CMSC 657: Introduction to Quantum Information Processing
Lecture 21

Instructor: Daniel Gottesman

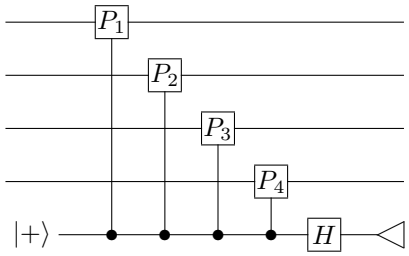
Fall 2024

1 Fault Tolerance

1.1 Fault-Tolerant Quantum Error Correction

One important component of a fault-tolerant protocol is the ability to perform error correction despite errors occurring while we are doing so. There are a variety of different types of fault-tolerant error-correction gadgets, but the simplest conceptually is Shor error correction. In Shor error correction, we measure the bits of the error syndrome one-by-one; that is, we measure the eigenvalue of each generator of the stabilizer.

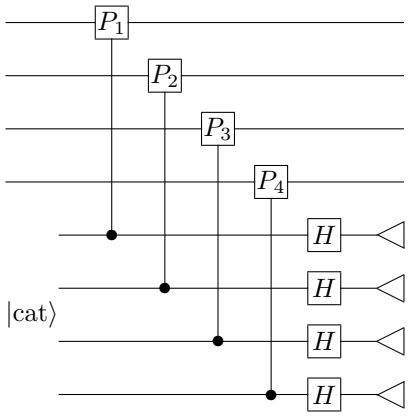
Let us start out by considering how to measure a stabilizer generator's eigenvalue non-fault-tolerantly. Here is a general procedure:



If we want to measure a generator $M = \bigotimes P_i$, start with an ancilla qubit in the $|+\rangle$ state, and then perform a controlled-P_i gate from the ancilla to the ith qubit for each i. When the ancilla is $|0\rangle$, the controls mean nothing happens. When the ancilla is $|1\rangle$, the controls are all active and we get a tensor product of the P_i's. That is, we perform M on the data. If we are in an eigenstate, we get a phase equal to the eigenvalue. If we are in a superposition of eigenstates, each eigenstate in the superposition gets a phase equal to its eigenvalue. If the eigenvalue is $+1$, therefore, the ancilla ends up in state $|+\rangle$, and if the eigenvalue is -1, the ancilla ends up in state $|-\rangle$. The Hadamard and measurement then gets us the relevant bit of the error syndrome.

However, this procedure is non-fault-tolerant because a single error in the ancilla could propagate into many different data qubits. Moreover, a single error can give us the wrong value for the syndrome bit, causing us to incorrectly deduce the error.

To make this fault-tolerant, we replace the $|+\rangle$ state with a "cat state" $|00\ldots0\rangle + |11\ldots1\rangle$. Instead of controlling all Paulis from one qubit, we control the ith Pauli P_i by the ith qubit of the cat state. This means that if one qubit of the ancilla goes wrong, only one data qubit is affected. When everything is OK, all the cat state qubits are 0 together, in which case we do nothing to the data block, or 1 together, in which case we perform M. The ancilla ends up as $|00\ldots0\rangle \pm |11\ldots1\rangle$, with sign equal to the eigenvalue of M (in the absence of errors). It is still possible that a single error will give us the wrong eigenvalue, but this can be dealt with by measuring the error syndrome multiple times.



(Doing the transversal Hadamard and then measuring gives us a random bit string of even parity if the state is $|00\ldots0\rangle + |11\ldots1\rangle$ and of odd parity if the state is $|00\ldots0\rangle - |11\ldots1\rangle$.)

Another point I am glossing over is how to create the cat state. If there is a bit flip error on a single qubit of the cat state, it can propagate into an error on a single qubit of the code, but if there are two or more bit flip errors on the cat state, which can happen through a non-fault-tolerant cat state creation procedure, they will all propagate into the code block, which is what we are trying to avoid. The solution is to verify the cat state after creating it by doing CNOTs from pairs of qubits to ancilla qubits to see if they are the same (as desired) or different (which indicates a potential problem).

