Introduction to Quantum Error Correction and Fault Tolerance

Kurt Jacobs

Centre for Quantum Computer Technology, Centre for Quantum Dynamics, School of Science, Griffith University, Nathan 4111, Australia

Workshop on Quantum Information, NTU, Taipei, Dec 14-15, 2004

The text for this tutorial is chapter 10 of:
Nielsen and Chuang, (CUP, Cambridge, 2000)
Note: The workshop presentation included slides 1-15, 19-24 and 26

Griffith UNIVERSITY

Queensland Government

CENTRE FOR QUANTUM COMPUTER TECHNOLOGY
AUSTRALIAN RESEARCH COUNCIL CENTRE OF EXCELLENCE

Outline

1. Classical error correction
2. An example of quantum error correction
3. Shor’s Nine qubit code
4. Hamming Bound - other codes
5. General conditions for a quantum error correcting code
6. Fault tolerance
7. Recent developments

Classical Error Correction

While classical computers do not need error correction, as their components are highly reliable. However, classical information storage devices (e.g. CD's), and transmission devices (e.g. modems), use error correction to protect against noise.

A simple error correction procedure involves using 3 bits to encode the value of a single bit:

1 ---> 111
0 ---> 000

If there is an error in one of the qubits, then the data is completely undamaged, because we can still tell what the value of original bit is. (Note that this is possible because the 3 states which result when there is an error on one of the bits for the initial state 1 (which are 110,101,011) are different from those that result when the initial state is 0 (which are 001,010,100)

So, what is the probability that we will have an error which does corrupt the data, when we encode it in this way? To corrupt the data there must be two or more errors in the qubits. If the probability of the error in a single bit is p, and we assume that the errors in different bits are *independent*, then the probability of having an error in more than one qubit is $3p^2 + p^3$.

So the error correction procedure reduces the error probability from p to $3p^2 + p^3$.

--- If p is much less than 1, then $3p^2 + p^3$ is much less than p, and this error correction method will give a great advantage.

--- If p is close to 1, then we can use the method repeatedly, and reduce the error as small as desired. This is called *concatenation*.

Note that it is crucial that errors on different bits are independent, and is the fundamental reason that the procedure works.

Quantum Error Correction?

So is it possible to do the same thing for quantum systems. That is, to obtain a procedure which involves encoding the state of a quantum system in a number of such system, such that, when an error occurs in a single one of these systems, the state of the original system can be perfectly recovered?

There are a number of reason why one might think that this is not possible:

1. No cloning
2. Quantum errors are continuous
3. Measurement on a system in an unknown state alters the state.

However, we will find that despite these things, quantum error correction is possible.

A simple example of Quantum Error Correction

To successfully protect a bit we have to be able to recover the initial state, which, for a classical bit is one of just two possibilities. The state of a quantum bit however is a two-dimension continuum:

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

Now, say the error which can happen to a single bit is an operator which flips the state when written in the basis $\{|0\rangle, |1\rangle\}$. We will encode the state of the qubit in three qubits as

$$|C\rangle = a|0\rangle|0\rangle|0\rangle \; + \; b|1\rangle|1\rangle|1\rangle$$

Note that all the possible states $|C\rangle$ lie in a 2-dimensional subspace of the 3-qubit system. This is called the code space, or simply the *code*, since choosing this space completely specifies the code. Also we will call the encoded qubit the *logical qubit*, and the others the *physical qubits*.

Now imagine that an error occurs in one of the three qubits. The three possible states that result are:

$$|C_1\rangle = a|1\rangle|0\rangle|0\rangle \ + \ b|0\rangle|1\rangle|1\rangle$$
$$|C_2\rangle = a|0\rangle|1\rangle|0\rangle \ + \ b|1\rangle|0\rangle|1\rangle$$
$$|C_3\rangle = a|0\rangle|0\rangle|1\rangle \ + \ b|1\rangle|1\rangle|0\rangle$$

Now we see that each of these states lives in a different space. This is essentially a result of the properties of the classical code $\{000,111\}$. As a result of this, we can perform a measurement which projects the system onto one of these spaces. If the result of the measurement is the space in which $|C_3\rangle$ lives, then the result of the measurement is just $|C_3\rangle$

$$|C_3\rangle = a|0\rangle|0\rangle|1\rangle \ + \ b|1\rangle|1\rangle|0\rangle$$

Now we can correct the error by flipping the last bit! And we can just as easily correct the error for the other two subspaces.

