	
	Profit	\$1	\$2	\$3
	\$1	0,0	50,-10	40,-20
Chuang's price	\$2	-10,50	20,20	90,10
	\$3	-20,40	10,90	50,50

There is no dominant (pure) strategy. There is a Nash equilibrium at (\$1,\$1), though.

A Nash Equilibrium is a set of strategies where no play had an incentive to change his/her action. This is a self-enforcing agreement.

For a third example:

	L	R
U	8,8	0,6
D	6,0	7,7

Now there are two Nash Equilibria. They may not be unique.

For the fourth, there are three players (payoffs are UD, LR, AB):

A	L	R
U	0, 0, 10	-5, -5, 0
D	-5, -5, 0	1, 1, -5

and

B	L	R
U	-2, -2, 0	-5, -5, 0
D	-5, -5, 0	-1, -1, 5

The pure strategy Nash equilibria are ULA and DRB. DRA, however, has interesting properties: if UD and LR collude, then they can drive a cycle DRA →DRB → ULB →ULA → DRB. So Nash equilibria only capture unilateral optima.

10.2 The PQ Penny flip over game

Picard and Q lifelong enemies and they are tossing a coin. The coin starts out tails. First Q gets to flip (turn over) the coin, the Picard can decide whether to flip it, then Q decides again. Q wins on on heads at the end (+1). (Neither play can look at the coin.)

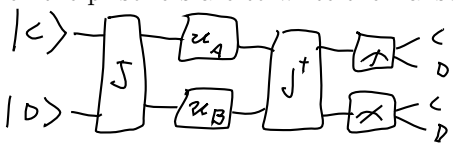
P,Q	NN	NF	FN	FF
N	-1	1	1	-1
F	1	-1	-1	1

There's no pure strategy Nash equilibrium. The mixed strategy Nash equilibrium is for each player to play uniformly at random (there are others, too).

But if Q is quantum and the coin is a qubit. Q plays H. Picard is classical and plays X or I. Then Q plays H and the result is always heads.

10.3 The quantum prisoner's dilemma

There is still no communication between prisoners. Define the states $|C\rangle$ and $|D\rangle$ as the basis for a Hilbert space of decisions (there is one of these qubits per player). In this model, there are two pieces of paper on which the prisoners are to write their answers.



$$J = \frac{II + iXX}{\sqrt{2}} = \exp i\frac{\pi}{4} XX$$

With operators $\hat{C} = I$ and $\hat{D} = X$, note that $\left[J, \hat{C}\hat{C}\right] = \left[J, \hat{D}\hat{D}\right] = 0$, so this is in some sense equivalent to the classical version.

The payoffs are $\$_A = rP_{cc} + pP_{DD} + tP_{DC} + sP_{CD}$ and $\$_B = rP_{cc} + pP_{DD} + sP_{DC} + tP_{CD}$.

The payoff matrix ends up being:

row,col	$\hat{C}$	$\hat{D}$	Z
$\hat{C}$	3,3	0,5	1,1
$\hat{D}$	5,0	1,1	0,5
Z	1,1	5,0	3,3

(The calculations look like, for $\hat{Q}\otimes\hat{C}$, $|\psi\rangle = J^\dagger (ZI)\, J = (II - iXX)\,(ZI)\,(II + iXX)\,|CC\rangle = YX|CC\rangle = -|DD\rangle$, etc.) $\hat{Q}\hat{Q}$ is now Pareto optimal and a Nash equilibrium.

Issues: what if the prisoners can do any rotation? What if the police use a different set of J operators? And why should the police help the prisoners?

There is a notion that one could replace the prisoners' trust in the police to just trusing quantum mechanics (and possibly some box).

10.4 The tragedy of the commons

This is a proportional game model. Suppose that some proportion of commuters use cars and some use busses. If everyone uses a car, we imagine that it's slow for everyone and, if almost everyone uses busses, the busses go really fast but the cars go faster.

The dominant strategy is for everyone to drive, but that is a bad outcome.

