Quantum error correction and fault-tolerance

May 31, 2021

Online material:

- John Preskill's course at Caltech
- Dan Browne's course at UCL on "Topological Codes and Computation"
- Daniel Gottesman `https://arxiv.org/abs/0904.2557`

1 Lecture 1

Outline of this lecture:

- general motivation,
- classical error correction and fault-tolerance,
- an example: Shor's 9 qubit code,
- correcting arbitrary errors.

1.1 General motivation

A main theme in quantum information science is the protection of quantum information. This is of course extremely relevant in the context of quantum cryptography when we want to prevent an adversary to access some information. Quite often, the adversary will be modeled as "the rest of the universe", or the "environment", which is exactly the same setup as when one wants to protect quantum information for communication (quantum internet), storage (quantum memories) and more crucially *for computation*. If quantum information decoheres during the computation or if every gate of the circuit is a little bit noisy, it's not clear at all how to perform a long calculation and get a meaningful result. For instance, if every gate has fidelity 99%, then the relevant quantum information is gone after about a hundred gates. Running interesting algorithms such as Shor's require much more gates than that. What's the solution? Quantum error correction!

In fact, and quite amazingly, it is possible to develop error correction and fault-tolerant techniques for quantum information. This is far from obvious since the quantum errors seem to belong to a continuum, much like what happens for analog computing, which was abandoned pretty much for this reason. The key difference is that measuring part of the system will collapse the

error onto a finite, discrete set, and it will be sufficient to be able to correct errors from that set. As we will see, quantum error correction is possible *in principle*, but remains quite costly. In theory, one can use quantum error correction and quantum fault-tolerance to perform arbitrarily long quantum computations with reasonable (polylogarithmic in the number of gates) overhead (if the qubits and gates are good enough): this is the *threshold theorem*.

Theorem 1 (Aharonov, Ben-Or '97)**.** *Provided that the noise level is below some constant threshold, a logical circuit using m qubits and containing T gates can replaced by a fault-tolerant circuit using $O(m \operatorname{polylog}(mT))$ qubits.*

In practice, however, the overhead is quite large and finding better techniques is a major open problem is we want to build large-scale universal quantum computers. For instance, breaking cryptographic size RSA key requires a few thousands ideal qubits, but between 10^6 and 10^9 physical (good enough) qubits. This is because qubits and quantum gates are much more noisy than classical bits and gates.

The NISQ era: computing without error correction. In fact, today, quantum error correction has barely been experimentally implemented. Some recent experiments have shown that a protected qubit can store information longer than unprotected ones, but only for restricted noise models. Because of the lack of experimentally available quantum error correcting schemes, a lot of the current research tries to study noisy intermediate scale quantum (NISQ) systems, i.e. systems of about 50 to 100 qubits, without any error correction mechanism. In this setup, one cannot implement Shor and it's interesting to understand whether this kind of machines can do anything useful better than a classical computer (say, for solving approximately optimization problems). This is related to the challenge of *quantum supremacy* (demonstrated by Google in Nov. 2019) which aims at demonstrating a task for which a quantum machine is *provably* much faster than any classical computer. Usually, the tasks for which we can prove this kind of advantage are pretty useless in practice.

In this course, however, we are optimistic and assume that experimentalists will manage to improve qubits as well as quantum gates, and will be able to individually control a large number of qubits. If this is the case, then they will be able to implement quantum error correction and fault tolerance techniques, which will allow them to perform arbitrary long computations. The main condition is that the noise level is below some constant threshold, which is around $0.1 - 1\%$, and that they can control a large number of qubits without increasing the noise level.

A number of challenges faces us:

- large entangled states are fragile: a single qubit decohering or leaking out kills the whole superposition. For instance, if you start with a cat state

  $$\frac{1}{\sqrt{2}}|0\rangle|\text{Alive}\rangle + \frac{1}{\sqrt{2}}|1\rangle|\text{Dead}\rangle$$

  and lose the first qubit (i.e. partial trace), then the remaining state is

  $$\frac{1}{2}|\text{Alive}\rangle\langle\text{Alive}| + \frac{1}{2}|\text{Dead}\rangle\langle\text{Dead}|,$$

  which corresponds to a classical mixture of Alive and dead.

- applying noisy gates just increases the total noise: errors pile up. A similar problem was present in the 50's for classical computing and Von Neumann developed a theory of fault-tolerant computation (that very much inspired the quantum version!). Fortunately, transistors quickly became so good that this theory became essentially pointless.
- as already mentioned, the possible errors seem to form a continuum in the quantum case. How do you deal with that?
- the no-cloning theorem says that an operation that does $|\psi\rangle \to |\psi\rangle|\psi\rangle$ is not unitary and therefore forbidden. How do you protect quantum information without the ability to copy it, as you would do in the classical case (using the repetition code, say)?
- it is not possible to simply measure the output state to determine what it is since measuring the state will disturb it, and collapse it on a basis element.

1.2 Classical error correction and fault-tolerance

Before discussing quantum error correction, let's discuss classical error correction first. A natural classical noise model is the *binary symmetric channel*, corresponding to a channel from Alice to Bob (or Alice at time $t = 0$ to Alice at time $t = 1$) where each bit is independently flipped with probability $p \in [0,1]$. The simplest solution to protect information is to add redundancy, for instance by repeating any single bit three times and decoding[1] by taking a majority vote:

- encoding: $0 \to 000 =: \bar{0}$, $1 \to 111 =: \bar{1}$. Here, $\bar{0}$ and $\bar{1}$ form a basis for the *logical* bit, while 000 and 111 refer to *physical* bits.
- channel: each bit is flipped independently with probability p,
- decoding: majority vote: $\{000, 100, 010, 001\} \to \bar{0}, \{111, 011, 101, 110\} \to \bar{1}$.

This scheme produces the correct result if the channel flipped 0 or 1 bit (and detects that an error occurred if 1 or 2 errors occurred). The error probability drops from p for the physical bit to $1-(1-p)^3-3p(1-p)^2 = 1-3p^2+2p^3$ for the logical bit. This reduces the error rate provided that $p < \frac{1}{2}$. If $p = \frac{1}{2}$, then the channel cannot transmit any information, and if $p > \frac{1}{2}$, one should simply flip the bit to recover the case $p < \frac{1}{2}$.

This coding strategy protects against bit flips, but is quite inefficient. Much better schemes exist, but this is a good starting point to discuss quantum coding.

Classical fault-tolerance theorem. In the scheme above, we assumed that the encoding and decoding could be implemented perfectly. In other words, that they are noiseless. While this is a reasonable assumption today, it wasn't the case in the early days of computing with vacuum tubes, which were faulty. Skeptics back then would argue that any sufficiently long computation was doomed to fail since errors would necessarily occur in encoding/decoding circuits. John von Neumann proved that this reasoning was actually incorrect.

[1] The word "decoding" can refer to 2 different operations. The true definition refers the map that is the inverse of the encoding operation: it maps an encoded logical state to an unencoded physical state. However, it is very usual to use the word "decoding" to also include the error correction procedure. In that case, it refers both to the action of trying to correct the error that occurred and in a second time, to return the unencoded state.