Features:

-- Measurement tells us the final space
-- knowing this we can correct by applying a unitary
-- If in the original code space, no correction
-- Also the measurement actually tells us which qubit the error was in.
-- The important point is that for each space,
   the error maps are unitary, and independent of the initial state.

Like correcting a classical probability density
In hindsight, this process is not so surprising. This is because a classical error correction procedure can clearly correct for any probability distribution over the initial state of the bit. This is because if both states 0 and 1 are correctly preserved, then an initial probability distribution over those states is also preserved. Correcting the quantum state in the above example is similar to correcting a classical distribution.

Note that the error correction involves an operation which is conditional upon the subspace, and this does not actually require a measurement, but can be done with a suitable unitary operator.

A Continuum of Errors

We have seen that neither the impossibility of cloning nor the fact that measurements on unknown states cause disturbance impose a fundamental barrier to quantum error correction. But what about the fact that there are a continuum of possible errors.

Lets say that the coding system has N qubits:

-- Write the Pauli operators for the n^{th} qubit as X_n, Y_n,Z_n .
-- Assume that we have a code which can correct any single error caused by X_n, Y_n or Z_n for any n.

What happens for an arbitrary error operator E_n acting on qubit n?

-- Important fact: any operator on a single qubit is a linear combination of the Pauli operators X, Y, Z.

Thus an arbitrary operator on the n^{th} qubit is

$$E_n = e_{0n} I_n + e_{1n} X_n + e_{2n} Y_n + e_{3n} Z_n \ ,$$

where the subscript n denotes the qubit. So what happens now when we have an error due to one of the operators E_n (selected at random with probability p_n) ? Well, after the action of E we can make a measurement which projects the system onto one of the mutually orthogonal spaces to which the system is mapped by the actions of I_n (no error), X_n, Y_n and Z_n. Let us say that the result of the measurement is a projection onto the space resulting from the action of X_1. If we denote the projection operator onto this space by P_{X1} , and the initial encoded state as ρ then the result is

$$\Sigma_n \, p_n \, P_{X1} \, E_n \, \rho \, E_n^{\dagger} \, P_{X1}$$

$$= \Sigma_n \; p_n \, P_{X1} \, (e_{0n} I_n + e_{1n} X_n + e_{2n} Y_n + e_{3n} Z_n) \, \rho \, (e_{0n} I_n + e_{1n} X_n + e_{2n} Y_n + e_{3n} Z_n)^{\dagger} P_{X1}$$

$= \; p_1 \, |e_1|^2 \, X_1 \, \rho \, X_1$ hence the result is only an X error one qubit 1.

Hence, because the actions of all the operators I_n, X_n, Y_n and Z_n take the code space to mutually orthogonal spaces, once we make a measurement which projects the system onto one of these spaces the result is that only one of this *discrete* set of errors will have affected the system (so long as an arbitrary error E_n has occurred on just one of the qubits).

-- A continuum of errors is not a barrier to performing QEC.
-- A single code that corrects X, Y and Z will correct all errors.
-- So is such a code possible?

-- The 3-qubit code that corrects for X can easily be modified to correct for Y or Z, just by changing the basis in which the code is written.

Shor was the first to construct a code that could do all three [*Shor, PRA 52, 2493 (1995)*]. He *concatenated* the code which corrects X with the same code transformed to the Z basis so that it corrects Z. (That is, to first encode the single qubit using three qubits with a code that corrects Z (say) and then to encode each of these three qubits using three qubits (for a total of nine qubits) with a code that corrects X.)