More realistically, suppose that the busses become faster than the cars at some point ($\geq p$ percent of the commuters use cars), due to carpool lanes or whatever. Then the crossover point is a Nash equilibrium (mixed in this case).

We'd like for everyone to take the bus, but the tragedy is that that won't happen.

Traditionally, we try to solve this problem with third-party regulators (toll booths, for example).

10.5 A better scenario: the public goods game

Suppose I give everyone $\$y$ and then pass around a "birthday pool" hat and ask for contributions (of $c_k \leq y$) each. The birthday pool earns interest for awhile and then gets redistributed evenly.

We have n, the number of players; y, the initial endowment; $c_k \leq y$ the individual contribution; and $1 < a < n$, the public gain. The individual payoff is

$$\$_k = (y - c_k) + \frac{a}{n} \sum c_k,$$

and the Nash equilibrium is $c_k = 0 \forall k$. The proof is trivial.

One way to improve it would be to make public the sequence c_k to add social pressure to contribute. This only works for small groups.

Suppose we want to contribute to the Support Madonna for Life fund so that she can continue getting music and so that we can all listen to her music for free. But we need an administrator of the fund and we don't know the people who might or might not contribute.

10.6 Quantum games

This is a scenario that has been worked out.

We have a vendor who makes $J = \frac{I^{\otimes n} + iX^{\otimes n}}{\sqrt{2}}$ boxes. (Open question: can these boxes be verified?)

We start in the state $|c\rangle^{\otimes n}$, run through a J box, then each player applies a controlled unitary gate, send it back through a $J^\dagger$, and the results are all measured. The individual payoff is $\$_k = y - \langle c_k \rangle + \frac{a}{n} \sum \langle c_k \rangle = \langle y \rangle = 1$. The classical Nash equilibrium has $\sum \langle c_k \rangle = 0$. In the quantum scenario, $\langle \$ \rangle = \frac{1+a}{2}$. This is better than the classical result.

For pairwise entanglement, $\langle \$ \rangle = a - 2^{-(n-1)} \approx a$ (quant-ph 0301013).

11 Unconitional security of quantum key distribution

11.1 Perspective

There's only one classical perfect encryption scheme, the Vernam cipher (aka one-time pad). Alice and Bob share a key k which is (at least) as long as the message. A sends $m \oplus k$ and B decrypts with $m = m \otimes k \otimes k$.

This is secure, because $I(m \oplus k; m) = H(m \oplus k) - H(m \oplus k|m) = H(m \oplus k) - H(k) = 0$.

We would like to distribute keys for the Vernam cipher securely, such that $I(\text{eve}; \text{key}) < 2^{-l}$ where l is the number of physical resources used.

11.2 BB84

A generates a random qubit ($|0\rangle$ or $|1\rangle$) and a random classical bit b. B generates a random bit b' and measures a', as follows:

Then A announces b and A and B only keep bits (a, a') where $b = b'$.

There's a theorem that information gain implies disturbance. In any attempt to distinguish non-orthogonal states $|\psi\rangle$ and $|\phi\rangle$, information gain is only possible at the expense of disturbing the states.

Proof:

$$\begin{aligned} |\psi\rangle|u\rangle &\rightarrow |\psi\rangle|v\rangle \\ |\phi\rangle|u\rangle &\rightarrow |\phi\rangle|v'\rangle \text{ (want } |v'\rangle \neq |v\rangle) \\ \langle\psi|\phi\rangle\langle u|u\rangle &= \langle\psi|\phi\rangle\langle v|v'\rangle \end{aligned}$$

11.3 History

There were several very complicated attempts to prove security, and then Shor and ??? were able to prove it simply.

One problem is the idea of a collective attack, where Eve is content to learn a little bit about the key while slipping under Alice and Bob's threshold for detection.

Shortly after proposing this protocol, B and B came up with a very simple protocol for bit committment, but that was proven insecure.

