1 General theory of quantum error correction and fault-tolerance

Some introductory examples of quantum error correction (QEC) methods were given in a previous lecture. Here we will give a summary of the simplest aspects of the more general theory.

QEC is based on three central ideas: digitization of noise, the manipulation of error operators and syndromes, and quantum error correcting code (QECC) construction. The degree of success of QEC relies on the physics of noise; we will turn to this after discussing the three central ideas.

1.1 Digitization of noise

"Digitization of noise" is based on the observation that *any* interaction between a set of qubits and another system (such as the environment) can be expressed by:

$$|\phi\rangle |\psi\rangle_e \to \sum_i (E_i |\phi\rangle) |\psi_i\rangle_e \tag{1}$$

where each 'error operator' E_i is a tensor product of Pauli operators acting on the qubits, $|\phi\rangle$ is the initial state of the qubits, and $|\psi\rangle_e$ are states of the environment, not necessarily orthogonal or normalised. We thus express general noise and/or decoherence in terms of Pauli operators σ_x, σ_y, σ_z acting on the qubits. These will be written $X \equiv \sigma_x$, $Z \equiv \sigma_z$, $Y \equiv -i\sigma_y = XZ$.

To write tensor products of Pauli matrices acting on n qubits, we introduce the notation $X_u Z_v$ where u and v are n-bit binary vectors. The non-zero coordinates of u and v indicate where X and Z operators appear in the product. For example,

$$X \otimes I \otimes Z \otimes Y \otimes X \equiv X_{10011} Z_{00110}. \tag{2}$$

Error correction is a process which takes a state such as $E_i |\phi\rangle$ to $|\phi\rangle$. Correction of X errors takes $X_u Z_v |\phi\rangle$ to $Z_v |\phi\rangle$; correction of Z errors takes $X_u Z_v |\phi\rangle$ to $X_v |\phi\rangle$. Putting all this together, we discover the highly significant fact that to correct *the most general possible* noise (eq. (1)), it is sufficient to correct just X and Z errors.

1.2 Error operators, stabilizer, and syndrome extraction

We will now examine the mathematics of error operators and syndromes, using the insightful approach put forward by Gottesman [10] and Calderbank *et. al.* [6, 5], building on the first discoveries of Steane [2, 4] and Calderbank and Shor [1, 3].

Consider the set $\{I, X, Y, Z\}$ consisting of the identity plus the three Pauli operators. The Pauli operators all square to I: $X^2 = Y^2 = Z^2 = I$, and have eigenvalues ± 1. Two members of the set only ever commute ($XI = IX$) or anticommute: $XZ = -ZX$. Tensor products of Pauli operators, i.e. error operators, also square to one and either commute or anticommute. N.B. the term 'error operator' is here just a shorthand for 'product of Pauli operators'; such an operator will sometimes play the role of an error, sometimes of a parity check, c.f. classical coding theory.

If there are n qubits in the quantum system, then error operators will be of *length* n. The *weight* of an error operator is the number of terms not equal to I. For example $X_{10011}Z_{00110}$ has length 5, weight 4.

Let $\mathcal{H} = \{M\}$ be a set of commuting error operators. Since the operators all commute, they can have simultaneuous eigenstates. Let $\mathcal{C} = \{|u\rangle\}$ be the orthonormal set of simultaneous eigenstates all having eigenvalue +1:

$$M|u\rangle = |u\rangle \quad \forall u \in \mathcal{C},\ \forall M \in \mathcal{H} \tag{3}$$

The set $\mathcal{C}$ is a quantum error correcting code, and $\mathcal{H}$ is its *stabilizer*. The orthonormal states $|u\rangle$ are termed *code vectors* or *quantum codewords*. In what follows, we will restrict attention to the case that $\mathcal{H}$ is a group. Its size is 2^{n-k}, and it is spanned by $n-k$ linearly independent members of $\mathcal{H}$. In this case $\mathcal{C}$ has 2^k members, so it encodes k qubits, since its members span a 2^k dimensional subspace of the 2^n dimensional Hilbert space of the whole system. A general state in this subspace, called an *encoded state* or *logical state*, can be expressed as a superposition of the code vectors:

$$|\phi\rangle_L = \sum_{u\in\mathcal{C}} a_u |u\rangle \tag{4}$$

Naturally, a given QECC does not allow correction of all possible errors. Each code allows correction of a particular set $\mathcal{S} = \{E\}$ of *correctable errors*. The task of code construction consists of finding codes whose correctable set includes the errors most likely to occur in a given physical situation. We will turn to this important topic in the next section. First, let us show how the correctable set is related to the stabilizer, and demonstrate how the error correction is actually achieved.