Shor’s Nine Qubit Code

So to obtain Shor’s code we first encode the logical qubit using the code which protects against *Z*, otherwise known as a *phase-flip* error. That is

$$|\psi\rangle = a|0\rangle + b|1\rangle \quad \text{-->} \quad |C_y\rangle = a|+\rangle|+\rangle|+\rangle \; + b|-\rangle|-\rangle|-\rangle$$

where $|+\rangle \propto |0\rangle + |1\rangle$ and $|-\rangle \propto |0\rangle - |1\rangle$ are the eigenstates of *Z* (we don’t worry about normalization). Then, we encode each of the three coding states using the code which corrects *X* errors (otherwise known as *bit-flip* errors). Then we have

$$|C_{xy}\rangle \propto \; a(|000\rangle + |111\rangle)\,(|000\rangle + |111\rangle)\,(|000\rangle + |111\rangle) \; +$$

$$b(|000\rangle - |111\rangle)\;(|000\rangle - |111\rangle)\;(|000\rangle - |111\rangle)$$

Now, to see that this corrects for both bit-flip and phase-flip errors, we just need to understand why the action of both *X* and *Z* on any bit take the system to mutually orthogonal spaces.

-- Each of the X operations take the system to orthogonal spaces - to see this look at the action on each of the three-qubit blocks.

-- A phase-flip maps $|0\rangle$ to $|0\rangle$ and $|1\rangle$ to - $|1\rangle$. Since it keeps us within the space which is made up of products of $|000\rangle$ and $|111\rangle$, all the resulting states are orthogonal to those produced by X. Also, we note that Z errors on different three-qubit blocks map to mutually orthogonal spaces.

-- However, Z errors on different qubits within a block map to the same space. But more than this, they map in an identical way - that is, they produce the same final state. Hence, we can still correct for each of them, once we know the final space, since the map required to correct the qubit is the same for all of them!

Here is the code again for reference:

$$\begin{aligned}|C_{xy}\rangle \propto\ & a(|000\rangle + |111\rangle)\,(|000\rangle + |111\rangle)\,(|000\rangle + |111\rangle)\ + \\ & b(|000\rangle - |111\rangle)\,(|000\rangle - |111\rangle)\,(|000\rangle - |111\rangle)\end{aligned}$$

Correcting Y errors

Now, what about correcting Y errors? Well, the first thing to note is that $Y \propto XZ$. So we can correct a Y error if we can correct the situation when both an X error and a Z error happen simultaneously. Inspection of the action of both of these together shows that this takes us yet again to a whole different set of orthogonal spaces where necessary, hence Y can be corrected too, and hence all errors on a single qubit can be corrected!

Definition of error syndrome

To sum up, we project the system onto one of the possible subspaces, and this tells us how the system has been affected by the error. The result of the measurement (i.e. the subspace) is called the error syndrome. Once we know the error syndrome, we can correct the error.

A degenerate code

Now, the Shoe code is an example of a degenerate quantum code. That is, a code in which some of the errors map to the same space, which is possible because the action of these different errors on the code state are identical. This is not something that occurs in classical codes.

Other Codes

Shor's code is not the smallest code which can be used to completely protect a qubit against single-qubit errors. If we assume that our codes are non-degenerate, then it is easy to obtain a lower bound of the number of qubits required for a code by counting the number of orthogonal 2D subspaces which are required. To correct for I (no error), and X_n, Y_n and Z_n on each qubit need $1 + 3n$ two dimensional subspaces. Now a code which has n qubits has a dimension of 2^n, so we must have $2^n \geq 2(3n+1)$. This is only satisfied for $n \geq 5$. This is called the quantum Hamming but is not a strict lower bound, because quantum codes can be degenerate. However, it is possible to derive a more sophisticated bound, the quantum Singleton bound which is strict, and confirms that $n \geq 5$.

There are a number of methods that have been developed to construct quantum codes. There are the Calderbank-Shor-Steane codes, which employ classical linear coding theory, and the stabilizer codes, developed by Gottesman. There is a stabilizer code which will correct single qubit errors using minimum 5 qubits.