He showed that if logic gates (AND, OR, NOT) fail with some independent probability ε, then provided that ε is small enough, it is possible to build a reliable circuit for any Boolean function f. Moreover, the circuit is only reasonably larger than the circuit for f built of perfect gates.

One question though: what happens if the final gate is faulty with probability ε? It seems that you can never get a correct answer with probability greater than $1 - \varepsilon$! A solution is to encode the final result into a long string or to repeat the computation several times and take a majority vote (assumed to be error-free, because simple to implement).

To prove the threshold theorem, one can use the 3-bit repetition code recursively, where each physical bit is actually replaced by the logical bit of another layer of repetition code. Even if each operation is noisy, there exists a threshold below which each level of encoding leads to a decrease of the logical error rate.

It turns out that the classical fault-tolerance theorem ended up being useless when vacuum tubes were replaced by transistors because the error rate for logic gates became ridiculously small (much smaller than the inverse of the number of gates in any useful computation).

The same thing might occur one day with quantum computing and topological quantum computing might be an idea to get there, but we are currently *very, very* far from that situation. And it is therefore crucial to develop a theory of quantum error correction and quantum fault-tolerance.

1.3 A first example: Shor's code

We want to generalize the 3-bit repetition code to the quantum setting. The goal is as follows: Alice has a qubit in a pure state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and wants to send it to Bob (or to keep it in a noisy memory for some time). Here, we suppose that Alice doesn't know α and β. If she knows the qubit exactly, she could just send the description to Bob by classical means who would reconstruct it. But this wouldn't be efficiently when sending n-qubit states (since you get 2^n amplitudes...) and in practical situations, Alice doesn't know the values of α and β.

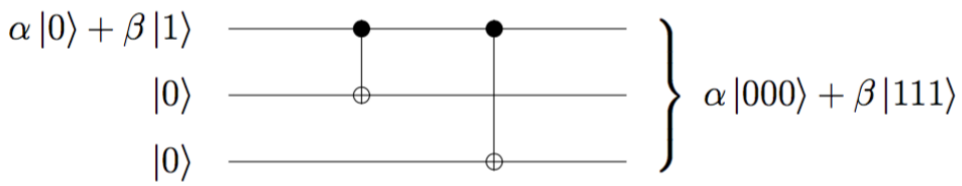
For concreteness, let us first consider a restricted noise model similar to the binary symmetric channel where a qubit is flipped independently with probability p. In other words, with probability $1 - p$, then channel acts as the identity, and with probability p, it applies a bit-flip error represented by $X = \sigma_X = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$. This quantum channel is called the *bit-flip channel* and is described by the admissible operation

$$E_{\text{bit}}(\rho) = (1 - p)\rho + pX\rho X.$$

First, the no-cloning theorem prevents us to apply the map $|\psi\rangle \to |\psi\rangle|\psi\rangle|\psi\rangle$ since it is not unitary. A better approach is to perform the following encoding

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \mapsto \alpha|000\rangle + \beta|111\rangle =: \alpha|\bar{0}\rangle + \beta|\bar{1}\rangle = |\bar{\psi}\rangle.$$

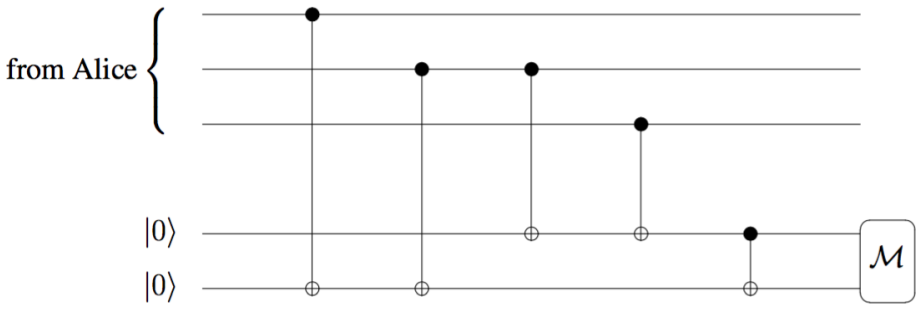
This encoding can be realized with a simple qubit with 2 CNOT gates.



The 3-qubit state after encoding is described by a density matrix (a pure state in fact) on $(\mathbb{C}^2)^{\otimes 3}$ and the quantum channel is described by the admissible operation:

$$\begin{aligned} E_{\text{bit}}^{\otimes 3}(\rho) =& (1-p)^2\rho + p(1-p)^2(X_1\rho X_1 + X_2\rho X_2 + X_3\rho X_3) \\ &+ p^2(1-p)(X_1X_2\rho X_1X_2 + X_1X_3\rho X_1X_3 + X_2X_3\rho X_2X_3) + p^3 X_1X_2X_3\rho X_1X_2X_3. \end{aligned}$$

How should we decode the state and perform the correction?? Measuring the whole state in the computational basis is a bad idea since one will get a basis state and the information about $|\psi\rangle$ will be destroyed. Rather, we want to copy what we did in the classical case to notice that an error had occurred: for this, we compare the values of the 3 bits pairwise, i.e., we measure parities. If they all agree, then we conclude that no error occurred. If some values disagree, we perform a correction.



Let us consider what this circuit does on an example, for instance if the second qubit was flipped (the error is X_2):

$$\begin{aligned} (\alpha|010\rangle + \beta|101\rangle)|00\rangle &= \alpha|010\rangle|00\rangle + \beta|101\rangle|00\rangle \\ &\mapsto \alpha|010\rangle|10\rangle + \beta|101\rangle|10\rangle \\ &= (\alpha|010\rangle + \beta|101\rangle)|10\rangle. \end{aligned}$$

Measuring the last two qubits yields the outcome 10 with certainty. This output string is called the *syndrome* and it tells us that a bit-flip occurred on the second qubit (qubit 10 in binary). Bob can therefore correct the error by applying $X = \sigma_X$ to qubit 2:

$$\alpha|010\rangle + \beta|101\rangle \mapsto \alpha|000\rangle + \beta|111\rangle = |\overline{\psi}\rangle.$$

Finally, he can apply the inverse of the encoding circuit to recover the initial qubit

$$\alpha|000\rangle + \beta|111\rangle \mapsto \alpha|0\rangle + \beta|1\rangle.$$

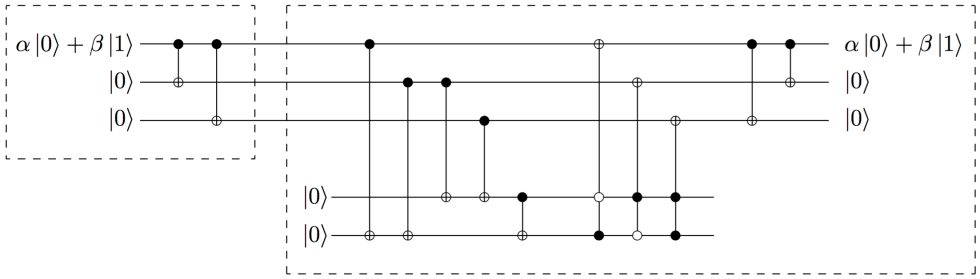
This procedure works if at most one bit flip occurs:

- classical states $|000\rangle, |111\rangle \longrightarrow$ syndrome 00,
- classical states $|100\rangle, |011\rangle \longrightarrow$ syndrome 01,
- classical states $|010\rangle, |101\rangle \longrightarrow$ syndrome 10,

- classical states $|001\rangle, |110\rangle \longrightarrow$ syndrome 11.