First, error operators in the stabilizer are all correctable, $E \in \mathcal{S}\ \forall\ E \in \mathcal{H}$, since these operators actually have no effect on a general logical state (4). If these error operators are themselves the only terms in the noise of the system under consideration, then the QECC is a noise-free subspace, also called decoherence-free subspace [] of the system.

There is a large set of further errors which do change encoded states but are nevertheless correctable by a process of extracting an error syndrome, and then acting on the system depending on what syndrome is obtained. We will show that $\mathcal{S}$ can be any set of errors $\{E_i\}$ such that every product E_1E_2 of two members is either in $\mathcal{H}$, or anticommutes with a member of $\mathcal{H}$. To see this, take the second case first:

$$E_1E_2M = -ME_1E_2 \ \text{ for some } M \in \mathcal{H}. \tag{5}$$

We say that the combined error operator E_1E_2 is *detectable*. This can only happen if

$$\begin{aligned} \text{either} \quad & \{ME_1 = -E_1M,\ \ ME_2 = E_2M\} \\ \text{or} \quad & \{ME_1 = E_1M,\ \ ME_2 = -E_2M\} \end{aligned} \tag{6}$$

To extract the syndrome we measure all the observables in the stabilizer. To do this, it is sufficient to measure any set of $n-k$ linearly independent M in $\mathcal{H}$. Note that such a measurement has no effect on a state in the encoded subspace, since such a state is already an eigenstate of all these observables. The measurement projects a noisy state onto an eigenstate of each M, with eigenvalue ± 1. The string of $n-k$ eigenvalues is the syndrome. Equations (6) guarantee that E_1 and E_2 have different syndromes, and so can be distinguished from each other. For, when the observable M is measured on the corrupted state $E|\phi\rangle_L$, (6) means a different eigenvalue will be obtained when $E = E_1$ than when $E = E_2$. Therefore, the error can be deduced from the syndrome, and reversed by re-applying the deduced error to the system (taking advantage of the fact that error operators square to 1).

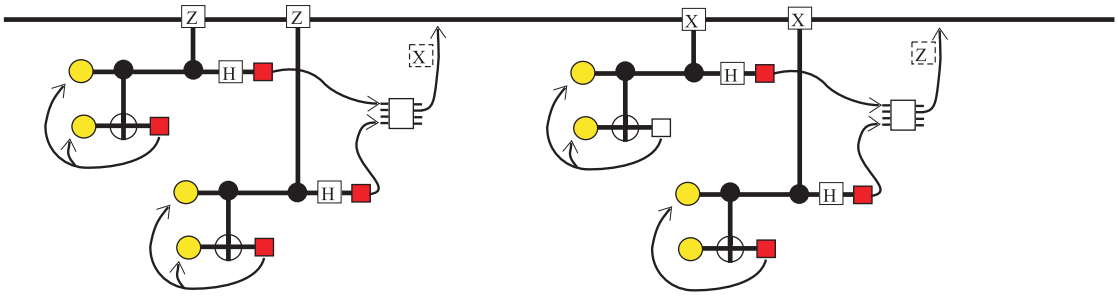
Let us see how this whole process looks when applied to a general noisy encoded state. The noisy state is

$$\sum_i \left(E_i \left|\phi\right\rangle_L\right) \left|\psi_i\right\rangle_e \tag{7}$$

The syndrome extraction can be done most simply by attaching an $n-k$ qubit ancilla a to the system, and storing in it the eigenvalues by a sequence of CNOT gates and Hadamard rotations. The exact network can be constructed either by thinking in terms of parity check information stored into the ancilla, or by the following standard eigenvalue measurement method. To extract the $\lambda = \pm 1$ eigenvalue of operator M, prepare an ancilla in $(|0\rangle + |1\rangle)/\sqrt{2}$. Operate controlled-$M$ with ancilla as control, system as target, then Hadamard rotate the ancilla. The final state of the ancilla is $[(1+\lambda)\left|0\right\rangle + (1-\lambda)\left|1\right\rangle]/2$. Carrying out this process for the $n-k$ operators M which span $\mathcal{H}$, the effect is to couple system and environment with the ancilla as follows:

$$\left|0\right\rangle_a \sum_i \left(E_i \left|\phi\right\rangle_L\right) \left|\psi_i\right\rangle_e \rightarrow \sum_i \left|s_i\right\rangle_a \left(E_i \left|\phi\right\rangle_L\right) \left|\psi_i\right\rangle_e . \tag{8}$$