11.4 The EPR protocol

There's another protocol for QKD using EPR pairs. Alice makes an EPR pair, measures one half in a random basis (X or Z), sends the other half to Bob (via Eve), who again makes a measurement, randomly in the X or Z basis. A announces her basis, and then Alice and Bob check some pairs at random and use them to bound Eve's errors. Then A and B use entanglement purification to fix up the rest of them, measure them, and get the key.

Define the privacy $P = \sup_{\text{strategies}} [I(B;A) - I(E;A)] \geq S\left(\rho^B\right) - S\left(\rho^E\right) = I_Q(\rho;\varepsilon)$, the coherent information.

We need to bound Eve's information and then to purify entanglement to reduce it. The most general ε_{eve} is, as in error correction, some probability each of I, Z, X, or iY. Then the resulting EPR pair is $\beta_{00} = 00 + 11$, $\beta_{10} = 00 - 11$, $\beta_{01} = 01 + 10$, or $\beta_{11} = 01 - 10$ respectively.

Define the projectors $\Pi_{bf} = |\beta_{01}\rangle\langle\beta_{01}| + |\beta_{11}\rangle\langle\beta_{11}|$ and $\Pi_{pf} = |\beta_{10}\rangle\langle\beta_{10}| + |\beta_{11}\rangle\langle\beta_{11}|$. Note that $[\Pi_{bf}, \Pi_{pf}] = 0$. Now we measure $\Pi_{bf}, 1 - \Pi_{bf}$ or $\Pi_{pf}, 1 - \Pi_{pf}$ at random. By the Chernoff inequality, if we start with $2n$ bits, measure n of them, and get Δ ones, $\Pr[|\Delta - \mu| > \epsilon] < 2^{-O\left(\epsilon^2 n\right)}$ as $n \to \infty$.

The purification step is to assume that all the remaining errors are due to Eve's meddling and we expect nt of the remaining bits to have errors. Let $\delta n = n - nt$, the number of error-free EPR pairs we expect to have left. Let E, D be the encoder and decoder from an $[[n, \delta n]]$ QECC. Now Alice makes δn new EPR pairs, encodes them with E, and does noisy teleportation to Bob, who decodes the result. The resulting state ρ is very close to being a perfect set of EPR pairs.

Recall the QECC guarantee: $F\left(\rho, |\beta_{00}\rangle^{\otimes n}\right)^2 \geq 1 - 2^{-l}$. How does this bound $S\left(\rho^E\right)$.

Lemma: High $F \implies$ low S. Specifically, $F(\rho, |\psi\rangle)^2 > 1 - 2^{-l} \implies S(\rho) < (n + l)\, 2^{-l}$.

Proof: If $\langle\psi|\rho|\psi\rangle > 1 - 2^{-l}$, then $\max(\operatorname{eig}\rho) \geq 1 - 2^{-l}$. This means that

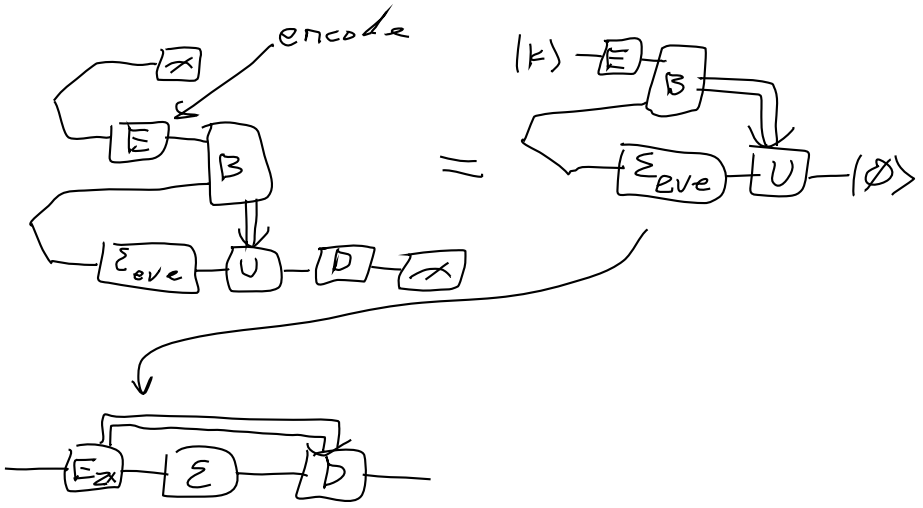
$$S(\rho) < S\begin{bmatrix} 1 - 2^{-l} & & & \\ & x & & \\ & & \ddots & \\ & & & x \end{bmatrix}, x = \frac{2^{-l}}{2^n - 1},$$