General Conditions for QEC

--- Bennett, Di Vincenzo, Smolin and Wootters, PRA 54, 3824 (1996)
--- Knill and Laflamme PRA 55, 900 (1997)

The quantum code we have analyzed worked because a finite set of unitary errors mapped the code space to mutually orthogonal spaces, which allowed the different errors to be identified and isolated, and the fact that any error operation could be written as a linear combination of a finite set of unitaries operators.

General necessary and sufficient conditions have been derived which a code space must satisfy to be able to correct a set of errors. If the set of errors is given by the set of operators $\{E_i\}$, and the projector onto the code space is P, then the condition is

$$P\, E_i^\dagger E_j P = a_{ij} P$$

where a_{ij} is a Hermitian matrix.

To show that these conditions are sufficient, we first diagonalise the matrix a_{ij} . If the matrix u_{ij} is the transform which diagonalises it, and we set

$$\Sigma_k F_k = u_{kl} E_l$$

then

$$\begin{aligned} P F_k^{\dagger} F_l P &= \Sigma_{ij} u_{ki} u^*_{lj} P E_i^{\dagger} E_j P \\ &= \Sigma_{ij} u_{ki} u^*_{lj} a_{ij} P \\ &= d_{kl} P \end{aligned}$$

where d_{kl} is diagonal. Now what is the action of F_l on P? Well, the trick is to use the polar decomposition theorem on the product $F_l P$:

$$F_l P = U_l (P F_k^{\dagger} F_l P)^{1/2} = (d_{kl})^{1/2} P$$

So the action is just a unitary on the code space, and what is more, The spaces to which the code is mapped by each of these unitaries are mutually orthogonal, since

$$P\, F_k^\dagger\, F_l\, P \;=\; d_{kl}\, P$$

So the code can correct the errors introduced by the E_i.

Necessity: Assume there is an operation described by the operators $\{R_i\}$ and consider the operation $\{E_i\, P\}$. Then, for all ρ we must have

$$\Sigma_{ij}\, R_i E_j\, P\, \rho\, P\, E_j^\dagger\, R_i^\dagger \;\propto\; P\, \rho\, P$$

Now we invoke a theorem which says that the operator representations for two identical operations are related by a unitary transformation of the sets of operators. Hence

$$R_i E_j\, P \;= c_{ij}\, P \quad \text{and thus also} \quad P\, E_k^\dagger\, R_i^\dagger = c^*{}_{ik}\, P$$

Putting these together we have $P\, E_k^\dagger\, R_i^\dagger\, R_i E_j\, P \;= c^*{}_{ik}\, c_{ij}\, P$ and with $\Sigma_i\, R_i^\dagger\, R_i = I$ we have $P\, E_k^\dagger E_j\, P \;= c^*{}_{ik}\, c_{ij}\, P = a_{kj}\, P$.

Fault Tolerance

What about errors during a computation?

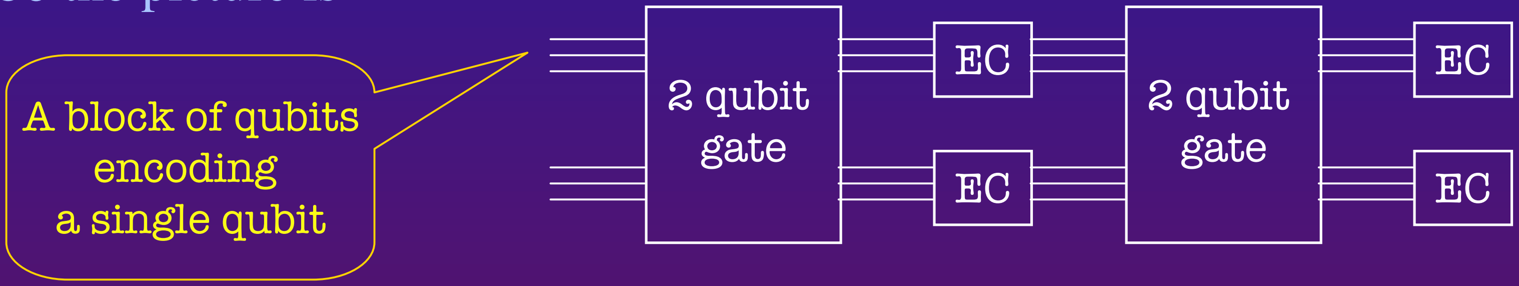
So we have seen that we can use quantum error correction to correct errors in a quantum memory. However, to perform quantum computing we must be able to correct errors which are happening while the computation proceeds, including errors in the gates and even the error correction procedures.