The s_i are $(n-k)$-bit binary strings, all different if the E_i all have different syndromes. A projective measurement of the ancilla will collapse the sum to a single term taken at random: $\left|s_i\right\rangle_a \left(E_i \left|\phi\right\rangle_L\right) \left|\psi_i\right\rangle_e$, and will yield s as the measurement result. Since there is only one E_i with this syndrome, we can deduce the operator E_i which should now be applied to correct the error!



A complete fault tolerant error correction network, for single-error-correcting code

Figure 1: Figure 1

This remarkable process can be understood as first forcing the general noisy state to 'choose' among a discrete set of errors, via a projective measurement, and then reversing the particular discrete error 'chosen' using the fact that the measurement result tells us which one it was. Alternatively, the correction can be accomplished by a unitary evolution consisting of controlled gates with ancilla as control, system as target, effectively transferring the noise (including entanglement with the environment) from system to ancilla.

We left out of the above the other possibility mentioned just before eq. (5), namely that

$$E_1 E_2 \in \mathcal{H}. \tag{9}$$

In this case E_1 and E_2 will have the same syndrome, so are indistinguishable in the syndrome extraction process. However, this does not matter! We simply interpret the

common syndrome of these two errors as an indication that the corrective operation E_1 should be applied. If it was E_1 that occured, this is obviously fine, while if in fact E_2 occured, the final state is $E_1 E_2 |\phi\rangle_L$ which is also correct! This situation has no analogue in classical coding theory. The quantum codes which take advantage of it are termed *degenerate* and are not constrained by the quantum Hamming bound.

The discussion based on the stabilizer is useful because it focusses attention on operators rather than states. Quantum codewords are nevertheless very interesting states, having a lot of symmetry and interesting forms of entanglement. The codewords in the QECC can readily be shown to allow correction of the set $\mathcal{S}$ if and only if [7, 9]

$$\langle u| E_1 E_2 |v\rangle = 0 \quad (10)$$

$$\langle u| E_1 E_2 |u\rangle = \langle v| E_1 E_2 |v\rangle \quad (11)$$

for all $E_1, E_2 \in \mathcal{S}$ and $|u\rangle, |v\rangle \in \mathcal{C}, |u\rangle \neq |v\rangle$. In the case that $E_1 E_2$ always anticommutes with a member of the stabilizer, we have $\langle u| E_1 E_2 |u\rangle = \langle u| E_1 E_2 M |u\rangle = -\langle u| M E_1 E_2 |u\rangle = -\langle u| E_1 E_2 |u\rangle$, therefore $\langle u| E_1 E_2 |u\rangle = 0$. This is a nondegenerate code; all the code vectors and their erroneous versions are mutually orthogonal, and the quantum Hamming bound must be satisfied.

1.3 Code construction

The power of QEC results from the physical insights and mathematical techniques already discussed, combined with the fact that useful QECCs can actually be found. Code construction is itself a subtle and interesting area, which we will merely introduce here.

First, recall that we require the members of the stabilizer all to commute. It is easy to show that $X_u Z_v = (-1)^{u \cdot v} Z_v X_u$, where $u \cdot v$ is the binary parity check operation, or inner product between binary vectors, evaluated in $GF(2)$. From this, $M = X_u Z_v$ and $M' = X_{u'} Z_{v'}$ commute if and only if

$$u \cdot v' + v \cdot u' = 0 \quad (12)$$

The stabilizer is completely specified by writing down the $n - k$ linearly independent error operators which span it. It is convenient to write these error operators by giving the binary strings u and v which indicate the X and Z parts, in the form of two $(n-k) \times n$ binary matrices H_x, H_z. The whole stabilizer is then uniquely specified by the $(n-k) \times 2n$ binary matrix

$$H = (H_x | H_z) \quad (13)$$

and the requirement that the operators all commute (i.e. $\mathcal{H}$ is an abelian group) is expressed by

$$H_x H_z^T + H_z H_x^T = 0 \quad (14)$$

where ^{T} indicates the matrix transpose.

