Quantum Computing (CST Part II)

Lecture 14: Fault Tolerant Quantum Computing

A quantum computer is a machine that performs quantum error correction; quantum computation is merely a side effect.
Andrew Steane

Why we need fault tolerance

Classical computers perform complicated operations where bits are repeatedly "combined" in computations, therefore if an error occurs, it could in principle propagate to a huge number of other bits. Fortunately, in modern digital computers errors are so phenomenally unlikely that we can forget about this possibility for all practical purposes.

Errors do, however, occur in telecommunications systems, but as the purpose of these is the simple transmittal of some information, it suffices to perform error correction on the final received data.

In a sense, quantum computing is the worst of both of these worlds: errors do occur with significant frequency, and if uncorrected they will propagate, rendering the computation useless. Thus the solution is that we must correct errors as we go along.

Fault tolerant quantum computing set-up

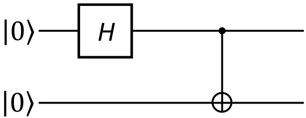
For fault tolerant quantum computing:

- We use encoded qubits, rather than physical qubits. For example we may use the 7-qubit Steane code to represent each *logical* qubit in the computation.
- We assume that when a gate is performed on a physical qubit it is corrupted by a depolarising channel. As we will periodically perform error correction, for the reasons given in the previous lecture, this is a physically reasonable model for a more general class of errors.
- We use *fault tolerant quantum gates*, which are defined such that a single error in the fault tolerant gate propagates to at most one error in each encoded block of qubits.

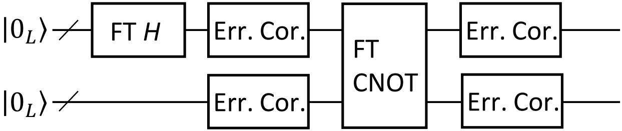
By a “block of qubits”, we mean (for example) each block of 7 physical qubits that represents a logical qubit using the Steane code.

Example of a fault tolerant circuit

Consider the quantum circuit for preparing the state $|\Phi^+\rangle$:



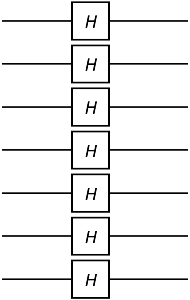
The fault tolerant version is:



Note that errors can occur even if a gate is not performed, therefore we must perform error correction on all qubits at regular intervals (even if they are idling).

The fault tolerant Steane code Hadamard gate

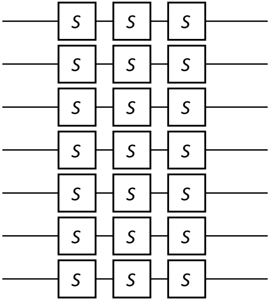
We can perform the Steane code Hadamard gate transversally, by applying a Hadamard gate to each of the seven qubits:



The encoded Hadamard gate is clearly fault tolerant as each component physical Hadamard gate acts only on a single qubit.

The fault tolerant Steane code S gate

We can also perform the Steane code S gate transversally, by applying three S gates to each of the seven qubits:



Again, this is fault tolerant as each component physical S gate acts on a single qubit.

The fault tolerant Steane code S gate (cont.)

To see that the Steane code S gate works, recall that the Steane code encodes the computational basis states as follows:

$$|0_L\rangle = \frac{1}{\sqrt{8}}\big(|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle$$
$$|0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle \big)$$
$$|1_L\rangle = \frac{1}{\sqrt{8}}\big(|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle$$
$$|1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle \big)$$

The action of the three S gates on each qubit is thus:

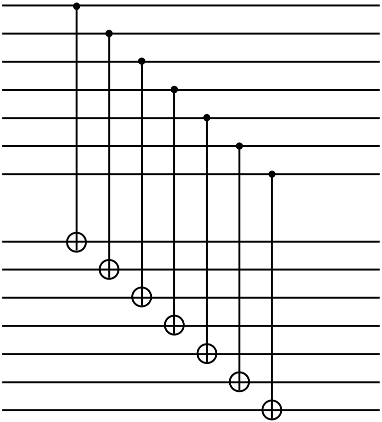
$$\begin{aligned} S_L|0_L\rangle =& \frac{1}{\sqrt{8}}\big(|0000000\rangle + (i^3)^4|1010101\rangle + (i^3)^4|0110011\rangle + (i^3)^4|1100110\rangle \\ &+ (i^3)^4|0001111\rangle + (i^3)^4|1011010\rangle + (i^3)^4|0111100\rangle + (i^3)^4|1101001\rangle \big) \\ =& |0_L\rangle \end{aligned}$$

because $i^{12} = 1$. Similarly for $|1_L\rangle$ (because $i^{21} = i^9 = i$):

$$\begin{aligned} S_L|1_L\rangle =& \frac{1}{\sqrt{8}}\big((i^3)^7|1111111\rangle + (i^3)^3|0101010\rangle + (i^3)^3|1001100\rangle + (i^3)^3|0011001\rangle \\ &+ (i^3)^3|1110000\rangle + (i^3)^3|0100101\rangle + (i^3)^3|1000011\rangle + (i^3)^3|0010110\rangle \big) \\ =& i|1_L\rangle \end{aligned}$$