We will see later that measuring the syndrome is equivalent to measuring the value of the observables Z_1Z_2 and Z_2Z_3, which check the parity of the first and second qubits, and of the second and third qubits. (With the convention above, it is really the observables $\frac{1}{2}(\mathbb{1} + Z_2Z_3)$ and $\frac{1}{2}(\mathbb{1} + Z_1Z_3)$.)

The overall circuit, including encoding on Alice's side and syndrome measurement, error correction and decoding on Bob's side is summarized here:



Similarly as in the classical case, the error probability goes from p for unencoded physical qubits to $3p^2 - 2p^3$ for the encoded logical qubit. Here, we have assumed that all the gates inside Alice and Bob's labs are perfect.

Going beyond bit-flips.

Of course, we've already seen that qubits are prone to more general errors than simply bit-flips. Another very natural error is the phase-flip, corresponding to an application of the matrix $Z = \sigma_Z = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$. It should be clear that our 3-qubit code does not protect against this type of errors. For instance, a phase-flip on any of the three physical qubits (i.e. Z_1, Z_2 or Z_3) yields

$$\alpha|000\rangle + \beta|111\rangle \mapsto \alpha|000\rangle - \beta|111\rangle.$$

As before, if we suppose that the probability for an individual phase-flip is $0 < p < \frac{1}{2}$, then the probability that the logical qubit is corrupted (corresponding to one or three corrupted physical qubits) is

$$3p(1-p)^2 + p^3 > p.$$

If we simply wanted to protect against phase-flip errors only, then the following encoding would work:

$$\alpha|0\rangle + \beta|1\rangle \mapsto \alpha|{+}{+}{+}\rangle + \beta|{-}{-}{-}\rangle.$$

To do that it, one can simply perform the same encoding circuit as before, followed by a Hadamard transformation on the output qubits. This is because $H|0\rangle = |+\rangle$ and $H|1\rangle = |-\rangle$. Then, exactly the same analysis as before goes through, and you get a coding scheme that can correctly deal with 0 or 1 phase-fip error ... but unfortunately not with a single bit-flip error.

Is there a way to protect against both bit-flip and phase-flip errors at the same time? Not with a 3-qubit code. But one can with a simple 9-qubit code: Shor's code. The idea relies on *concatenation*:

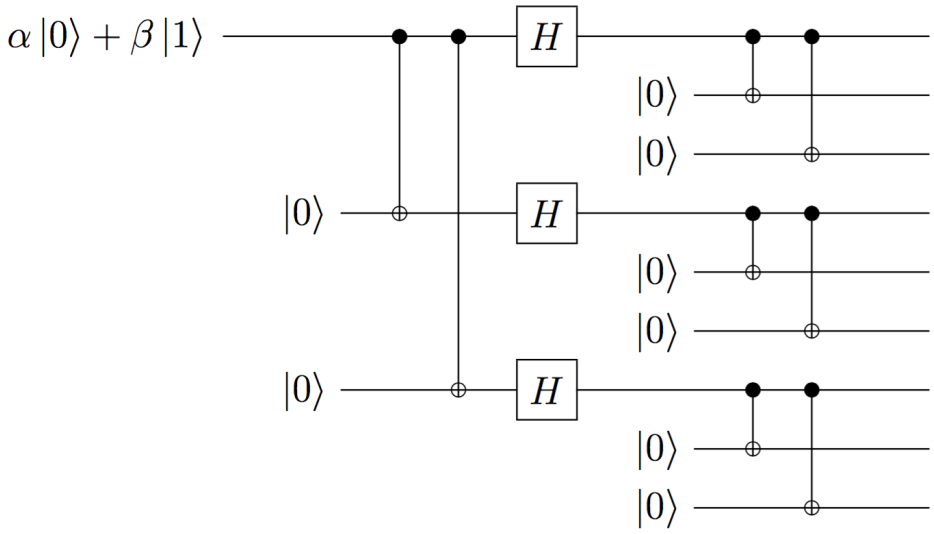
one first applies the code that protects against phase-flips then one encodes the resulting logical qubits with the code that protects against bit-flips. The encoding of a single-qubit $\alpha|0\rangle + \beta|1\rangle$ is given by

$$\begin{aligned}
\alpha|0\rangle + \beta|1\rangle &\mapsto \alpha|{+}{+}{+}\rangle + \beta|{-}{-}{-}\rangle && \text{(first level of encoding)} \\
&= \alpha\frac{1}{2\sqrt{2}}(|0\rangle + |1\rangle)^{\otimes 3} + \beta\frac{1}{2\sqrt{2}}(|0\rangle - |1\rangle)^{\otimes 3} \\
&\mapsto \alpha\frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle)^{\otimes 3} + \beta\frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle)^{\otimes 3} && \text{(second level of encoding)}
\end{aligned}$$

The logical qubits consist of 9 physical qubits:

$$\begin{aligned}
|\bar{0}\rangle &= \left(\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)\right)^{\otimes 3} \\
|\bar{1}\rangle &= \left(\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)\right)^{\otimes 3}
\end{aligned}$$

The encoding circuit is obtained by concatenating the two encoding circuits of the two 3-qubit codes.



To decode, Bob applies successively the decoding procedures of the two 3-qubit codes:

(i) he first considers the three blocks of three qubits and each such block is an encoding of one of the 3 qubits of the state $\alpha|+\rangle^{\otimes 3} + \beta|-\rangle^{\otimes 3}$. He then corrects bit flips as before, and obtains 3 qubits (keeping only the first one in each block). Provided that at most one bit-flip occurred in each block, then the 6 other qubits are all in the state $|0\rangle$ and can be discarded.

(ii) Bob now has the same three qubits as he would when encoding with the code that protects against at most a single phase-flip. He can decode it as before, and provided there is at most a single phase-flip, he recovers the correct state $\alpha|0\rangle + \beta|1\rangle$.

Overall, the qubit is recovered if there was at most a single bit-flip or a single phase-flip.