1.2 Threshold Theorem

The threshold theorem is perhaps the central result of the theory of fault tolerance. It tells us that it is possible, in principle, to build a large quantum computer, even in the presence of error, provided we can get the error rates in our experimental system below some fixed value, the threshold error rate.

Theorem 1 (Threshold theorem)**.** *There exists a threshold error rate p_T such that if the error rate p per gate or time step is below p_T, then it is possible to do arbitrarily long quantum computations reliably. In particular, to do a computation of length T with overall logical error rate ϵ in the final answer, we need* polylog(T/ϵ) *physical qubits per logical qubit and a similar overhead in the number of gates.*

Polylog (T/ϵ) here indicates a polynomial in the logarithm of T/ϵ.

To prove a theorem about fault tolerance, we need to rigorously model the system. In particular, we usually model the errors in a simplified way as follows: Each gate or other circuit element (state preparations and measurements) has some probability p of error. We also allow a possibility of error on qubits that are sitting around and waiting. If there is an error, the qubit(s) involved in the circuit element undergo the circuit element plus some error, which may be specified in the model, but sometimes is just taken to be "anything could happen on those qubits." Importantly, we assume in this case that no other qubits get errors directly as a result of this event, although of course errors could potentially propagate from these qubits to others or there could be separate error events that cause errors on other qubits.

This is just a simple model for getting the basic results, but of course you can extend it or complicate it in various ways. For instance, you can have different error rates for different circuit elements (e.g., two-qubit gates might have a higher error rate than one-qubit gates, as is true in most implementations). You can have a single error event that causes errors in multiple additional qubits not involved in the gate. You can have non-Markovian errors where the quantum computer interacts weakly with an environment with a persistent memory. In most though not all of these cases, we still have a threshold theorem. The only situation in terms of weak errors where we know the threshold theorem fails is when single error events can cause correlated errors on many qubits, a constant fraction of the whole system.

There are a number of other assumptions that go into the threshold theorem. Some of them are inessential — for instance, assuming we can do gates between pairs of qubits arbitrarily far apart and assuming we can

make measurements in the middle of the computation to learn the error syndrome. Others are more-or-less essential. In particular, we need to be able to perform operations in parallel on many qubits at once so that we can error correct all the qubits in the computer. Otherwise, some qubits will have to wait around a long time without being corrected and during that time, are likely to accumulate too many errors for our code to correct. We also need a way of preparing new ancillas during the computation or resetting existing ones to use for measuring the error syndrome. Otherwise, out syndrome measurements will reflect a lot of accumulated noise and not tell us much about the actual error in the system.

Proof. The threshold theorem is usually proved via concatenated codes: Encode one qubit as 7 qubits, using a fault-tolerant protocol. This reduces the logical error rate per gate from p to Cp^2. Encode again, to reduce the logical error rate to $C(Cp^2)^2$. Let $p_T = 1/C$. After ℓ levels of concatenation, we have a logical error rate

$$p = p_T(p/p_T)^{2^\ell}. \tag{1}$$

Thus, if we want the overall error rate for the whole computation to be ϵ, we need a logical error rate of ϵ/T per gate, which means that we need $p/p_T < 1$ (i.e., $p < p_T$, the error rate is below the threshold), and ℓ to be $O(\log\log(T/\epsilon)$. That is not many levels of concatenation.

Each level multiples the number of qubits needed by a constant D, which includes the overhead from the code itself, but also all the ancillas needed for error correction, magic state distillation, etc. The total overhead (ratio of physical qubits to logical qubits) is therefore D^ℓ. An exponential growth in the overhead sounds bad, but remember the logical error rate is decreasing as a double exponential. Consequently, we need overhead $D^{O(\log\log(T/\epsilon))} = O(\text{polylog}(T/\epsilon))$. □

Obviously, the actual numerical size of the threshold is an important value, because it sets a target for the error rate that experimentalists need to achieve. Unfortunately, the threshold is not a single number. For one thing, it depends on the set of assumptions we are making about the computer and on the details of the error model. When different gates can have different error rates and the error models involve multiple parameters, the threshold is actually a higher-dimensional surface separating correctable error models from uncorrectable ones.

