Quantum Programming Algorithms: Error Correction

Jens Palsberg

May 26, 2020

Outline

Hook: Quantum computers are noisy and their results are unreliable. Can we correct the errors that happen due to noise? While the hardware people are busy making more reliable qubits, the algorithms people are busy making better algorithms for error correction.

Purpose: Persuade you that we can correct many errors at the cost of adding many qubits.

Preview:

1. We will define two particular kinds of errors.
2. We will give three reasons why quantum error correction is hard.
3. We will define three schemes for error correction, culminating with the Shor code.

Transition to Body: Now let us talk about errors.

Main Point 1: We will define two particular kinds of errors.
[Bit flip errors]
[Phase flip errors]
[Arbitrary quantum errors]

Transition to MP2: In classical computing, we handle errors with replication and redundancy.

Main Point 2: We will list three reasons why quantum error correction is hard.
[We cannot clone a qubit]
[Measurement destroys quantumness]
[Errors are continuous]

Transition to MP3: So, what can we do about it?

Main Point 3: We will define three schemes for error correction, culminating with the Shor code.
[The three qubit bit flip code and the three qubit phase flip code]
[The Shor code]
[The Threshold Theorem]

Transition to Close: So there you have it.

Review: Quantum error correction is harder than classical error correction due to quantum limitations such as no-cloning. We can correct many errors but at the cost of adding many qubits.

Strong finish: More reliable quantum computing can be achieved by better hardware, better algorithms for error correction, or both. Given the huge reliability gap between classical computing and quantum computing, we need both.

Call to action: Now is a great time to do research on quantum error correction.

Detailed presentation

Hook: Quantum computers are noisy and their results are unreliable. In particular, they are much more unreliable than classical computers. Can we correct the errors that happen due to noise? While the hardware people are busy making more reliable qubits, the algorithms people are busy making better algorithms for error correction. Today I will explain quantum error correction in a way that ties together the No-Cloning theorem, the World's most reliable qubits from UCLA Physics, and the number $\frac{1}{36}$.

Purpose: Persuade you that we can correct many errors at the cost of adding many qubits.

Preview:

1. We will define two particular kinds of errors.
2. We will give three reasons why quantum error correction is hard.
3. We will define three schemes for error correction, culminating with the Shor code.

Transition to Body: Now let us talk about errors.

Main Point 1: We will define two particular kinds of errors.

[Bit flip errors]

In classical computing, a bit flip error maps 0 to 1, and 1 to 0.

In quantum computing, a bit flip error maps $|\psi\rangle$ to $X|\psi\rangle$, that is, $a|0\rangle + b|1\rangle$ to $a|1\rangle + b|0\rangle$.

[Phase flip errors]

In quantum computing, a phase flip error maps $|\psi\rangle$ to $Z|\psi\rangle$, that is, $a|0\rangle + b|1\rangle$ to $a|0\rangle - b|1\rangle$.

In classical computing, we have nothing like a phase flip error.

[Arbitrary quantum errors]

We can consider an arbitrary error on a single qubit. Such an arbitrary error can be anything from a tiny change to the phase to removing the qubit entirely and replacing it with garbage.

Transition to MP2: In classical computing, we handle errors with replication and redundancy.

Main Point 2: We will list three reasons why quantum error correction is hard.

[We cannot clone a qubit]

In classical computing we handle noise via encoding and decoding. For example, we can encode a bit as three copies of itself and then decode by majority vote. If at most one bit has flipped, we will get the right result. Notice that when errors are rare, one error is more likely than two.

In quantum computing, we cannot clone a qubit. So the idea of three copies of a qubit won't work.

[Measurement destroys quantumness]

The basis for doing a majority vote is that we can observe the things we are voting on. In quantum computing, we can measure the qubits but measurement destroys quantumness so we cannot recover the qubits we measured.