Theorem 2. *The 9-qubit Shor code protects against any of the four Pauli errors $\mathbb{1}, X, Y = iZX, Z$ occurring on a single qubit.*

Example 3. *Suppose the error XZ occurs on qubit 6. The encoded state becomes*

$$\frac{\alpha}{2\sqrt{2}}(|000\rangle + |111\rangle)(|00\mathbf{1}\rangle - |11\mathbf{0}\rangle)(|000\rangle + |111\rangle) + \frac{\beta}{2\sqrt{2}}(|000\rangle - |111\rangle)(|00\mathbf{1}\rangle + |11\mathbf{0}\rangle)(|000\rangle - |111\rangle)$$

The syndrome measurement for the code protecting against bit-flips yields 00, 11 and 00 for the 3 blocks of 3 qubits. Bob infers that there is no bit-flip error in the first block, a bit-flip on the third qubit of the second block and no bit-flip error in the third block. Applying the correction X_6 yields

$$\frac{\alpha}{2\sqrt{2}}(|000\rangle + |111\rangle)(|000\rangle - |111\rangle)(|000\rangle + |111\rangle) + \frac{\beta}{2\sqrt{2}}(|000\rangle - |111\rangle)(|000\rangle + |111\rangle)(|000\rangle - |111\rangle)$$

Finishing the decoding of the first code (by applying the inverse of the encoding circuit) gives

$$\frac{\alpha}{2\sqrt{2}}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)(|0\rangle + |1\rangle) + \frac{\beta}{2\sqrt{2}}(|0\rangle - |1\rangle)(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) = \alpha|+-+\rangle + \beta|-+-\rangle.$$

Measuring the syndrome of the code protecting against phase-flips gives 10 *indicating that a phase-flip occurred on the second qubit. Correcting for it by applying a Z correction and inverting the encoding circuit finally yields $\alpha|0\rangle + \beta|1\rangle$, as expected.*

As before, we can understand the measurement of the syndrome as measuring some observables:

- one can detect a single bit flip in each subblock of size 3 as before, by measuring the Pauli operators $Z_1Z_2, Z_2Z_3, Z_4Z_5, Z_5Z_6, Z_7Z_8, Z_8Z_9$
- in order to detect phase errors, we proceed similarly by testing the parities of successive subblocks. Now the equivalent of Z_1 is X applied to the first logical qubit, which is simply $X_1X_2X_3$. In other words, one simply measures the operators $X_1X_2X_3X_4X_5X_6$ and $X_4X_5X_6X_7X_8X_9$. If they both give +1, then there wasn't a phase error, otherwise one can deduce where the phase-flip occurred as before.

We will now see that the 9-qubit Shor code can in fact correct arbitrary errors occurring on a single qubit.

1.4 Correcting arbitrary single-qubit errors

So far, we have seen how to protect against any of the Pauli errors occurring on a single qubit. But why would an error be a Pauli error?

Theorem 4. *If a quantum code protects against an arbitrary Pauli error, then it protects against an arbitrary error. The same statement holds for multiple-qubit errors.*

Proof. Let's establish this result for single-qubit errors. The very short answer is that the Pauli matrices form a basis of all possible 2×2 complex matrix. Indeed, let A be an arbitrary complex 2×2 matrix. There exist $a, b, c, d \in \mathbb{C}$ such that

$$A = a\mathbb{1} + bX + cY + dZ.$$

Forgetting about the unitarity of A for a moment, and assuming that A is applied to the j^{th} qubit of some n-qubit state $|\psi\rangle$ belonging to some quantum code, one obtains

$$A_j|\psi\rangle = a|\psi\rangle + bX_j|\psi\rangle + cY_j|\psi\rangle + dZ_j|\psi\rangle.$$

By assumption, the code can correct arbitrary Pauli errors on qubit j, which means that measuring the syndrome will result in

$$a|\psi\rangle|''\mathbb{1}''\text{ syndrome}\rangle + bX_j|\psi\rangle|''X''\text{ syndrome}\rangle + cY_j|\psi\rangle|''Y''\text{ syndrome}\rangle + dZ_j|\psi\rangle|''Z''\text{ syndrome}\rangle.$$

Applying the appropriate correction gives:

$$|\psi\rangle\left(a|''\mathbb{1}''\text{ syndrome}\rangle + b|''X''\text{ syndrome}\rangle + c|''Y''\text{ syndrome}\rangle + d|''Z''\text{ syndrome}\rangle\right),$$

which means that the state $|\psi\rangle$ is recovered.

Note that we assumed here that everything was occurring in superposition, but this is not necessary. If one simply measures the syndrome, it will collapse the state onto of of the four possibilities, and one can then decode as before. The crucial point is that the measurement of the syndrome projects the error (that belongs to some continuum of errors) onto one of 4 possible errors, which we know how to correct.

So far, we have only considered the effect of an operator A on the state. What we are really interested in is the effect of an *admissible operation* (or channel) on qubit j. Such an admissible operation can always be written as

$$\Phi_j(|\psi\rangle\langle\psi|) = \sum_{k=1}^{N} A_j^k|\psi\rangle\langle\psi|A_j^{k\dagger},$$

for some collection of matrices $\{A_j^1, \ldots, A_j^N\}$ satisfying

$$\sum_{k=1}^{N} A_j^{k\dagger}A_j^k = \mathbb{1}$$

and with a decomposition of the form

$$A_j^k = a_k\mathbb{1} + b_kX + c_kY + d_kZ.$$

We obtain

$$\Phi_j(|\psi\rangle\langle\psi|) = \sum_{a,a'} a\overline{a'}E_a|\psi\rangle\langle\psi|E_{a'}^{\dagger} \otimes |s_a\rangle\langle s_{a'}|$$

where a, a' are complex coefficients, $E_a, E_{a'}$ are Pauli errors and $s_a, s_{a'}$ are the corresponding syndromes. Measuring the syndrome register in a classical basis (i.e. measuring the syndrome) removes the non diagonal terms of the form $|s_a\rangle\langle s_{a'}|$ for $s_a \neq s_{a'}$. The resulting state has the form

$$\sum_a |a|^2 E_a|\psi\rangle\langle\psi|E_a^{\dagger} \otimes |s_a\rangle\langle s_a|.$$

This means that measuring the syndrome has projected the error onto one of the four Pauli errors. This state is in fact given by

$$\sum_{k=1}^{N} \left(|a_k|^2 |\psi\rangle\langle\psi| \otimes |s_{\mathbb{1}}\rangle\langle s_{\mathbb{1}}| + |b_k|^2 X|\psi\rangle\langle\psi|X \otimes |s_X\rangle\langle s_X| + |c_k|^2 Y|\psi\rangle\langle\psi|Y \otimes |s_Y\rangle\langle s_Y| \right.$$
$$\left. + |d_k|^2 Z|\psi\rangle\langle\psi|Z \otimes |s_Z\rangle\langle s_Z| \right).$$

And one can correct the Pauli errors by assumption, yielding

$$|\psi\rangle\langle\psi| \otimes \sum_{k=1}^{N} \left(|a_k|^2 |s_{\mathbb{1}}\rangle\langle s_{\mathbb{1}}| + |b_k|^2 |s_X\rangle\langle s_X| + |c_k|^2 |s_Y\rangle\langle s_Y| + |d_k|^2 |s_Z\rangle\langle s_Z| \right),$$

and discarding the syndrome gives the initial state $|\psi\rangle\langle\psi|$.

□

The proof can be generalized in a straightforward manner to deal with mutliple-qubit errors. Again, the main idea behind this fact is that measuring the syndrome projects an a priori arbitrary error onto a Pauli error.

The important consequence is that one can restrict the analysis of quantum error correcting codes to Pauli errors.