But with those caveats, the actual numbers remain important, and so there's been a lot of attention to them. The best we know how to do, using ridiculously large constant factors in that polylog, is a threshold about 5% for a depolarizing noise model (X, Y, Z equally likely when there is an error).

A more realistic protocol uses the surface code. In this case, we don't use concatenated codes, since the surface code has a threshold error rate as we increase the size of the surface. When we have an $L \times L$ grid on the torus (so $2L^2$) and an error rate of p, we expect about $2L^2p$ errors. For large L, this is much greater than the distance L. However, while the worst-case error of size L wraps around the torus and causes a logical error, a set of randomly-located errors are unlikely to do so for low p. At high p, typical errors connect up and include curves that wrap around the torus, but at low p, errors form small isolated islands and can be corrected. The transition between these two regimes is a *percolation* transition, and the point where it occurs is a threhold for error correction. The threshold for fault tolerance is lower since we have the possiblity of multiple errors before we can complete error correction, but again with depolarizing noise, it turns out the threshold is a bit below 1%. That is a pretty reasonable error rate and moreover, the surface code works well with a two-dimensional arrangement of qubits (although on a flat surface, not a torus). The overhead depends on how big a computation we want to do and on the physical error rate, but is likely to be hundreds or thousands of physical qubits per logical qubit. That is a bit high, but not completely unworkable. Consquently, the surface code has gotten a lot of attention and is currently the leading candidate for building large fault-tolerant quantum computers. However, high-rate low-density parity check codes have been drawing attention recently as well. They can tolerate comparable error rates with substantially lower overhead, but cannot be arranged in two dimensions like a surface code.

2 Distances and Entropy in Quantum Systems

2.1 Fidelity of Quantum States

How do we measure whether two quantum states are close to each other? This is something we addressed before, but now it is time to go into more detail. Recall that there is not a unique answer to this question.

For pure states, the most natural measure is the inner product between two states. This can be complex, so we actually use the *fidelity*, the absolute value of the inner product:

$$F(|\phi\rangle, |\psi\rangle) = |\langle\phi|\psi\rangle|. \tag{2}$$

The fidelity of two pure states has an *operational* meaning, the probability of getting the states confused. In particular, if we measure $\{|\psi\rangle\langle\psi|, I - |\psi\rangle\langle\psi|\}$, for $|\psi\rangle$, we always get the first outcome, but for $|\phi\rangle$, we might still get the first outcome with probability $F(|\phi\rangle, |\psi\rangle)^2$.

It is easy to extend this operational definition of the fidelity to apply to one pure and one mixed state:

$$F(|\psi\rangle, \sigma)^2 = \mathrm{Tr}(|\psi\rangle\langle\psi|\sigma). \tag{3}$$

For two mixed states, the definition of fidelity is more tricky:

$$F(\rho, \sigma) = \mathrm{Tr}\,\sqrt{\rho^{1/2}\sigma\rho^{1/2}}. \tag{4}$$

Since ρ is a density matrix, it is positive semi-definite and has a unique positive semi-definite square root $\sqrt{\rho}$ such that $(\sqrt{\rho})^2 = \rho$. You can find it by diagonalizing ρ and taking the square roots of the diagonal elements.

It is not at all obvious that this is the correct definition for fidelity between density matrices, but it is. Note that if $\rho = |\psi\rangle\langle\psi|$ is a pure state, then $\sqrt{\rho} = \rho$, so

$$\begin{aligned} F(\rho, \sigma) =& \mathrm{Tr}\,\sqrt{|\psi\rangle\langle\psi|\sigma|\psi\rangle\langle\psi|} && (5)\\ =& \sqrt{\langle\psi|\sigma|\psi\rangle}\,\mathrm{Tr}\,\sqrt{|\psi\rangle\langle\psi|} && (6)\\ =& \sqrt{\mathrm{Tr}(|\psi\rangle\langle\psi|\sigma)}, && (7) \end{aligned}$$

as desired.