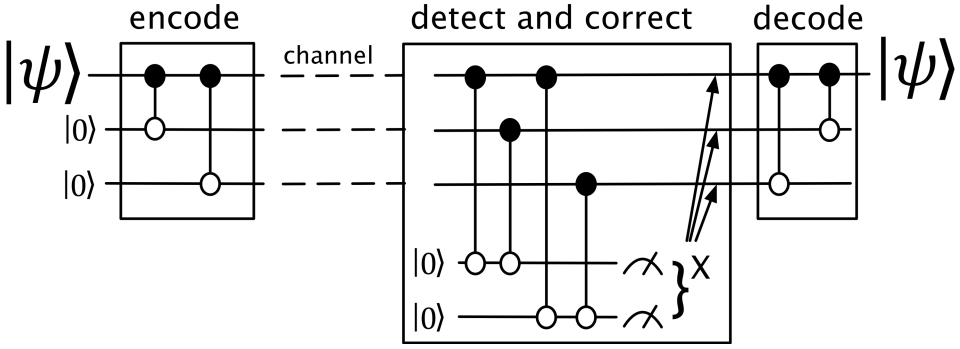
[Errors are continuous]
In classical computing, errors are discrete: 0 to 1, or 1 to 0.

In quantum computing, errors can be continuous and modify the complex numbers that describe the qubits.

Transition to MP3: So, what can we do about it?

Main Point 3: We will define three schemes for error correction, culminating with the Shor code.

[The three qubit bit flip code and the three qubit phase flip code]
If we have a bit flip error, then we can handle it with the bit flip code: map $a|0\rangle + b|1\rangle$ to $a|000\rangle + b|111\rangle$. The circuit labeled *encode* in this diagram does the mapping.



For now, we assume perfect encoding.

Now let us assume that at most one bit gets flipped and that we want to recover the quantum state. We do this in two steps.

First we do error detection and correction by computing and measuring two additional helper qubits. The outcome of the measurement of the two helper qubits is known as the *error syndrome*.

The first measured bit says whether the first and second bits of the encoded state are different (1) or the same (0). The second measured bit says whether the second and third bits of the encoded state are different (1) or the same (0).

If we are sending an encoding of the qubit $a|0\rangle + b|1\rangle$, then the error syndrome does not depend on a, b. Thus, we can learn about errors without learning about a, b, and thereby we preserve the superposition of $a|0\rangle + b|1\rangle$.

Given that at most one bit gets flipped, we will get one of the following four measurements and a corrective action to take:

Measurement	corrective action
00	do nothing
01	flip the third qubit
10	flip the second qubit
11	flip the first qubit

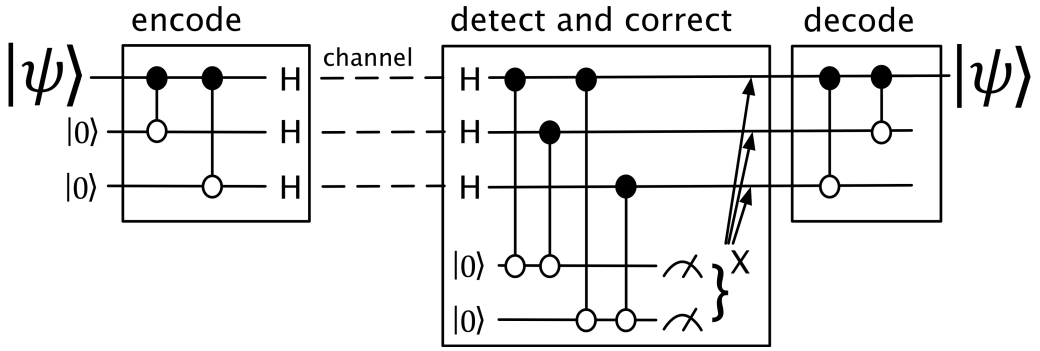
Second we bring the two additional qubits back to their original state using *decode* in the diagram above.