What we want to do when devising a QECC is to identify a subset $\mathcal{E}$ of Pauli operators

$$\mathcal{E} \subseteq \{E_a\} \equiv \{\mathbb{1}, X, Y, Z\}^{\otimes n}$$

that are the errors that we wish to be able to correct. Then, the idea is to perform a collective measurement (to measure the syndrome) and try to determine which E_a occurred and reverse it by applying $E_a^\dagger$.

A typical choice for the set $\mathcal{E}$ is the set of all Pauli errors of weight up to t. If we can correct any such error, then we say that the code can correct t errors. In particular, it is sufficient to correct X and Z errors of weight up to t.

2 Lecture 2

2.1 Stabilizer codes

To go further than the 9-qubit code of Shor, and develop a general formalism it is useful to take inspiration from classical coding theory. This formalism – *stabilizer codes* – was developed by Daniel Gottesman in his PhD thesis in the late 90s.

Let us recall first what the Pauli and Clifford groups are. The single-qubit Pauli group $\mathcal{P}_1$ is the group $\langle i\mathbb{1}, X, Z\rangle$ generated by the Pauli matrices. Its n-qubit generalisation is the n-fold tensor product of $\mathcal{P}_1$, that is $\mathcal{P}_n = \mathcal{P}_1^{\otimes n}$:

$$\mathcal{P}_n = \langle E_1 \otimes E_2 \otimes \ldots \otimes E_n \,:\, E_i \in \mathcal{P}_1 \rangle,$$

and has cardinality 4^{n+1}. An important property of Pauli operators is that any two of them either commute or anticommute.

Theorem 5. *Let $P, Q \in \mathcal{P}_n$. Then either $PQ = QP$ or $PQ = -QP$.*

Proof. For single-qubit matrices, we have

$$[X, X] = [Y, Y] = [Z, Z] = 0, \quad \{X, Y\} = \{X, Z\} = \{Y, Z\} = 0.$$

Let us write the Pauli operators as products of n single-qubit matrices: $P = P_1 \dots P_n$ and $Q = q_1 \dots Q_n$. Then P and Q commute if and only if they anticommute in an even number of positions. Otherwise, they anticommute. □

The *Clifford group* $\mathcal{C}_n$ is the automorphism group of the Pauli group:

$$\mathcal{C}_n = \{U \in \mathcal{U}(\mathcal{C}^{2^n}) \,:\, U\mathcal{P}_n U^\dagger = \mathcal{P}_n\}.$$

In words, any Pauli operator P is mapped to a Pauli operator via conjugation by a Clifford unitary.

Theorem 6. *The Clifford group $\mathcal{C}_n$ is generated by H, P and CNOT:*

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad P = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

A quantum code $\mathcal{Q}$ of parameters $[[n, k, d]]$ is a linear subspace of $(\mathbb{C}^2)^{\otimes n}$ of dimension 2^k, and *minimum distance* d. The stabilizer construction is very much inspired by the classical construction of linear codes, and there are two main ways to define a *stabilizer code*:

(i) via its *encoding circuit*: this is *Clifford unitary* $U \in \mathcal{U}(\mathbb{C}^{2^n})$ applied on $|\psi\rangle \otimes |0\rangle^{n-k}$ where $|\psi\rangle \in (\mathbb{C}^2)^{\otimes k}$ is a logical state of k qubits and $|0\rangle^{n-k}$ is an $(n-k)$-qubit ancilla. This gives

$$\mathcal{Q} = \{U(|\psi\rangle \otimes |0\rangle^{n-k}) \,:\, |\psi\rangle \in (\mathbb{C}^2)^{\otimes k}\}.$$

We denote by $|\overline{\psi}\rangle = U(|\psi\rangle \otimes |0\rangle^{n-k})$ the encoded version of $|\psi\rangle$.

(ii) via its *stabilizer*, i.e. a group $\mathcal{S} = \langle g_1, \dots, g_{n-k}\rangle$ generated by a set of $n-k$ Pauli operators that commute and that don't contain $-\mathbb{1}$. The code is then defined as the elements of the Hilbert space that are stabilized by $\mathcal{S}$:

$$\mathcal{Q} = \{|\psi\rangle \in (\mathbb{C}^2)^{\otimes n} \,:\, g_i|\psi\rangle = |\psi\rangle, \forall i \in [n-k]\}.$$

In other words, the code is defined as the $+1$ eigenspace of the generators. This space is well defined since the commutation condition ensures that the generators are all codiagonalizable. Moreover, since each generator is a Pauli operator, it has eigenvalues equal to ± 1.

In order to make reliable computations with a noisy quantum computer, the idea is to encode information with a quantum error correcting code and then perform the computation on the encoded state. We need a way to act on such states. This is done via logical operators.

Definition 7 (Normalizer)**.** *The normalizer of S in the Pauli group $\mathcal{P}_n$ is*

$$N(S) = \{g \in \mathcal{P}_n \,:\, gS = Sg\}.$$

Definition 8 (Logical operator)**.** *A logical operator of the stabilizer code with stabilizer $\mathcal{S}$ is a Pauli operator that leaves the code globally invariant, but that acts nontrivially on codewords. It is given by the set $N(\mathcal{S}) \setminus \mathcal{S}$, and corresponds to a Pauli operator that commutes with all the generators of $\mathcal{S}$, but that doesn't belong to $\mathcal{S}$.*

A logical operator will map a word in the quantum code to an orthogonal codeword.

The fact that the encoding circuit of a stabilizer code is a Clifford operation is particularly useful because it implies that logical Pauli errors correspond to Pauli physical errors.

Example 9 (Logical Z operator)**.** *Let us suppose that $|\overline{\psi}\rangle$ is the encoded version of $|\psi\rangle$, that is $|\overline{\psi}\rangle = U(|\psi\rangle|0\rangle^{n-k})$, where U is the Clifford encoding operation. The logical Pauli Z_1 operator, denoted $\overline{Z_1}$ maps the encoding of $|\psi\rangle$ to the encoding of $Z_1|\psi\rangle$, i.e. $|\overline{\psi}\rangle$ to*

$$U(Z_1|\psi\rangle|0\rangle^{n-k}) = UZ_1U^\dagger|\overline{\psi}\rangle,$$

which implies that $\overline{Z}_1 = UZ_1U^\dagger$, which is a Pauli matrix since U belongs to the Clifford group.

Translating from the encoding circuit to the stabilizer formalism: it is easy to infer the set of generators of the code if we know the encoding circuit. For this, one simply notes that the initial state before the encoding circuit is $|\psi\rangle|0\rangle^{n-k}$, where $|\psi\rangle$ is an arbitrary k-qubit state. All such states are stabilized by $n-k$ (independent) generators: $Z_{k+1}, \ldots, Z_n$ since the state $|0\rangle$ is defined as the $+1$ eigenstate of the Z Pauli operator. We obtain a description of the $n-k$ generators of the stabilizer group:

$$UZ_{k+1}U^\dagger, \ldots, UZ_nU^\dagger,$$

and it is straightforward to check that they commute pairwise:

$$(UZ_iU^\dagger)\cdot(UZ_jU^\dagger) = UZ_iZ_jU^\dagger = UZ_jZ_iU^\dagger = (UZ_jU^\dagger)\cdot(UZ_iU^\dagger).$$

Example 10. *Both the 3-qubit code and Shor's 9-qubit code are stabilizer codes. We have already seen their encoding circuit, which is a Clifford circuit. Moreover, we have seen that the stabilizer of the 3-qubit code is $\langle Z_1Z_2, Z_2Z_3\rangle$ and that the stabilizer of Shor's 9-qubit code is*

$$\langle Z_1Z_2, Z_2Z_3, Z_4Z_5, Z_5Z_6, Z_7Z_8, Z_8Z_9, X_1X_2X_3X_4X_5X_6, X_4X_5X_6X_7X_8X_9\rangle.$$

It is straightforward to check that these groups are Abelian. As expected, these stabilizers admit respectively $2 = 3-1$ (three physical qubits and one logical qubit) and $8 = 9-1$ (9 physical qubits for a single logical qubit) elements.

The Pauli operators Z_1 and $X_1X_2X_3$ are logical operators $\bar{Z}$ for the 3-qubit code and $\bar{Z}$ for Shor's 9-qubit code, respectively:

$$\begin{aligned} \textit{3-qubit code}: &\quad Z_1|\bar{1}\rangle = -|\bar{1}\rangle \\ \textit{Shor's code}: &\quad X_1X_2X_3|\bar{1}\rangle = -|\bar{1}\rangle. \end{aligned}$$

The stabilizer description is particularly useful to correct the errors. The idea is to measure the *syndrome*, that is to measure the eigenvalues of the stabilizer generators for the quantum state.

Definition 11 (Syndrome)**.** *The syndrome associated with error $E \in \mathcal{P}_n$ is the $(n-k)$-bitstring $\vec{s} = (s_1, \ldots, s_{n-k})$ defined by*

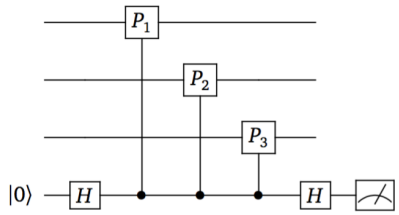
$$s_i = \begin{cases} 0 & \text{if} \quad [E, g_i] = 0, \\ 1 & \text{if} \quad \{E, g_i\} = 0. \end{cases}$$

If $s_i = 1$, meaning that the error anticommutes with a stabilizer, then $E|\psi\rangle$ is a -1 eigenvalue of g_i (if $|\psi\rangle$ is a valid codeword):

$$g_i E|\psi\rangle = -E g_i|\psi\rangle = -E|\psi\rangle.$$

In particular, the syndrome doesn't depend on the specific codeword, only on the Pauli error.

Note that it is easy to devise a quantum circuit to measure any Pauli operator (and in particular, any generator of the stabilizer group). The following picture for instance depicts a circuit to measure $P_1P_2P_3$ with Pauli operator P_i acting on qubit i:



The eigenvalues of any Pauli measurement P are 1 and -1, and the projector on the eigenspaces are $P_\pm = \frac{1}{2}(\mathbb{1} \pm P)$.

An alternative solution to measure the syndrome is to undo the encoding circuit, via $U^\dagger$, and measure the ancilla qubits in the computational basis, but this leaves the quantum state unprotected and might therefore not be the preferred solution in practice. In fact, in the context of quantum fault-tolerance, one must find a way to fight errors that occur during the syndrome measurement.

Definition 12 (Minimum distance)**.** *The* minimum distance *of a stabilizer code with stabilizer S is the minimum weight of a nontrivial logical operator:*

$$\begin{aligned} d_{\min}(\mathcal{Q}) &= \min\{|E| \,:\, [E, g_i] = 0 \forall i, E \notin \langle g_1, \ldots, g_{n-k} \rangle\} \\ &= \min\{|E| \,:\, E \in N(S) \setminus S\}. \end{aligned}$$

Example 13. *We have seen in the previous example that Z_1 and $X_1X_2X_3$ are logical errors for both the 3-qubit code and Shor's 9-qubit code, respectively. This implies that their respective minimum distances are upper bounded by 1 and 3. Since the 9-qubit code can correct any single-qubit error, its minimum distance is at least 3, which means that it is exactly 3.*

2.2 CSS codes

The CSS codes (for the name of their inventors, Calderbank, Shor, Steane) form an interesting subclass of all stabilizer codes where the generators of the stabilizer group are either products of Pauli-X or products of Pauli-Z. This is an appealing restriction because the commutativity condition between the generators now needs to be checked only between X-type and Z-type generators, since both X-type generators and Z-type generators obviously commute among themselves. In that case, both types of generators are described by binary words (with 1s at the coordinates corresponding to an X or Z-type operator).

A general way of defining both classes of generators is by choosing special types of classical codes. Let $\mathcal{C}$ be an $[n,k]$ classical linear code with a $n \times k$ generator matrix G and an $(n-k) \times n$ parity-check matrix H. This means

$$\mathcal{C} = \{Gy \,:\, y \in \mathbb{F}_2^k\} = \ker H.$$

Here we consider column vectors.

The *dual code* of $\mathcal{C}$, denoted $\mathcal{C}^\perp$, is defined as

$$\mathcal{C}^\perp = \{y \in \mathbb{Z}_2^n \,:\, x \cdot y = 0, \forall x \in \mathcal{C}\}.$$

This is an $[n, n-k]$ linear code. Moreover the generator and parity-check matrices of $\mathcal{C}$ and $\mathcal{C}^\perp$ are swapped (up to transposition):

$$G^\perp = H^T, \quad H^\perp = G^T.$$

We say that a code $\mathcal{C}$ is *weakly self-dual* if $\mathcal{C} \subseteq \mathcal{C}^\perp$ and *(strictly) self-dual* if $\mathcal{C} = \mathcal{C}^\perp$. A necessary and sufficient condition for $\mathcal{C}$ to be weakly self-dual is that $G^T G = 0$.

Definition 14. *A* CSS code $\mathrm{CSS}(\mathcal{C}_1, \mathcal{C}_2)$ *is defined from two classical linear codes* $\mathcal{C}_1$, $\mathcal{C}_2$ *of parameters* $[n,k_1]$ *and* $[n,k_2]$*, such that* $\mathcal{C}_2 \subseteq \mathcal{C}_1$*. The quantum code has parameters* $[[n, k_1 - k_2]]$ *and is spanned by the vectors*

$$|x_j + \mathcal{C}_2\rangle := \frac{1}{2^{k_2/2}} \sum_{y \in \mathcal{C}_2} |x_j + y\rangle,$$

where the elements of $\{x_j\}_{j=1\ldots 2^{k_1-k_2}}$ *belong to the quotient* $\mathcal{C}_1/\mathcal{C}_2$*. In other words, they satisfy* $x_i + x_j \notin \mathcal{C}_2$*, for any pair* $x_i \neq x_j$*.*

Lemma 15. *The code* $\mathrm{CSS}(\mathcal{C}_1, \mathcal{C}_2)$ *is a stabilizer code.*

The proof will be treated in exercise.