so $S(\rho_{max}) \approx (n + l)\, 2^{-l}$ and, by Holevo's theorem, $I(\text{Eve}; A + B) < S(\rho) < O\left(2^{-l}\right)$.

This has flaws: we need efficient codes (we have them now, but we didn't); we need quantum memory; and we need quantum computers. But it turns out that this scheme is very similar to the BB84 protocol, which needs none of these things.

11.5 CSS codes protocol

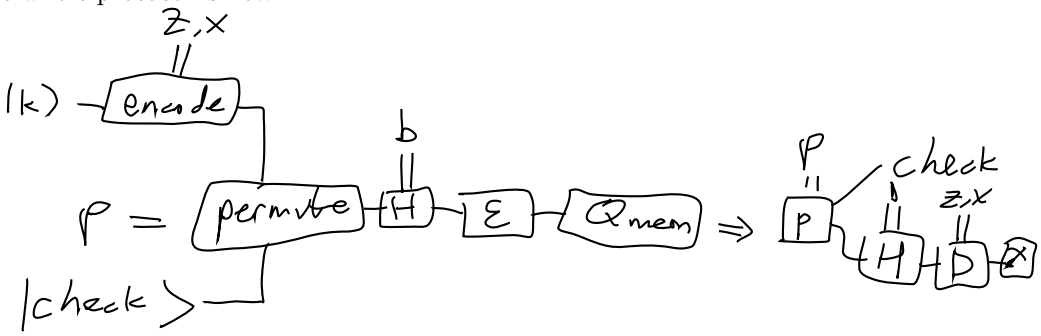
Step 1: EPR, E, and D give a random code. This can be simplified:



Step 2 (due to Shor and Preskill): Use CSS codes to simplify it further. Recall $CSS(C_1, C_2)$ is a $[[n, k_1 - k_2]]$ QECC with states $|\psi_k\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{w \in C_2} |v_k + w\rangle$, so it's a map onto cosets indexed by k. With z and x as random error vectors, we have

$$|\psi_{kzx}\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{w \in C_2} (-1)^{(w+v_k)\cdot z} |v_k + w + x\rangle.$$

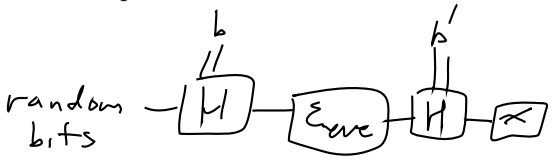
The whole protocol is now:



But Bob doesn't care about Z errors, so he can skip the Z correction and decode classically (remember that the CSS codes have their stabilizers partitioned into Z and X). This means that he doesn't need a quantum memory or a quantum computer, and he doesn't need to know z!

So Alice can skip sending z and produce the mixed state (before interleaving) $\rho_{kx} = \frac{1}{2^n} \sum_z |\psi_{kzx}\rangle\langle\psi_{kzx}|$. Applying this to the CSS codewords will cause all the pieces with nonzero z to average over the complex circle and cancel. All that's left is $\rho_{kx} = \frac{1}{|C_2|} \sum_{w \in C_2} |v_k + w + x\rangle\langle v_k + w + x|$. This is a diagonal mixture of random codewords plus random codes, and it entirely fills the Hilbert space (assuming you chose good codes). So it's just an entirely random bitstring. This is trivial to encode. But these random bits are indistinguishable from the check bits, so no interleaving is required and the selection of which bits to use can be made afterwards. Now the quantum memory can be removed, too, by doubling the number of bits, so B performs H with b' randomly and discards if $b \neq b'$, eliminating the quantum memory as well.