The ability to use error correction so as to reduce the effect of errors on a computation to an arbitrary low level without requiring resources which scale exponentially with the problem size or the level of error is referred to as the ability to perform *fault tolerant* computation.

To work out how fault tolerance can be achieved, the following approach has been taken:

1. All gates act on encoded data.
2. Perform error correction after each gate operates.

So the picture is



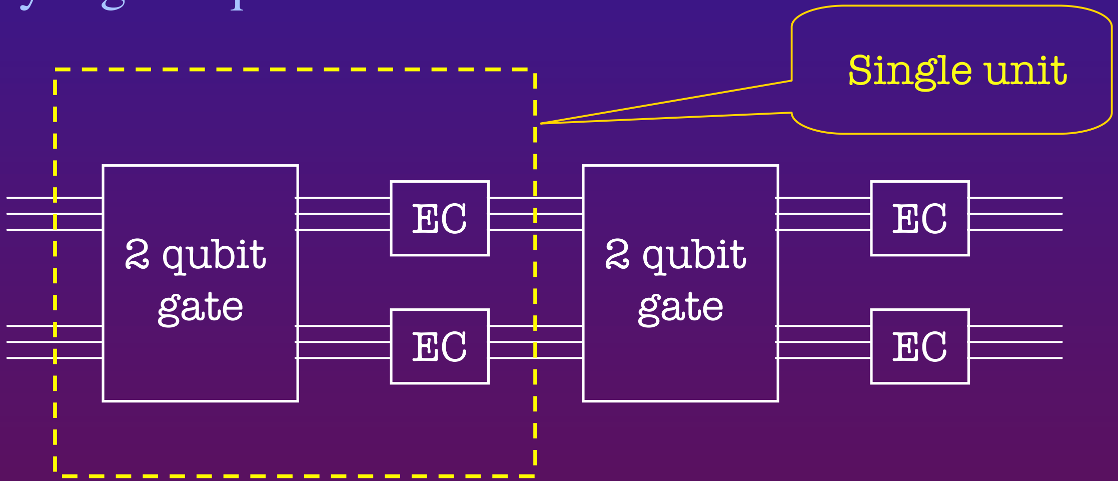
We now ask that the gates and EC steps have the following property:

A single failure in the operation causes at most 1 error in each output block.

It is also required that the measurement part of the EC step fails with probability p^2. Operations with this property is called *fault tolerant*.

By a single error in a gate we mean an error in a single unitary operation involving 2 physical qubits in an encoding block. We will call these physical gates. We assume as before that the errors in each of our physical qubits and physical gates are independent.

Now we note that if there is a single error in any block of the final output the computation is fine. So the computation is only in error if there is more than one error in any logical qubit.



So we consider the probability that a single unit will produce more than one error in any output block. If the probability of an error in any single input block is p, then the probability of two in the output is p^2.

This is because, one of the elements in the unit must fail in addition to the error in the input block, or, if there are no errors in the input block, then two elements of the unit must fail.

As a result, so long as the probability of any error in an input block is of order p, this remains of order p for the output blocks. The probability of an error in more than one output block is also higher than first order in p.