In particular, if $\mathcal{C}$ is weakly self-dual with parameters $[n,k]$, then $\mathrm{CSS}(\mathcal{C}^\perp, \mathcal{C})$ is a stabilizer code with parameters

$$[[n, n-2k]].$$

Codewords of the CSS code have the form $|x + \mathcal{C}_2\rangle$ where $x \in \mathcal{C}_1$ and two codewords $|x + \mathcal{C}_2\rangle$ and $|x' + \mathcal{C}_2\rangle$ differ if and only if x and x' belong to different cosets of $\mathcal{C}_2$ in $\mathcal{C}_1$.

An example of CSS code is Steane's 7-qubit code, where we take $\mathcal{C}_2 = \mathcal{C}_1^\perp$ and $\mathcal{C}_1$ to be the $[7,4]$ Hamming code with generator[2] and parity-check matrices

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

One can construct general Hamming codes by taking all possible words (except the all-zero word) for the columns of the parity-check matrix. These generalized codes have parameters $[2^m - 1, 2^m - m - 1, 3]$ and can tolerate one bit-flip since they all have distance 3. One can check that the dual of the Hamming code is weakly self-dual, since $(G^\perp)^T G^\perp = HH^T = 0$, and therefore $\mathrm{CSS}(\mathcal{C}_1, \mathcal{C}_1^\perp)$ is a valid CSS code encoding $4 - 3 = 1$ logical qubit.

Taking $x_0 = 0000000$ and $x_1 = 1111111$ as representatives of $\mathcal{C}_1/\mathcal{C}_1^\perp$, and enumerating the elements of $\mathcal{C}_1^\perp = \mathrm{Im}(H)$ as

$$\mathcal{C}_1^\perp = \{0000000, 0001111, 0110011, 0111100, 1010101, 1011010, 1100110, 1101001\},$$

we obtain the logical qubits as

$$\begin{aligned} |\bar{0}\rangle = \frac{1}{2\sqrt{2}}(&|0000000\rangle + |0001111\rangle + |0110011\rangle + |0111100\rangle + |1010101\rangle \\ &+ |1011010\rangle + |1100110\rangle + |1101001\rangle, \\ |\bar{1}\rangle = \frac{1}{2\sqrt{2}}(&|1111111\rangle + |1110000\rangle + |1001100\rangle + |1000011\rangle + |0101010\rangle \\ &+ |0100101\rangle + |0011001\rangle + |00101101\rangle. \end{aligned}$$

This illustrates one of the main strengths of the stabilizer formalism: in general, the logical qubits are given by very long expressions (a superposition over an exponential number of basis states), and the generators of the stabilizer yield a much more efficient description of the code.

The generators of the stabilizer can be chosen to be the rows of H_1 for X-type generators and the rows of $H_2^\perp = H_1$ for Z-type generators:

$$\begin{aligned} &X_4X_5X_6X_7, \\ &X_2X_3X_6X_7, \\ &X_1X_3X_5X_7, \\ &Z_4Z_5Z_6Z_7, \\ &Z_2Z_3Z_6Z_7, \\ &Z1Z_3Z_5Z_7. \end{aligned}$$

Lemma 16. *The minimum distance of a CSS code* $\mathrm{CSS}(\mathcal{C}_1, \mathcal{C}_2)$ *is* $\min(d(\mathcal{C}_1), d(\mathcal{C}_2^\perp))$.

[2]Here, we take the convention that the image of G is the right image, i.e., $\mathrm{Im}(G) = \{Gx : x \in \mathbb{Z}_2^4\}$.

This implies that Steane's 7-qubit code is a $[[7,1,3]]$ quantum code.
We prove the lemma by describing an explicit error correction strategy for a CSS code.

Error correction for a CSS code.

The general strategy is as usual: measure the syndrome and apply a correction. For CSS codes, the syndrome is naturally divided into two parts: X-type errors are corrected with the syndrome of the Z-type generators, and Z-type errors are corrected with the syndrome of the X-type generators. This suggests a two-part procedure:

- **X-type errors:** (i) compute the syndrome for code $\mathcal{C}_1$ (i.e. for generators of the form Z^f for f a row of H_1): this is the reversible operation

  $$|y\rangle|0\dots0\rangle \mapsto |y\rangle|s_1(y)\rangle,$$

  with $s_1(y) = H_1 y$, the syndrome of y with respect to code $\mathcal{C}_1$; (ii) measure the syndrome, and (iii) correct for bit-flips by applying Pauli-X corrections;
- **swapping between codes:** apply a Hadamard transform to every qubit;
- **Z-type errors:** same procedure as before but for the code $\mathcal{C}_2^\perp$, i.e. generators of the form X^e for e a row of $H_2^\perp$;
- **returning to the initial code:** apply again a Hadamard transform to every qubit.

Let us verify that this decoding procedure correctly recovers the codeword provided that the weight of the error is less than t, where t is the maximum number such that both $\mathcal{C}_1$ and $\mathcal{C}_2^T$ can tolerate t errors[3].

Consider a Pauli error $X^v Z^w$ with bit strings $v, w \in \{0,1\}^n$ applied to a codeword $\sum_j \alpha_j |x_j + \mathcal{C}_2\rangle$. The state becomes

$$X^v Z^w \sum_j \alpha_j |x_j + \mathcal{C}_2\rangle = \sum_j \alpha_j (-1)^{v\cdot w} Z^w |x_j + v + \mathcal{C}_2\rangle$$

where we used that $X^v Z^w = (-1)^{v\cdot w} Z^w X^v$.

The first step of the correction procedure corrects X-type errors by computing the syndrome relative to $\mathcal{C}_1$. This will yield the syndrome of v for $\mathcal{C}_1$, and provided that $|v| \le t$, the procedure will return v and apply the Pauli correction X^v, giving,

$$\sum_j \alpha_j (-1)^{v\cdot w} X^v Z^w |x_j + v + \mathcal{C}_2\rangle = Z^w \sum_j \alpha_j |x_j + \mathcal{C}_2\rangle.$$

[3]The parameter t is related to the minimum distance via $d = 2t + 1$.

After the Hadamard transformation, the state becomes

$$\begin{aligned} H^{\otimes n} Z^w \sum_j \alpha_j |x_j + \mathcal{C}_2\rangle &= X^w H^{\otimes n} \sum_j \alpha_j |x_j + \mathcal{C}_2\rangle && \text{(since } H^{\otimes n} Z^w = X^w H^{\otimes n}\text{)} \\ &= \sum_j \alpha_j X^w H^{\otimes n} X^{x_j} |\mathcal{C}_2\rangle \\ &= \sum_j \alpha_j X^w H^{\otimes n} X^{x_j} H^{\otimes n} |\mathcal{C}_2^\perp\rangle && \text{(proven in exercise)} \\ &= \sum_j \alpha_j X^w Z^{x_j} |\mathcal{C}_2^\perp\rangle. \end{aligned}$$

Using the same argument as before, one can compute the syndrome relative to $\mathcal{C}_2^\perp$ and correct the X-type error. This will work correctly provided that $|w| \leq t$, where t is a lower bound on the correction capacity of $\mathcal{C}_2^\perp$. Undoing the Hadamard transform then returns the original codeword.

2.3 An example: the toric code

At the end of the 90s, Alexei Kitaev showed that cellullations of surfaces (and of higher-dimensional manifolds) gave a very general method to derive CSS codes, with parameters depending on the properties of the surface. The most famous example is the *toric code*, which can be realized by taking a square cellullation of a torus.

Consider an $N \times N$ square grid on a torus, and put a qubit on each of the $2N^2$ edges. We define a CSS code by choosing the following generators of weight 4:

- *plaquette* operators: for each plaquette p on the grid, define $g_p^X := \bigotimes_{e \in \partial p} X_e$, where $e \in \partial p$ means that edge e belongs to the boundary of plaquette p,
- *vertex* operators: for each vertex v in the grid, define $g_v^Z := \bigotimes_{e \sim v} Z_e$, where $i \sim v$ means that edge e is incident to vertex v.

Let us immediately verify that these generators commute: for this, it is enough to notice that a vertex and a plaquette operator either do not overlap, or else overlap in exactly 2 positions.

There are N^2 vertices on the grid and N^2 plaquettes, so we have defined $2N^2$ generators. Note, however, that these generators are not independent since the product of all vertex operators is the identity, and the product of all plaquette operators is also the identity:

$$\bigotimes_p g_p^X = \mathbb{1}, \quad \bigotimes_v g_v^Z = \mathbb{1}.$$

There are the only nontrivial relations, meaning that there are $2N^2 - 2$ independent generators for $2N^2$ qubits, which yields 2 logical qubits.

From our earlier definition of CSS codes, we need two classical codes $\mathcal{C}_1$ and $\mathcal{C}_2$, with $\mathcal{C}_2 \subseteq \mathcal{C}_1$:

- the code $\mathcal{C}_1$ is the *cycle code* of the grid: the support of codewords corresponds to a cycle, i.e., its boundary is null;
- the code $\mathcal{C}_2$ is generated by words whose support is the boundary of a set of plaquettes.

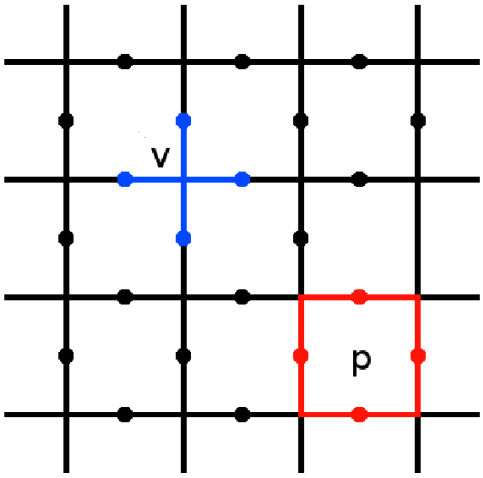


Figure 1: Local structure of the toric code: qubits are placed on edges, vertex operators are the product of X operators applied to the 4 neighboring qubits of a vertex, plaquette operators are the product of Z operators applied to the 4 qubits on the boundary of a plaquette (By James Wootton, `https://commons.wikimedia.org/w/index.php?curid=11823316`)

The inclusion $\mathcal{C}_2 \subset \mathcal{C}_1$ follows from the fact that the boundary of a boundary is always null:

$$\partial\partial = 0.$$

This relation is at the heart of all topological/homological quantum error correcting code constructions.

In order to describe the logical qubits, we need to understand the equivalence classes of $\mathcal{C}_1/\mathcal{C}_2$, that is, the cycles that are not a boundary. There are indeed two inequivalent families of such cycles, corresponding to a loop around the torus. These cycles are *homologically nontrivial* meaning that they cannot be deformed (by addition of boundary) to yield the null cycle. For this reason, the toric code is an example of *topological code*: properties of the quantum code result from the topology of the underlying manifold.

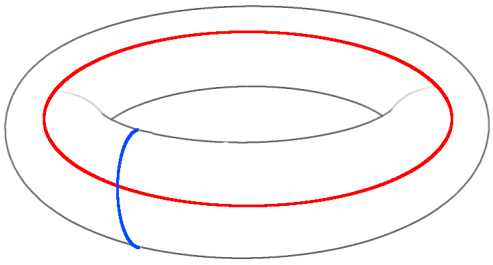


Figure 2: Local structure of the toric code: qubits are placed on edges, vertex operators are the product of X operators applied to the 4 neighboring qubits of a vertex, plaquette operators are the product of Z operators applied to the 4 qubits on the boundary of a plaquette (By James Wootton, `https://commons.wikimedia.org/w/index.php?curid=11823316`)

In particular, since the minimum size of a nontrivial cycle is N, we deduce that the minimum

distance of the code is also N, and the parameters of the toric code read:

$$[[2N^2, 2, N]].$$

Another particularly interesting feature of the toric code is that it is an example of *low-density parity-check (LDPC)* code, meaning that each generator only involves a constant number of qubits (4 for the toric code) and that each qubit is only involved in a constant number of generators (4 again for the toric code). This LDPC condition is particularly important when it comes to experimental implementation since it is much easier to engineer *local interactions* involving a constant number of qubits. In addition, the qubits that interact via the generators for the toric code are spatially local: qubits only interact with their neighbors. In fact, the leading approaches to build a quantum computer are based on the toric code (or its cousin the surface code).

Despite an intensive study of quantum LDPC codes, it turned out to be extremely difficult to find better LDPC codes than the toric codes. For about 20 years, the best bound for the minimum distance was $n^{1/2} \log^{1/4} n$. In 2020, a series of papers showed that $d_{\min} = \Theta(n/\log n)$ is achievable!

Decoding the toric code.

Let us first consider X-type errors. Their associated syndrome is obtained via the Z-type generators, corresponding to vertices. Let X^E be an X-type error with support on the set E. Then the syndrome of X_E is given by the set of vertices in the boundary of E:

$$s(X^E) = \partial E.$$

In order to decode, one must therefore find an error with the appropriate syndrome. A particularly popular decoder is the algorithm *minimum weight perfect matching*. This is not optimal, however, since the most probable error isn't necessarily the error of minimum weight, but rather the error whose equivalence class is the most probable. However, the minimum weight perfect matching algorithm is efficient.

In order to address Z-type errors, it is convenient to exploit *Poincaré duality*: the dual of the cellulation looks exactly like the initial cellulation, but with the roles of vertices and plaquettes exchanged. In other words, one can correct Z-type errors with the same approach as explained for the X-type errors, but working in the dual cellulation.

The threshold of the toric code is about $10 - 11\%$ for the depolarizing channel, corresponding to an error model where X, Y, Z errors occur independently with probability $p/3$ and not error occurs with probability $1 - p$.

Beyond the toric code.

This approach isn't limited to tessellations of the torus, but can be generalized in a straightforward way to cellullations of arbitrary closed manifolds in arbitrary dimensions. Even dimensions are convenient to exploit the Poincaré duality, and this is why the 4-dimensional toric code is also quite popular. One can also change the geometry and work with hyperbolic geometry rather than Euclidean space.