The matrix H is the analogue of the parity check matrix for a classical error correcting code. The analogue of the generator matrix is the matrix $G = (G_x | G_z)$ satisfying

$$H_x G_z^T + H_z G_x^T = 0. \quad (15)$$

In other words, H and G are duals with respect to the inner product defined by (12). G has $n + k$ rows. H may be obtained directly from G by swapping the X and Z parts and extracting the usual binary dual of the resulting $(n+k) \times 2n$ binary matrix.

Note that (15) and (14) imply that G contains H. Let $\mathcal{G}$ be the set of error operators generated by G, then also $\mathcal{G}$ contains $\mathcal{H}$.

Since by definition (15), all the members of $\mathcal{G}$ commute with all the members of $\mathcal{H}$, and since (by counting) there can be no further error operators which commute with all of $\mathcal{H}$, we deduce that all error operators not in $\mathcal{G}$ anticommute with at least one member of H. This leads us to a powerful observation: if all members of $\mathcal{G}$ (other than the identity) have weight at least d, then all error operators (other than the identity) of weight less than d anticommute with a member of $\mathcal{H}$, and so are detectable. Such a code can therefore correct all error operators of weight less than $d/2$.

What if the only members of $\mathcal{G}$ having weight less than d are also members of $\mathcal{H}$? Then the code can still correct all error operators of weight less than $d/2$, using property (9) (a degenerate code). The weight d is called the minimum distance of the code.

The problem of code construction is thus reduced to a problem of finding binary matrices H which satisfy (14), and whose duals G, defined by (15), have large weights. We will now write down such a code by combining well-chosen classical binary error correcting codes:

$$H = \left(\begin{array}{c|c} H_2 & 0 \\ 0 & H_1 \end{array}\right), \qquad G = \left(\begin{array}{c|c} G_1 & 0 \\ 0 & G_2 \end{array}\right). \tag{16}$$

Here H_i, $i = 1, 2$, is the check matrix of the classical code C_i generated by G_i. Therefore $H_i G_i^T = 0$ and (15) is satisfied. To satisfy commutativity, (14), we force $H_1 H_2^T = 0$, in other words, $C_2^\perp \subset C_1$. By construction, if the classical codes have size k_1, k_2, then the quantum code has size $k = k_1 + k_2 - n$. The quantum codewords are

$$|u\rangle_L = \sum_{x \in C_2^\perp} |x + u \cdot D\rangle \tag{17}$$

where u is a $k-bit$ binary word, x is an $n-bit$ binary word, and D is a $(k \times n)$ matrix of coset leaders. These are the CSS (Calderbank Shor Steane) codes. Their significance is first that they can be efficient, and second that they are useful in fault-tolerant computing (see below).

By "efficient" we mean that there exist codes of given d/n whose rate k/n remains above a finite lower bound, as $k, n, d \to \infty$. The CSS codes have $d = \min(d_1, d_2)$. If we choose the pair of classical codes in the construction to be the same, $C_1 = C_2 = C$, then we are considering a classical code which contains its dual. A finite lower bound for the rate of such codes can be shown to exist [3]. This is highly significant: it means that QEC can be a very powerful method to suppress noise (see next section).

There exist QECCs more efficient than CSS codes. Good codes can be found by extending CSS codes, and by other methods. For illustration, we finish this section with the stabilizer and generator of the $[[n, k, d]] = [[5, 1, 3]]$ perfect code. It encodes a single qubit ($k = 1$), and corrects all errors of weight 1 (since $d/2 = 1.5$).

$$H = \left(\begin{array}{c|c} 11000 & 00101 \\ 01100 & 10010 \\ 00110 & 01001 \\ 00011 & 10100 \end{array}\right), \qquad G = \left(\begin{array}{c|c} H_x & H_z \\ 11111 & 00000 \\ 00000 & 11111 \end{array}\right). \tag{18}$$

1.4 The physics of noise

Noise and decoherence is itself a large subject. Here we will simply introduce a few basic ideas, in order to clarify what QEC can and cannot do. By 'noise' we mean simply any unknown or unwanted change in the density matrix of our system.