Let us check that these steps work in the case of flipping the second bit, that is, the channel delivers $a|010\rangle + b|101\rangle$. After introducing the two additional helper bits, the state is $a|01000\rangle + b|10100\rangle$. Now we can do the steps of *detect and correct*:

$$\begin{aligned} & & a|01000\rangle + b|10100\rangle \\ \mathrm{CX}(1,4) & : & a|01000\rangle + b|10110\rangle \\ \mathrm{CX}(2,4) & : & a|01010\rangle + b|10110\rangle \\ \mathrm{CX}(1,5) & : & a|01010\rangle + b|10111\rangle \\ \mathrm{CX}(3,5) & : & a|01010\rangle + b|10110\rangle \\ \text{measure the two helper qubits} & : & 10 \\ \text{flip the second qubit} & : & a|00010\rangle + b|11110\rangle \end{aligned}$$

Finally, *decode* brings the two additional qubits back to their original state $|00\rangle$. The end result is that the first qubit is, correctly, $a|0\rangle + b|1\rangle$.

If we have a phase flip error, then we can handle it with the phase flip code, which, intuitively, uses Hadamard to map the problem to the bit flip problem. Here is how it works.



Let us check that these steps work in the case of flipping the phase on the third qubit. Reminder: $Z|+\rangle = |-\rangle$ and $Z|-\rangle = |+\rangle$.

We map $a|0\rangle + b|1\rangle$ to $a|+++\rangle + b|---\rangle$, and then the channel delivers $a|++-\rangle + b|--+\rangle$. After introducing the two additional helper bits, the state is

$$a|++-\,00\rangle + b|--+\,00\rangle$$

Now we can do the steps of *detect and correct*:

$$\begin{aligned} & & a|++-\,00\rangle + b|--+\,00\rangle \\ H \otimes H \otimes H & : & a|00100\rangle + b|11000\rangle \\ \mathrm{CX}(1,4) & : & a|00100\rangle + b|11010\rangle \\ \mathrm{CX}(2,4) & : & a|00100\rangle + b|11000\rangle \\ \mathrm{CX}(1,5) & : & a|00100\rangle + b|11001\rangle \\ \mathrm{CX}(3,5) & : & a|00101\rangle + b|11001\rangle \\ \text{measure the two helper qubits} & : & 01 \\ \text{flip the third qubit} & : & a|00001\rangle + b|11101\rangle \end{aligned}$$

Finally, *decode* brings the two additional qubits back to their original state $|00\rangle$. The end result is that the first qubit is, correctly, $a|0\rangle + b|1\rangle$.

[The Shor code]
The Shor code corrects an arbitrary error on a single qubit. The idea is to first encode a qubit using the phase flip code and then encode each of the three qubits using the bit flip code. Thus, the encoding is: map $a|0\rangle + b|1\rangle$ to

$$\frac{a(|000\rangle + |111\rangle)^{\otimes 3} \; + \; b(|000\rangle - |111\rangle)^{\otimes 3}}{2\sqrt{2}}$$

The idea of using one code after the other is known as *concatenation*.

Intuitively, the Shor code can correct both a bit flip and a phase flip on any qubit. But the Shor code can do much more than that. First, the Shor code can correct the combination of a bit flip and phase flip on any qubit. Second, and more impressively, the Shor code can correct an arbitrary error on a single qubit. Thus, we are correcting a continuum of errors by correcting a discrete subset of those errors. This discretization of the errors is central to why quantum error-correction works.

[The Threshold Theorem]
Can we use error correction to create highly reliable quantum computation? The error correction will run on noisy quantum hardware so who will correct errors in the error correction? Can this work out, or will we have "turtles all the way down"?

The Threshold Theorem says that if the noise in individual quantum gates is below a known threshold, then we can efficiently perform an arbitrarily large quantum computation. This means that noise won't stop us from doing reliable large-scale quantum computation.