The fault tolerant Steane code CNOT gate

The Steane code CNOT can also be performed transversally:



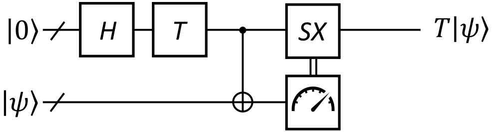
This is fault tolerant, as each of the component CNOT gates acts on exactly one qubit in the first block and one qubit in the second block. Therefore a single error will propagate to (at most) one qubit in each block, as per the definition of fault tolerant quantum gates.

Does transversality suffice?

- We have seen that, when qubits are encoded using the Steane code, we can perform H, S and CNOT gates transversally (and therefore fault tolerantly). Together, these gates generate the *Clifford group*, and the Gottesman-Knill theorem (that we met in lecture 5) tells us that Clifford group circuits can be efficiently simulated on a classical computer.
- We must add the T gate to have a universal gate-set. Unfortunately the T gate cannot be performed transversally.
- The fact that the very gate which yields the (conjectured) quantum advantage is the one which cannot be performed transversally is not a coincidence and is a general feature of this approach to error correction (a more detailed study of quantum error correction in the stabiliser formalism reveals why this is so).
- Therefore we take a different approach to the construction of fault tolerant T gates – this yields a solution which is efficient in an asymptotic sense, but is more resource intensive in practise. Therefore T gate count reduction is an important factor in quantum circuit compilation.

The fault tolerant Steane code T gate

To perform the T gate fault tolerantly, we must prepare an encoded ancilla state, which we use to perform the following circuit:



Here the H, T, CNOT and classically controlled SX gates are all performed "transversally", and together perform an encoded T gate.

Elements of fault tolerance

We have seen that we can construct a fault tolerant universal gate-set using the Steane code. For a fully fault tolerant circuit, we must also construct:

- Fault tolerant state preparation
- Fault tolerant error correction
- Fault tolerant measurement

All of these can be achieved, but we won't look at the details of exactly how in this course (see the referenced pages in Nielsen and Chuang if you are interested).

Fault tolerant quantum computing

Three things make fault tolerant quantum computing possible, the first two of which we have already seen:

1. Fault tolerant quantum gates, in which a single error propagates to at most one qubit in each encoded block of qubits.
2. The existence of quantum error correcting codes, which guarantee to correct a single error if it occurs.
3. The ability to concatenate error correcting codes.

Concatenated codes

We have already seen that the Steane code suppresses the error in the depolarising channel from p_e to $\mathcal{O}(p_e^2)$ – let this be (for some constant c):

$$p_e' = cp_e^2$$

We can now concatenate the Steane code by encoding a single *logical* qubit not with seven physical qubits, but rather with seven qubits that are themselves encoded using the Steane code. Therefore we have used $7^2 = 49$ physical qubits in a two-level concatenated Steane code to form one logical qubit. In this case, the error has been suppressed to:

$$c \times (cp_e^2)^2 = \frac{(cp_e)^{2^2}}{c}$$

It follows that if we concatenate k times, we suppress the error to:

$$\frac{(cp_e)^{2^k}}{c}$$

Therefore using concatenation we can suppress the error in each encoded, logical qubit as much as we would like. Moreover, we will see that when the physical qubit error p_e is sufficiently small this error suppression is efficient.

The fault tolerance threshold of a code

If a quantum computer has n qubits then any polynomial time quantum algorithm will have $\mathsf{p}(n)$ quantum gates, for some polynomial $\mathsf{p}(\cdot)$. Let p'_e be the probability of the failure of a single fault tolerant gate, when encoded in a k level concatenated code. We can thus bound the probability of the computation failing, p_f, using the Union bound:

$$\begin{aligned} p_f &\le \mathsf{p}(n) p'_e \\ &= \mathsf{p}(n) \frac{(cp_e)^{2^k}}{c} \end{aligned}$$

Therefore, in order to achieve a desired maximum error of ϵ, it suffices to choose k such that:

$$\frac{(cp_e)^{2^k}}{c} \le \frac{\epsilon}{\mathsf{p}(n)}$$

Such a k exists if the physical qubit error rate, p_e, is such that $p_e < p_{th} = \frac{1}{c}$. For any error correcting code, p_{th} is known as the code's threshold.

How many operations are required for fault tolerance?

We must now ask how much additional resource is required to perform the fault tolerant computation. To do so, notice that a k level concatenated code requires d^k operations, where d is some constant. We therefore require $d^k \mathsf{p}(n)$ operations to perform the fault tolerant computation, compared to $\mathsf{p}(n)$ in the original circuit. To evaluate d^k, consider an encoding that satisfies the threshold condition with equality, from the equation on the previous slide we can re-arrange and take logarithms of both sides to get:

$$
\begin{aligned}
2^k \log(p_e c) &= \log(c\epsilon/\mathsf{p}(n)) \\
\implies -2^k \log(p_e c) &= -\log(c\epsilon/\mathsf{p}(n)) \\
\implies 2^k \log(1/(p_e c)) &= \log(\mathsf{p}(n)/(c\epsilon)) \\
\implies 2^k &= \frac{\log(\mathsf{p}(n)/(c\epsilon))}{\log(1/(p_e c))} \\
\implies d^k &= \left(\frac{\log(\mathsf{p}(n)/(c\epsilon))}{\log(1/(p_e c))}\right)^{\log_2 d} \\
&\in \mathcal{O}(\mathsf{poly}(\log \mathsf{p}(n)/\epsilon))
\end{aligned}
$$

The threshold theorem

The analysis on the previous slide leads to the threshold theorem:

> *A quantum circuit containing $\mathsf{p}(n)$ gates may be simulated with probability of error at most ϵ using*
>
> $$\mathcal{O}(\mathsf{poly}(\log \mathsf{p}(n)/\epsilon)\mathsf{p}(n))$$
>
> *quantum gates on hardware whose gates fail with probability at most p_e, provided p_e is less than some constant threshold $p_e < p_{th}$.*

Note that we have proved this for the case of the depolarising channel, but owing to the digitisation of errors discussed in the last lecture, a general proof follows along similar lines.

Significance of the threshold theorem

The importance of the threshold theorem cannot be understated:

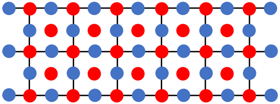
- When the probability of failure of each quantum gate is below the threshold, then we can correct errors as we go along so that the computation succeeds with high probability. Moreover, this error correction is efficient in the sense that only a poly-logarithmic increase in the number of gates in the circuit is incurred.
- This contrasts with analogue computing, which theoretically can achieve impressive performance improvements over digital computing, but in fact offers no significant improvements at all when the physical reality of noise, and the subsequent necessity of error correction are taken into account.
- It follows that a crucial goal for quantum computer hardware designers is to reduce the error rate of the quantum gates as much as possible; and for quantum computing theorists is to design codes with as high a threshold as possible.

However, in this analysis we have overlooked locality constraints, and these potentially hamper our ability to find good error correcting codes for quantum computing.

Error correction with locality constraints

Concatenated codes require parity-check measurements between the physical qubits used to encode logical qubits – however these physical qubits may be distant from each other in any actual quantum computer architecture. Thus SWAP gates would be required to move these qubits to be local. However, these SWAP gates would be on actual physical qubits, and thus not fault tolerant.

There is, however, an alternative to this concatenated code approach – surface codes work when qubits are laid out on a rectangular grid with only nearest-neighbour interactions possible, the surface code assigns every other qubit as a parity-check ancilla, which usually illustrated as:



with the red qubits on the faces and vertices as parity-check ancillas, and the blue qubits on the edges being the data qubits. The surface code only requires local parity-check measurements, and can encode logical qubits in an appropriate number of physical qubits.

State-of-the-art thresholds and gate error rates

- The threshold of the surface code is estimated to be 0.01.
- Ion-trap qubits in the University of Oxford currently have world-leading "fidelity" – with two-qubit gate error rates less than 0.001.
- So it follows that error rates below the surface code threshold have been achieved, and therefore all that is required, in principle, for fault tolerant quantum computing to be achieved is to scale up the number of qubits in the quantum computer.
- In practise there are a number of significant engineering (and some theoretical) hurdles to overcome. In particular, even though error rates below the threshold have been achieved, for practical purposes it is still important to increase the fidelity of the quantum operations further, so that fewer resources will be required to achieve fault tolerance.

Summary

In this lecture we have looked at:

- The elements of fault tolerant quantum computation, in particular constructing a fault tolerant universal gate-set.
- The threshold theorem.
- Using the surface code to achieve fault tolerance in physically realistic quantum computer architectures.
- State-of-the art code thresholds and gate error rates.