The statement (1) about digitization of noise is equivalent to the statement that any interaction between a system of qubits and its environment has the form

$$H_I = \sum_i E_i \otimes H_i^e \tag{19}$$

where the operators H_i^e act on the environment. Under the action of this coupling, the density matrix of the system (after tracing over the environment) evolves from ρ_0 to $\sum_i a_i E_i \rho_0 E_i$. QEC returns all terms of this sum having correctable E_i to ρ_0. Therefore, the fidelity of the corrected state, compared to the noise-free state ρ_0, is determined by the sum of all coefficients a_i associated with uncorrectable errors.

For a mathematically thorough analysis of this problem, see [7, 8]. The essential ideas are as follows. Noise is typically a continuous process affecting all qubits all the time. However, when we discuss QEC, we can always adopt the model that the syndrome is extracted by a projective measurement. Any statement such as 'the probability that error E_i occurs' is just a short-hand for 'the probability that the syndrome extraction projects the state onto one which differs from the noise-free state by error operator E_i'. We would like to calculate such probabilities.

To do so, it is useful to divide up (19) into a sum of terms having error operators of different weight:

$$H_I = \sum_{\mathrm{wt}(E)=1} E \otimes H_E^e + \sum_{\mathrm{wt}(E)=2} E \otimes H_E^e + \sum_{\mathrm{wt}(E)=3} E \otimes H_E^e + \ldots \tag{20}$$

There are $3n$ terms in the first sum, $3^2 n!/(2!(n-2)!)$ terms in the second, and so on. The strength of the system-environment coupling is expressed by coupling constants which appear in the H_E^e operators. In the case that only the weight 1 terms are present, we say the environment acts independently on the qubits: it does not directly produce correlated errors across two or more qubits. In this case, errors of all weights will still appear in the density matrix of the noisy system, but the size of the terms corresponding to errors of weight w will be $O(\epsilon^{2w})$, where ϵ is a parameter giving the system-environment coupling strength.

Since QEC restores all terms in the density matrix whose errors are of weight $\leq t = (d-1)/2$, the fidelity of the corrected state, in the uncorrelated noise model, can be estimated as one minus the probability $P(t+1)$ for the noise to generate an error of weight $t+1$. This is probability is approximately

$$P(t+1) \simeq \left(3^{t+1} \binom{n}{t+1} \epsilon^{t+1} \right)^2 \tag{21}$$

when all the single-qubit error amplitudes can add coherently (i.e. the qubits share a common environment), or

$$P(t+1) \simeq 3^{t+1} \binom{n}{t+1} \epsilon^{2(t+1)} \tag{22}$$

when the errors add incoherently (i.e. either seperate environments, or a common environment with couplings of randomly changing phase). The significance of (21) and (22) is that they imply QEC works extremely well when t is large and $\epsilon^2 < t/3n$. Since good codes exist, t can in fact tend to infinity while t/n and k/n remain fixed. Therefore as long as the noise per qubit is below a threshold around $t/3n$, almost perfect recovery of the state is possible. The ratio t/n constrains the rate of the code through the quantum Hamming bound or its cousins.

Such uncorrelated noise is a reasonable approximation in many physical situations, but we need to be careful about the degree of approximation, since we are concerned with

very small terms of order ϵ^d. If we relax the approximation of completely uncorrelated noise, equations (21) and (22) remain approximately unchanged, if and only if the coupling constants in (20) for errors of weight t are themselves of order $\epsilon^t/t!$.

A very different case in which QEC is also highly successful is when a set of correlated errors, also called burst errors, dominate the system-environment coupling, but we can find a QEC whose stabilzer includes all these correlated errors. This is sometimes called 'error avoiding' rather than 'error correction' since by using such a code, we don't even need to correct the logical state: it is already decoupled from the environment. The general lesson is that the more we know about the environment, and the more structure there exists in the system-environment coupling, the better able we are to find good codes.

1.5 Fault tolerant quantum computation

The above discussion of QEC is relevent to high-fidelity communication down noisy quantum channels, but it is not yet clear how relevent it may be to quantum computing. This is because so far we have assumed the quantum operations involved in syndrome extraction are themselves noise-free. Therefore we are using processing power to combat noise, but it is not clear what degree of precision of the processing is necessary in order to gain something.