The final protocol is:



A+B discard if $b \neq b'$, compare check bits, and obtain x (which A knows) and $x + \epsilon$ (which B knows), and abort if $wt(\epsilon)$ is too large. Then A and B do privacy amplification:

A announces $x - v_k$; B computes $x + \epsilon - x + v_k = v_k + \epsilon$ and computes the coset index of $v_k = k$.

What would happen if you used a different set of codes? For example, Alice never sends $0+i1$ or $0-i1$. Could this be improved?

What if there's a small quantum computer (say 20 qubits) available?

There are now other protocols that use other states (e.g. coherent states). Are they provably secure like this?

12 Quantum complexity

12.1 Intro

[Missed a bunch of definitions of BQP, NP, P, etc.] It's assumed that $P \subseteq BQP \subseteq NP$, but this is unproven. We can prove a bunch of complexity results about oracle problems, but we know essentially nothing about non-oracle problems.

Complexity theorists can study things like the Grover problem (it takes $c\sqrt{N}$ queries to find a needle in a haystack of N items, and this bound is tight in both directions).

There are things called quantum proofs: A might want to prove something to B via B communicating with A and being convinced of the truth of the proof, even if B couldn't subsequently prove it to anyone else. These proofs might be significantly shorter using quantum information.

The class QMA (Quantum Merlin-Arthur) is the set of all problems that admit short quantum proofs. For example, suppose we have a subgroup $H \subseteq G$ described by a list of generators. If we'd like to prove that $x \in H$, we could give a polynomially-long string of generators which, when multiplied, give x. But suppose we'd like to prove that $x \notin H$. The naive approach (listing all of H) takes exponential time. But Watrous (2000) showed that the quantum state $|H\rangle = \frac{1}{\sqrt{|H|}} \sum_{h \in H} |h\rangle$ allows a solution. The verifier could produce $\frac{1}{\sqrt{2}} [|0\rangle|H\rangle + |1\rangle|Hx\rangle]$, then apply H to the first qubit and measure. If $x \in H$, then you measure $|0\rangle$. If, on the other hand, $x \notin H$, then you get $|0\rangle$ and $|1\rangle$ with equal probability. You'd also need to verify that the state $|H\rangle$ is prepared honestly, but this can be done.

12.2 How much classical information is in an n-qubit state?

One question might be how much classical information is needed to describe it exactly. In this case, the answer is ∞. With an ϵ-mask, the answer is $\left(\frac{c}{\epsilon}\right)^n$. To teleport it exactly, we need $2n$ bits. We can only store n classical bits retrievably in n qubits.

12.3 An example theorem

Theorem: Given an n-qubit state $|\psi\rangle$ (it could also be mixed, but for simplicity we'll assume it's pure), and given m 2-outcome measurements $E_1, \cdots, E_n$. I can give you $O(n \log n \log m)$ classical bits from which you can approximate $\langle\psi|E_i|\psi\rangle$. (You'd presumably need $O(nk \log n \log m)$ bits to approximate strings of k measurements.) It might take exponential time to prepare the string and to interpret the results, though.

The class $BQP/qpoly$ is the class of problems for which a piece of fixed quantum piece of e advice of polynomial size allows a solution in BQP. It turns out that $BQP/qpoly \subseteq PostBQP/poly = PP/poly$. There's another result that proves $PostBQP = PP$, which can be used to give much simpler proofs about PP. (PP is the class for which there is a classical algorithm that gets the right answer with probability better than $\frac{1}{2}$, but with no bounded distance away from $\frac{1}{2}$. $PostBQP$ is BQP plus the ability to post-select.)

To prove the theorem above, amplify the probabilities of the outcomes to near 0 or 1 (repeated measurements). Then you take the totally mixed state and post-select on getting the right results. Let $N = cn \log n$. It takes $O(\log m)$ bits to describe one of these post-selections (the outcome and which E_i^*. It can be shown that $O(N)$ steps suffice to describe the statistics well enough (i.e. the measurements all give the right results).

