Quantum error correction

Online material:

– John Preskill's course at Caltech

– Dan Browne's course at UCL on "Topological Codes and Computation"

– Daniel Gottesman's PhD thesis introducing stabilizer codes

Outline:

– lecture 1: How to protect a single qubit from noise: Shor's code

– lecture 2: the stabilizer formalism

– lecture 3: the toric code

– lecture 4: quantum LDPC codes with good performance

I) How to protect a single qubit from noise: Shor's code

a) 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 all the relevant quantum information is gone after a couple hundred gates. Running interesting algorithms about quantum chemistry or Shor's factoring algorithm require much more gates than that, say $10^{12} - 10^{15}$ gates. 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 project 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*: 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 10^7 and 10^8 physical (good enough) qubits. This is in part because qubits and quantum gates are much more noisy than classical bits and gates, and also because we still haven't discovered the optimal solution to quantum error correction and fault tolerance.

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: for instance, sampling a string from the output distribution of a random quantum circuit.

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 requirements are 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 destroys the whole superposition. For instance, if you start with a cat state

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

  and lose the first qubit (*i.e.,* corresponding mathematically to a partial trace), then the remaining state is

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

  which corresponds to a classical mixture of Alive and Dead.

- applying noisy gates just increases the total noise: errors pile up. Even worse, any error correction mechanism that we wish to implement will also be prone to errors. The question then is whether we can correct the errors more quickly than they appears. 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 $|\psi\rangle \to |\psi\rangle|\psi\rangle$ for an arbitrary state $|\psi\rangle$ 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.

- assume that you manage to perform a quantum computation in a fault-tolerant way. At the very last step, you still need to measure the final outcome and this measurement will be a little bit noisy, hence making the result incorrect.

b) Classical error correction and fault-tolerance

Before discussing quantum error correction, let's quickly 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.

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

[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.

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[2]).
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.

c) A first example: Shor's code

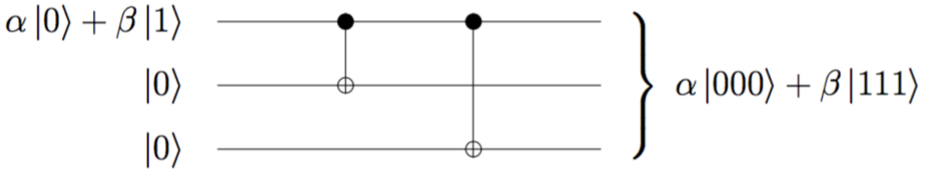
We want to generalize the 3-bit repetition code to the quantum setting. The goal is as follows: Alice has a qubit in a pure state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and wants to send it to Bob (or to keep it in a noisy memory for some time). Here, we suppose that Alice doesn't know α and 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:

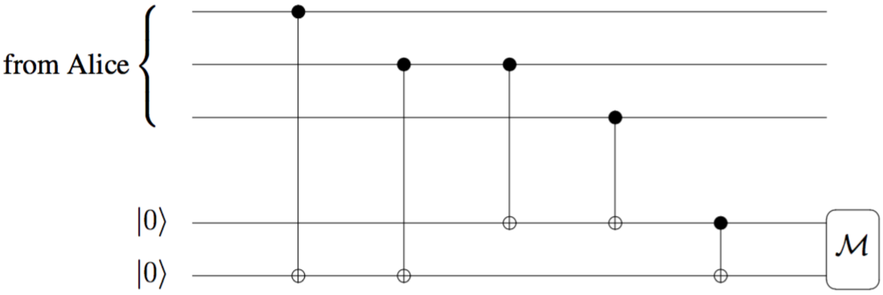

$\alpha|0\rangle + \beta|1\rangle$
$|0\rangle$
$|0\rangle$
$\alpha|000\rangle + \beta|111\rangle$

[2] The idea that something simple to implement can be done error-free is actually crucial. In the quantum computing case, we will assume that *classical operations* can be done without any error, and this is indeed a very reasonable hypothesis. In particular, it will be sufficient to encode the final result with a classical error correcting code to make sure that the final quantum step doesn't ruin the whole computation.

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:

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

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:

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

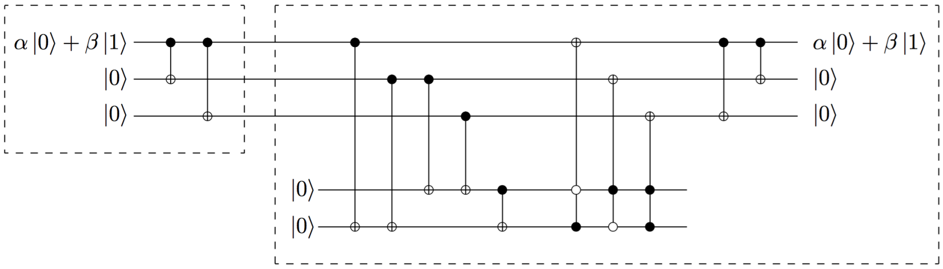
Finally, he can apply the inverse of the encoding circuit to recover the initial qubit

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

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:


$\alpha|0\rangle + \beta|1\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$\alpha|0\rangle + \beta|1\rangle$
$|0\rangle$
$|0\rangle$

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.

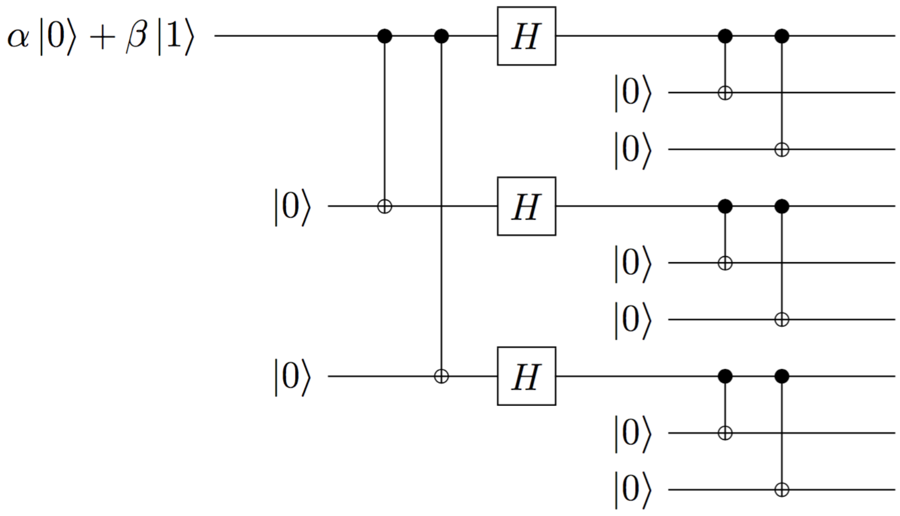
Is there a way to protect against both bit-flip and phase-flip errors at the same time? Not with a 3-qubit code. But one can with a simple 9-qubit code: Shor's code. The idea relies on *concatenation*: one first applies the code that protects against phase-flips then one encodes the resulting logical 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 qubit is spread over 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 (forgetting about normalization)*

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

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

$$\alpha(|000\rangle+|111\rangle)(|000\rangle-|111\rangle)(|000\rangle+|111\rangle)+\beta(|000\rangle-|111\rangle)(|000\rangle+|111\rangle)(|000\rangle-|111\rangle)$$

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

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

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.

d) 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

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

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

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

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.