Fault tolerant computation is concerned with processing information reliably even when every elementary operation, and every period of free evolution, is itself noisy. One way to approach this is to use QEC repeatedly, but with the syndrome extraction procedure carefully constructed in such away that it corrects more noise than it introduces. Most of the essential new insights which permit us to do this were introduced by Shor [11] and helpfully discussed by Preskill [12]; see also [16, 18, 15]. Here we will adopt Shor's general approach, but with significant improvements introduced by Steane [13, 14]. Note that this subject is much less mature than QEC; many avenues remain unexplored. Here we will concentrate on explaining one method to extract syndromes in the right way.

A complete fault-tolerant syndrome extraction network is shown in figure 1. For brevity, we consider the simplest case of a single-error correcting code; the ideas can be generalised to codes correcting many errors. The fundamental 2-state entities in the computer are called physical qubits. Each horizontal line in the network represents not a single physical qubit, but a block of n such qubits. Operators such as Hadamard and CNOT are applied across the relevent block or blocks, i.e. n operations, one for each qubit or pair of qubits.

The method relies on the careful use of repetition, on the fact that X and Z errors propagate differently, and on useful properties of CSS codes. Define an *error location* to be any 1 or 2-qubit gate on physical qubits (including preparation and measurement operations), or the free evolution of any single physical qubit during one timestep. The noise is assumed to be uncorrelated and stochastic, so that failures occur independently with probability $\sim \gamma$. The aim of the whole network is to achieve a single-error correction of the computer block, in such a way that no failure at a single location can result in an error of weight ≥ 2 in the computer block. The idea is that while the syndrome extraction must make single-qubit errors in the computer more likely, these are the very ones which are correctable. The important thing is not to generate uncorrectable errors with $O(\gamma)$ probability.

We begin by introducing 2 ancilla blocks, and preparing each in the logical zero state $|0\rangle_L$. Each preparation is not fault tolerant, it will fail in such a way that the prepared

state can have any error of any weight with probability $O(\gamma)$. Operate CNOT blockwise between the two ancillas, and measure all the bits of one of them in the computational basis. Here we are trying to verify that the correct state was prepared, using the fact that blockwise physical CNOT acts as a logical CNOT for a CSS code. Therefore, the measurement result should be a member of the classical code $C_2^\perp$ (eq. (17)). If it is not, then reprepare the pair of ancillas and repeat until it is. At this stage, the probability for the remaining unmeasured ancilla to have X errors of weight ≥ 2 is $O(\gamma^2)$, because it can only happen if failures occur in at least two locations. Note that the ancilla might still have Z errors of any weight.

Now couple the verified ancilla to the computer by blockwise physical CNOT. Once again, we use the fact that this acts as logical CNOT, so there should be no effect! In fact something does happen: X errors propagate from ancilla to computer, and Z errors propagate from computer to ancilla. This is a sneaky, and fault tolerant, way to gather the Z-error syndrome into the ancilla. We read it out by Hadamard transforming the ancilla (to convert Z errors to bit flips) and measuring all the bits of the ancilla in the computational basis. Here we have used the property, valid for a certain class of CSS codes, that blockwise physical H acts as logical H, so will keep the ancilla state in the encoded subspace, except for the Z errors which become X errors.

There is still no single error location which can produce a weight-2 error in the computer, but now we are in danger, since there are many locations where a single failure would lead to an incorrect syndrome. If we were to 'correct' the computer on the basis of the wrong syndrome, we would actually introduce more errors. Therefore, the whole of the process described up till now is repeated. We finally end up with two syndromes. If they agree, then the only way they can be wrong is if failures occured at two different locations, an $O(\gamma^2)$ process, so we go ahead and believe them. If they disagree, a third syndrome must be extracted, and we act on the majority vote.

We have now completed the correction of Z errors in the computer (while generating further Z errors, which will be caught in the next round of correction). The second half of the network acts similarly, but now gathers up and corrects the X errors in the computer.

Note that the whole process depends on the fact that X and Z errors propagate differently. We can fault-tolerantly verify the ancilla against X errors, but only by accepting the chance to have high-weight Z errors in the ancilla. This is OK because those Z errors stay put, they don't propagate up to the computer, they just make the syndrome wrong. We subsequently check for their presence by generating the syndrome again. Note also the heavy reliance on useful properties of CSS codes, such as their behaviour under blockwise gates.

In a repeated series of error recoveries, each round of recovery corrects not just the errors developed in the computer during that round, but also the errors caused by the previous round (as long as they are correctable). It leaves uncorrected the errors it itself caused. The noise level accumulated after R rounds is therefore suppressed from $O(R\gamma)$ to $O(R\gamma^2 + \gamma)$, which is beneficial for large R and sufficiently small γ.