Proof: Let $I = \frac{1}{2^n} \sum_{i=1}^{2^n} |\phi_i\rangle\langle\phi_i|$ and let $|\phi_1\rangle = |\psi\rangle$, the desired state. We've picked measurements that are nearly deterministic on $|\phi_1\rangle$ but maximize the chance of getting the next one wrong, so the state isn't damaged much each time. Let $q(\rho)$ be the probability of succeeding on all $k = O(N)$ measurements with ρ as the initial hypothesis. Then $q(|\psi\rangle) \geq 0.9$ and $q(I) \geq \frac{0.9}{2^N}$ and the result follow somehow (there are a few details missing in here).

If we're have enough different measurements, then we can just pick measurements at random.

13 Class presentations

13.1 Adiabatic QC (David Gossett)

13.1.1 Adiabatic QC

AQC is a quantum computation technique on n qubits. We compute with a Hamiltonian

$$H(s) = (1-s) H_B + sH_P$$

where $0 \leq s \leq 1$ is a parameter (typically $s = \frac{t}{T}$ where T is the time that the computation runs for). H_B is chosen to have an easy-to-prepare ground state ($H_B = \sum_{i=1}^{n} \left(\frac{1-\sigma_x}{2}\right)$), and the adiabatic theorem guarantees that we find a ground state of H_P when $T \sim \frac{\epsilon}{g^2}$ (or longer), where $g = \min_s \{$difference between lowest two eigenvalues of $H(s$ and ϵ is a matrix element. We'll cleverly choose H_P such that its ground state solves an interesting problem.

13.1.2 Group representations

AQC on an Abelian group In an Abelian group, every character can be labelled by a group element, and the characters themselves form a group under $\chi_h \chi_g = \chi_{(hg)}$, which we call the dual group $\hat{G}$. The dual group is isomorphic to the original group.

The Fourier transform over the group converts the basis $\{|g\rangle : g \in G\}$ to $\{|\chi_g\rangle : g \in G\}$.

We can define operators $h|g\rangle = |g+h\rangle$ and $\chi_h|g\rangle = \chi_h(g)|g\rangle$.

If we equate $\{1, \sigma_x\} \to \{1, g\}$ and $\{1, \sigma_z\} \to \{1, \chi_g\}$, then H_B is diagonal in $\{|\chi_g\rangle\}$ and H_P is diagonal in $\{|g\rangle\}$. So the adiabatic algorithm moves from a Fourier basis element to a computation basis element.

Non-abelian groups The non-Abelian FT basis is $\{|\rho, i, j\rangle \; s.t. \; 1 \leq i, j \leq \dim \rho\}$. We can write $|\rho, i, j\rangle = \sqrt{\frac{d(\rho)}{|G|}} \sum_{g \in G} (\rho(g))_{ij} |g\rangle$.

We could define a non-Abelian AQC Hamiltonian that moves from the FT basis to the computation basis.

13.1.3 AQC on $G^{\otimes n}$

Let $H_B = \sum H_B^{(i)}$ with $H_B^{(i)} = \sum_{g \in G} (1-g)$. The ground state of this is $|\rho_0, 1, 1\rangle = \frac{1}{\sqrt{|G|}} \sum |g\rangle$, which is the trivial representation. (We could also write a Hamiltonian based on the adjacency matrix of the Cayley graph.) Let $H_P = \sum_h F(h) |h\rangle\langle h|$.

Now we compute $gH_Bg^\dagger = (1-s) H_B + s\tilde{H_P}$ (because $[g, H_B] = 0$). This means that $\tilde{H_P}$ and H_P are equally easy to solve. But $\tilde{H_P} = \sum_h F(h) |gh\rangle\langle gh| = \sum_s F\left(g^{-1}s\right) |s\rangle\langle s|$. This allows us to solve the shifted oracle if the unshifted oracle problem has a large gap.

13.1.4 Hidden shifts

Let $H(s) = (1-s) \sum_i \frac{1-\sigma_x^{(i)}}{2} + s \sum_i \frac{1-\sigma_z^{(i)}}{2}$. Farhi, Goldstone, Gutman, and Nagaj showed that $T \sim O(\sqrt{n})$. This thing minimizes the Hamming weight of $\vec{z}$. Suppose you had an oracle for $f(\vec{z}) = Wt(\vec{z} \oplus \vec{s})$. Then you could find $\vec{s}$ classically in $O(n)$ queries.

It turns out that with a quantum query oracle for $\vec{z} \cdot \vec{s}$ we could compute $\frac{1}{\sqrt{2^n}} \sum_{\vec{z}} (-1)^{\vec{z} \cdot \vec{s}} |\vec{z}\rangle$ in one query and then to a (fully parallel) Hadamard gate to recover $\vec{s}$.

Ike suggests to look for an adiabatic algorithm over the generators of the group instead of over all the group elements.

13.2 Measurement-based QC (Xie Chen)

13.2.1 Intro

[Same as above]

13.2.2 Experimental difficulties

- Entangling gates are probabilistic at best (i.e. there will be lattice defects)
- Entanglement decoheres (so there is a limited lifespan of the circuit)

One was to make the cluster state more robust is to make it the ground state of some Hamiltonian. The standard cluster state requires a five-body Hamiltonian.

There are no condensed matter Hamiltonians where it's easy to prove the existence of a unique ground state with a gap.

13.2.3 Teleportation

With an initial state $CZ|++\rangle$, we can make a Bell-basis measurement and correct the resulting Pauli error.

We can also teleport a unitary operation. If there are four qubits at each vertex in a cluster state and the edges connect one qubit from one vertex to one vertex at the other side, we can do measurement-based QC. This requires four-body measurements.

If we now view each site as a 16-dim particle, we have a two-body Hamiltonian with two-body measurements. With a honeycomb lattice instead, we get 8 dimensions.

The projection of this state onto span $\{|000\rangle, |111\rangle\}$ is the standard cluster state. The projection onto the orthogonal (local) space can be called a "spin $\frac{5}{2}$ Honeycomb lattice state."

We can measure in $\{|011\rangle, |100\rangle\}$, etc., to perform QC.

If we let $H = \sum H_{ab}$ (sum of local projectors), we can prove that the desired state is the unique ground state. (I think these projectors project onto the space orthogonal to everything in the local reduced density matrix.)

13.3 Quantum Compressed Sensing (Hyun Sung Chang)

13.3.1 Intro

The goal is to relate quantum tomography to compressed sensing, both to show that they are analogous and to use optimal strategies from compressed sensing to quantum tomography.

13.3.2 Compressed sensing

Compressed sensing is the idea of capturing sparse signals in some basis with many fewer sensors than the original dimension of the signals. (For example, many images are sparse in the wavelet domain.) More recently (Yair Weiss et al., Proc Allerton Vonf. Comun Control Comput (2007)), compressed sensing has been generalized to deal with a more generic class of signals, lossy schemes even when there are two few sensors, and possibly noise measurements.

The problem is to take a linear measurement

$$y = W^T x + \eta \text{ where } W \in \mathbb{R}^{d \times p} \, (p \leq d) \text{ and } \eta \sim \mathcal{N}(o, rI)$$

and to do optimal decoding, which is, in general, nonlinear. We use an inference process based on $p(x|y)$. We would like the optimal $W^* = \arg\max_{W \in \mathcal{W}} I(x; y)$. This looks like a channel capacity problem except that we get to adjust the channel itself but the source distribution is fixed.

Usually the optimization will be taken w.r.t. a power budget.

13.3.3 Quantum tomography

Quantum tomography is the estimation of a quantum state from a set of measurement outcomes. The literature can be classified in terms of pure states vs. mixed states, discrimination vs. identification, and unambiguous output vs. error-bounded.

The assumption is that we have an ensemble of copies of an n-qubit state with some density matrix. The state is uniquely identifiable if [lost the condition]. There's a paper on "Learnability of quantum states" by Scott Aaronson.

The constraint for the quantum case is the POVM constraint, as opposed to power.

13.3.4 Combining these

Suppose we have a state drawn from $|0\rangle$, $|1\rangle$, $|+\rangle$, and $|-\rangle$. Then we can find two POVM elements that distinguish them nicely, but we can project all of them onto some line such that the single measurement separates them well.

We can decode with MAP (maximum a posteriori) or BLS (Bayesian least-squares).

There are pretty pictures showing that for some input distributions the optimal (classical) compressed sensing approach gives excellent results. The pretty pictures assume exact measurement of the output probability, which is, of course, impossible. But we can just take a bunch of measurements and average. By the CLT, a Gaussian noise model is decent.

Claim: Given n qubits and M possible density matrices, all of them can be distinguished arbitrarily small error with $O(n \log n)$ measurements.

13.4 Adiabatic computing using 2-local Hamiltonians on a line (Brian Pepper)

13.4.1 Hamiltonian problems, BQP, and QMA

There are all kinds of interesting Hamiltonian questions (ground state, etc.) that are BQP-complete or QMA-complete.

The adiabatic theorem says that, given $\mathcal{H} = (1-s)\mathcal{H}_0 + s\mathcal{H}_1$ with $s = \frac{t}{T}$ and $T \sim \frac{1}{\Delta E}$, you stay in the ground state.

QMA is the class of decision problems for which, for each instance in the language, there exists a proof state such that a verifier will accept w.p. $\frac{2}{3}$ and, if not in the language, the verifier rejects w.p. $\frac{2}{3}$ no matter what. (John Watrous proved that you can boost even though it doesn't look like it.)

13.4.2 k-LOCAL r-STATE HAMILTONIAN

Let $\mathcal{H}$ be k-local on r-state particles. The system accepts if $E_0 > \beta$ and rejects if $E_0 < \alpha$, with $\beta - \alpha < \frac{1}{\text{poly}(n)}$.

The 9-state solution is complicated. I'm not going to try to record it. The history state is not the only ground state, but it's the only one accessible from the initial state. Ahoronov et al show that the gap is good, so that $T = O\left(n^4\right)$.

13.5 TQC (Meagan Thompson)

SSB is when a full symmetry group G is broken into a finite subgroup H, and where it settles is called the vacuum expectation value. In three or more dimensions, configuration space has an S_n symmetry, and we expect a 1-d represenatation. In two dimensions, the fundamental group of the configuration space is the n-particle braid group.

Braiding in two dimensions satisfies the Yang-Baxter equation, which says $\sigma_{12}\sigma_{23}\sigma_{12} = \sigma_{23}\sigma_{12}\sigma_{23}$. The problem of determining all solutions to the Yang-Baxter equation is unsolved (i.e. finding all representations that satisfy it). (Peter has never thought about whether there are solutions that are not representations of the braid group.)

There's an R matrix relating to fusion rules (I think) and an S matrix that takes care of change of basis. The Verlinde formula describes the fusion rules.

13.6 Fault tolerant measurement-based QC (Daniyar Nurgaliev)

Traditional fault-tolerance works just fine on measurement-based QC, but the length of the wires will make the thresholds terrible.

It turns out that ZZ and XX syndromes can be measured in the cluster state by simply measuring the spot between two wires in the X basis. There is a code with the right locality properties: the Bacon-Shor code. In three dimensions, the entire syndrome measurement consists of these local measurements on a $5 \times 5 \times 5$ lattice.

The logical Pauli operations are free (you just track them classically). For S and T gates, you need to track ancillas, and, for $CNOT$, you need to route everything.

A T gate can be realized with an ancilla $|0_L\rangle + e^{i\pi/4}|1_L\rangle$, which can be prepared with a cat state. General measurement might be usable to realize this as well (open question).