The idea of the proof is to concatenate codes. Each level of concatenation decreases the error rate. After k levels of concatenation, the error rate is vanishingly small. Specifically, for a computation of length T, we need $\log(\log T)$ levels of concatenation and thus polylog(T) extra qubits for high reliability.

So, the underlying concept of the Threshold Theorem is that by nesting error correction codes a sufficient number of times, we can make the logical error rate arbitrarily small.

How do we determine the threshold value? One can try a theoretical approach, which will tend to give a low threshold. One can try an experimental approach, possibly via simulation, which will tend to give a high threshold. In either case, one can make assumptions that will simplify the task but also give more approximate answers.

Some theoretical papers have reported a threshold value of 10^{-5}, while some simulation-based papers have reported a threshold value of 10^{-3}. Here, instead, we will do a back-of-the-envelope calculation.

Let us apply the idea of the Threshold Theorem to the case where the code state has two errors. In many cases, the two errors will result in a logical error, like when both qubits 1 and 2 have bit-flip errors. We can solve this using the idea of the Threshold Theorem, if we make a big assumption: all errors occur independently with small probability p. Note: experiments have shown that this assumption is sometimes wrong, but the following ideas can be adapted to cases beyond the big assumption.

Let us assume that p is the error rate of the original qubits. Now we can calculate:

$$\text{Prob(two errors occurring in the Shor code)} \;\; = \;\; (9 \text{ choose } 2) \times p^2 \;\; = \;\; 36p^2$$

So, if $p < \frac{1}{36}$, then Prob(two errors occurring in the Shor code) $< p$. In other words, if $p < \frac{1}{36}$, then the two-error rate for the encoding is less than the one-error rate for the original qubits. Notice that the assumption that $p < \frac{1}{36}$ is true on today's quantum computers.

We can improve further by adding another level of the Shor code and thereby get a logical error rate of $36 * (36 * p^2)^2$, which is smaller than $36p^2$. Thus, two levels of Shor code is better than one level of Shor code, though now we are up to $9 \times 9 = 81$ helper qubits for each logical qubit. So, if we pick the threshold value to be $\frac{1}{36}$, then we can nest the encoding for as many layers as we like, and make the logical error rate arbitrarily small. This nesting is the key to do quantum computation with faulty qubits and arbitrarily high accuracy.

So, is $\frac{1}{36}$ the threshold that we are looking for? No! The problem is the hidden assumption that the measurement of error syndromes can be done perfectly, which is unrealistic. This motivates the field of *fault-tolerant* quantum computation, which is an active and exciting field. Now we get into problems such as: how do we perform logical gates on the encoded qubits? So far, the best schemes for fault tolerance are highly complicated, require a lot of additional qubits, and depend on a tiny threshold.

Research shows that we definitely need a threshold of no more than 10^{-3}. For example, Craig Gidney from Google and Martin Ekerå from Sweden had an impressive paper in 2019. In that paper they showed how to factor 2,048 bit RSA integers in 8 hours on a model of a quantum computer with 20 million qubits and a gate error of 10^{-3}. We know that the World's most reliable qubits from UCLA Physics have an error rate of $(1 - 99.97)\%$, which is about 10^{-4}, so that is good. A big question is whether scaling down the error rate can go hand in hand with scaling up the number of qubits.

Overall, we need both the hardware people to make more reliable qubits and the algorithms people to make better algorithms for error correction. But the Threshold Theorem says that the stack of turtles has an end to it.

Transition to Close: So there you have it.

Review: Quantum error correction is harder than classical error correction due to quantum limitations such as no-cloning. We can correct many errors but at the cost of adding many qubits.

Strong finish: More reliable quantum computing can be achieved by better hardware, better algorithms for error correction, or both. Given the huge reliability gap between classical computing and quantum computing, we need both.

Call to action: Now is a great time to do research on quantum error correction.