To complete the task of fault tolerant *computation*, not just memory storage, we need to be able to evolve the computer state through the desired quantum algorithm. We already saw how to perform logical Hadamard and CNOT operations on the state encoded by a CSS code: operate blockwise on the qubits. This is fault tolerant since each physical gate only connects to one physical qubit per block. To obtain a complete set of operations, we use the fact that the members of the continuous set of all gates can be approximated efficiently by using members of a discrete set. To complete the set, it is sufficient to have a fault-tolerant Toffoli gate, or one of a set of closely related

gates, among which is the controlled-$\pi/2$ rotation. Shor [11] proposed a (somewhat obscure) network for Toffoli. It is possible to understand the construction as related to teleportation. Teleportation can be understood as a form of fault tolerant swap operation, and it is useful for moving information around fault-tolerantly in a quantum computer[17, 14]. These and other methods are under active investigation.

At the time of writing, fault tolerant computation based on repeated QEC seems to be the most promising way to realise large quantum algorithms, though the requirements on the physical hardware, both in terms of computer size and noise level, remain formidable.

References

[1] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev. A*, vol. 52, pp. R2493-R2496, Oct. 1995.

[2] A. M. Steane, "Error correcting codes in quantum theory," *Phys. Rev. Lett.*, vol. 77, pp. 793-767, July 1996.

[3] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Phys. Rev. A*, vol. 54, pp. 1098-1105, Aug. 1996.

[4] A. M. Steane, "Multiple particle interference and quantum error correction," *Proc. Roy. Soc. Lond. A*, vol. 452, pp. 2551-2577, Nov. 1996.

[5] A. R. Calderbank, E. M. Rains, P. W. Shor and N. J. A. Sloane, "Quantum error correction via codes over $GF(4)$", *IEEE Trans. Information Theory*, vol. 44, pp1369-1387, July 1998.

[6] A. R. Calderbank, E. M. Rains, N. J. A. Sloane and P. W. Shor, "Quantum error correction and orthogonal geometry," *Phys. Rev. Lett.*, vol. 78, pp. 405-409, (1997).

[7] E. Knill and R. Laflamme, "A theory of quantum error correcting codes," *Phys. Rev. A*, vol. 55, pp. 900-911, (1997).

[8] E. Knill and R. Laflamme, "Concatenated quantum codes," LANL eprint quant-ph/9608012.

[9] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin and W. K. Wootters, "Mixed state entanglement and quantum error correction," *Phys. Rev. A*, vol. 54, pp. 3822-3851, (1996).

[10] D. Gottesman, "Class of quantum error-correcting codes saturating the quantum Hamming bound," *Phys. Rev. A*, vol. 54, pp. 1862-1868, (1996).

[11] P. W. Shor, "Fault-tolerant quantum computation," in *Proc. 37th Symp. on Foundations of Computer Science*, (Los Alamitos, CA: IEEE Computer Society Press), pp15-65 (1996).

[12] J. Preskill, "Reliable quantum computers," *Proc. Roy. Soc. Lond. A* **454**, 385 (1998).

[13] A. M. Steane "Active stabilisation, quantum computation and quantum state synthesis," Phys. Rev. Lett. **78**, 2251-2577 (1997).

[14] A. M. Steane, "Efficient fault-tolerant quantum computing," Nature, vol. **399**, 124-126 (May 1999). (LANL eprint quant-ph/9809054).

[15] E. Knill, R. Laflamme and W. H. Zurek, "Resilient quantum computation: Error Models and Thresholds," *Proc. Roy. Soc. Lond A* **454**, 365-384 (1998); *Science* **279**, 342-345 (1998). (LANL eprint quant-ph/9702058)

[16] A. M. Steane, "Space, time, parallelism and noise requirements for reliable quantum computing," *Fortschr. Phys.* **46**, 443-457 (1998). (LANL eprint quant-ph/9708021).

[17] D. Gottesman, "A theory of fault-tolerant quantum computation," *Phys. Rev. A* **57**, 127 (1998). (LANL eprint quant-ph/9702029)

[18] D. Aharonov and M. Ben-Or, "Fault-Tolerant Quantum Computation With Constant Error Rate" LANL eprint quant-ph/9906129.