Put another way, we have a quantum computation that only fails if

1. Two elements in a given unit fail, which happens with probability mp^2, where m is the number of pairs of elements in a single unit, or

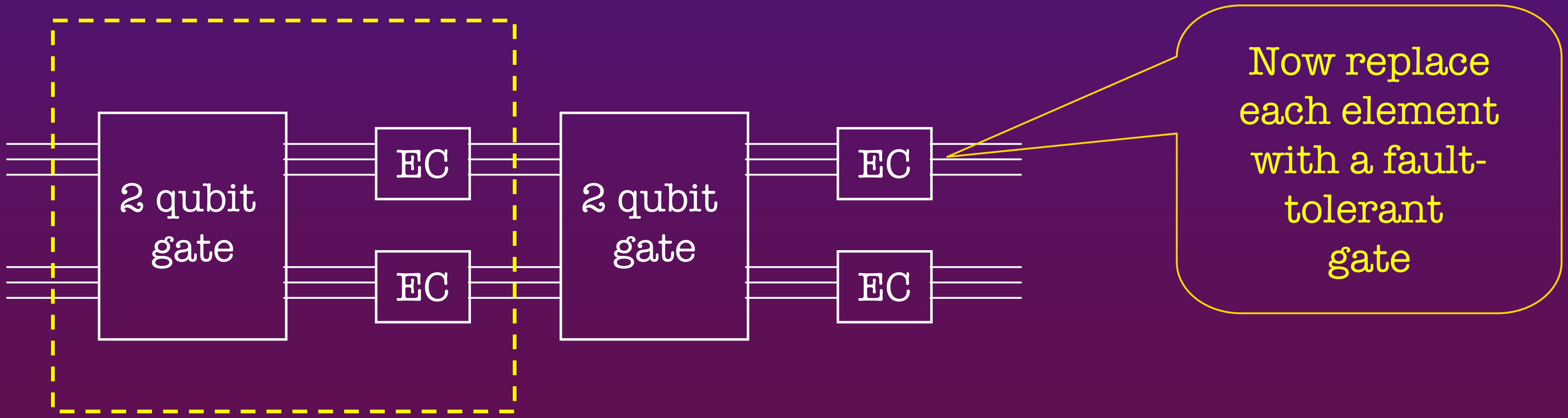
2. If pairs of consecutive units fail -- but this scales as the number of units, just as if we hadn't used any error correction.

So we win so long as $p > 1/m \approx 10^4$ for most fault tolerant constructions.

Concatenation

Now, each of our units is a gate for the logical qubits which are the blocks. So, if we wish we can replace all the physical bits in our fault tolerant scheme with blocks which encode logical qubits, and all the physical gates with fault tolerant gates acting on those logical qubits.

If we do this, then each of our units will only fail with probability $m\,p^2$. As a result, the top level of our computation will only fail with probability $m(m\,p^2)^2$. If we *concatenate* in this recursive fashion *k* times, then the error probability is reduced to $(m\,p)^{2\wedge k}\,/m$ which is doubly exponential in *k*. The size of the circuit, however, is d^k which is merely exponential in *k*.



Now replace each element with a fault-tolerant gate

The threshold theorem

Now, lets say we have a computation which requires N gates. For this to succeed with probability $1-\varepsilon$ we need the error probability for each gate to be less than ε/N. Thus we need k such that

$$(m\,p)^{2^\wedge k}/m \;<\; \varepsilon/N$$

which we can always do for some k so long as p is smaller than the threshold $1/m$. Now, the number of gates required to do this is Nd^k.
Finding k by taking logs in the above formula, we find that this scales as

$$Nd^k \;\propto\; O(\; log(N/\varepsilon)N\;)$$

This is the threshold theorem - that given the error probability p per gate is less than some threshold, a quantum circuit containing N gates may be simulated with error at most ε, with a number of gates which is polynomial in N/ε.

Note this says that quantum computing is possible in principle - not that it is necessarily practical.

Fault tolerant gates: A simple example

The construction of fault tolerant units (that is, units which implement single gates with a reduced error probability) required fault tolerant gates - that is, gates in which a single error in one of their basic elements caused at most a single error in each output block.

Current proofs of the threshold theorem use such fault tolerant gates, and thus a universal set of which has thus been found. We wont talk about these in detail, but here is a simple example of a fault tolerant C-NOT gate operating on the 3-qubit bit-flip correcting code: