Proceedings of the International School of Physics "Enrico Fermi", course CLXII, "Quantum Computers, Algorithms and Chaos", G. Casati, D. L. Shepelyansky and P. Zoller, eds., pp. 1–32 (IOS Press, Amsterdam 2006).

A Tutorial on Quantum Error Correction

Andrew M. Steane

Centre for Quantum Computation,
Department of Physics, University of Oxford,
Clarendon Laboratory, Parks Road, Oxford OX1 3PU, England.

1. Introduction

Quantum error correction (QEC) comes from the marriage of quantum mechanics with the classical theory of error correcting codes. Error correction is a central concept in classical information theory, and quantum error correction is similarly foundational in quantum information theory. Both are concerned with the fundamental problem of communication, and/or information storage, in the presence of noise. The codewords which arise in QEC are also interesting objects in their own right, displaying rich forms of entanglement.

This introduction will concentrate on the essential ideas of QEC and on the construction and use of quantum error correcting codes. The motivation is that although QEC is now quite a well established subject, it remains only vaguely understood by many people interested in quantum information and computing, or little studied by students learning it, because it appears to require mathematical knowledge they do not possess, and to be limited in application. Introductions tend to stay at the level of single-error-correcting codes, and single-qubit encoding codes, but this is a mistake because some of the essential properties only emerge when multiple errors and encoding of multiple qubits are considered. The mathematical methods are elegant and fairly easy to acquire, they are probably essential tools for anyone wishing to think of good ways to manipulate multi-qubit systems.

An important area only briefly touched on in this tutorial is the physics of noise, and the fidelity that can be achieved via QEC methods. This is important because it underlies much of what we think can actually be achieved by quantum information processing in the future. A further important subject closely connected to QEC is that of fault-tolerant methods. They are not discussed here (for introductory discussions see [1, 2, 3, 4]).

Error correction is especially important in quantum computers, because efficient quantum algorithms make use of large scale quantum interference, which is fragile, i.e. sensitive to imprecision in the computer and to unwanted coupling between the computer and the rest of the world [5, 6, 8, 7]. This makes large scale quantum computation so difficult as to be practically impossible unless error correction methods are used [1, 2, 9]. Indeed, before quantum error correction was discovered, it seemed that this fragility would forbid large scale quantum computation. The ability to correct a quantum computer efficiently without disturbing the coherence of the computation is highly non-intuitive, and was thought more or less impossible. The discovery of powerful error correction methods therefore caused much excitement, since it converted large scale quantum computation from a practical impossibility to a possibility.

The first quantum error correcting codes were discovered independently by Shor [10] and Steane [11]. Shor proved that 9 qubits could be used to protect a single qubit against general errors, while Steane described a general code construction whose simplest example does the same job using 7 qubits (see section 7.2). A general theory of quantum error correction dates from subsequent papers of Calderbank and Shor [12] and Steane [13] in which general code constructions, existence proofs, and correction methods were given. Knill and Laflamme [14] and Bennett *et. al.* [15] provided a more general theoretical framework, describing requirements for quantum error correcting codes (section 5.3), and measures of the fidelity of corrected states. They, and independently Ekert and Macchiavello [16] and Gottesman [17] derived the quantum Hamming bound (section 5.4). The close relationship between entanglement purification [19, 18] and quantum error correction was discussed by Bennett *et. al.* [15]. They, and independently Laflamme *et. al.* [20] discovered the perfect 5 qubit code (section 5.2).

The important concept of the stabilizer (section 5.2) is due to Gottesman [17] and independently Calderbank *et. al.* [21]; this yielded many useful insights into the subject, and permitted many new codes to be discovered [17, 21, 22, 23, 24]. Stabilizer methods will probably make a valuable contribution to other areas in quantum information physics. The quantum MacWilliams identities were discovered by Shor and Laflamme [25], these provide further important constraints on codes and fidelity measures. The idea of recursively encoding and encoding again was explored by Knill and Laflamme [26], and Aharonov and Ben-Or [27]. This uses more quantum resources in a hierarchical way, to permit communication over arbitrarily long times or distances. Van Enk *et. al.* [28] have discussed quantum communication over noisy channels using a realistic model of trapped atoms and high-quality optical cavities, and recursive techniques for

systems in which two-way classical communication is possible have been discussed. Since these early works, quantum code constructions have been discussed by many authors, and general questions about the information capacity of noisy channels have been settled, see [29] for a review.

The organisation of the article is as follows. Section 2 is very basic, it introduces the 3-bit code for the absolute beginner. Sections 3 and 4 cover classical coding theory, linear vector spaces and so on. They can be omitted by readers familiar with classical error correction, but are needed by those not familiar with it in order to progress beyond the most simple quantum codes later on. Sections 5 and 6 contain the main meat of the article. They introduce the basic concepts of noise and errors, the mathematics of error operators and stabilizers, and the elements of quantum error correction coding (the construction and behaviour of codes). Section 7 then goes into the code construction and the error correction process in more detail, for the benefit of students wishing to acquire as much physical insight as possible; the other sections can stand alone without it. Section 8 is very brief and incomplete but nevertheless necessary to complete the foundations: it discusses some aspects of the physics of noise which we need in order to understand to what extent QEC will be successful in practice.

2. Three bit code

We will begin by analysing in detail the workings of the most simple quantum error correcting code. Exactly what is meant by a quantum error correcting code will become apparent.

Suppose a source A wishes transmit quantum information via a noisy communication channel to a receiver B. Obviously the channel must be noisy in practice since no channel is perfectly noise-free. However, in order to do better than merely sending quantum bits down the channel, we must know something about the noise. For this introductory section, the following properties will be assumed: the noise acts on each qubit independently, and for a given qubit has an effect chosen at random between leaving the qubit's state unchanged (probability $1-p$) and applying a Pauli σ_x operator (probability $p < 1/2$). This is a very artificial type of noise, but once we can correct it, we will find that our correction can also offer useful results for much more realistic types of noise.

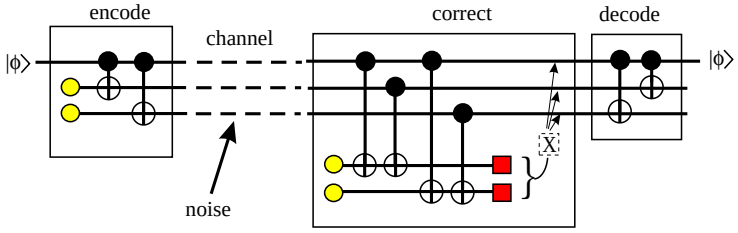

Figure 1. Simple example illustrating the principles of quantum error correction. Alice wishes to transmit a single-qubit state $|\phi\rangle = a|0\rangle + b|1\rangle$ to Bob through a channel which introduces σ_x errors ($|0\rangle \leftrightarrow |1\rangle$) randomly. Alice prepares two further qubits in the state $|0\rangle$, represented by a small circle. She then *encodes* her single qubit into a joint state of three qubits, by two controlled-NOT operations. These three qubits are sent to Bob. At the receiving end, Bob *recovers* the joint state by extracting a syndrome, and correcting on the basis of this syndrome. The correction is a σ_x operation applied to one (or none) of the qubits. Finally, a *decoding* operation disentangles one qubit from the others, giving Bob a single qubit in the state $|\phi\rangle$ with probability $1 - O(p^2)$.

The simplest quantum error correction method is summarised in fig. 1. We adopt the convention of calling the source Alice and the receiver Bob. The state of any qubit that Alice wishes to transmit can be written without loss of generality $a|0\rangle + b|1\rangle$. Alice prepares two further qubits in the state $|0\rangle$, so the initial state of all three is $a|000\rangle + b|100\rangle$. Alice now operates a controlled-NOT gate from the first qubit to the second, producing $a|000\rangle + b|110\rangle$, followed by a controlled-NOT gate from the first qubit to the third, producing $a|000\rangle + b|111\rangle$. Finally, Alice sends all three qubits down the channel.

Bob receives the three qubits, but they have been acted on by the noise in the channel. Their state is

one of the following:

state	probability
$a\|000\rangle + b\|111\rangle$	$(1-p)^3$
$a\|100\rangle + b\|011\rangle$	$p(1-p)^2$
$a\|010\rangle + b\|101\rangle$	$p(1-p)^2$
$a\|001\rangle + b\|110\rangle$	$p(1-p)^2$
$a\|110\rangle + b\|001\rangle$	$p^2(1-p)$
$a\|101\rangle + b\|010\rangle$	$p^2(1-p)$
$a\|011\rangle + b\|100\rangle$	$p^2(1-p)$
$a\|111\rangle + b\|000\rangle$	p^3

(1)

Bob now introduces two more qubits of his own, prepared in the state $|00\rangle$. This extra pair of qubits, referred to as an *ancilla*, is not strictly necessary, but makes error correction easier to understand and becomes necessary when fault-tolerant methods are needed. Bob uses the ancilla to gather information about the noise. He first carries out controlled-NOTs from the first and second received qubits to the first ancilla qubit, then from the first and third received qubits to the second ancilla bit. The total state of all five qubits is now

state	probability
$(a\|000\rangle + b\|111\rangle)\|00\rangle$	$(1-p)^3$
$(a\|100\rangle + b\|011\rangle)\|11\rangle$	$p(1-p)^2$
$(a\|010\rangle + b\|101\rangle)\|10\rangle$	$p(1-p)^2$
$(a\|001\rangle + b\|110\rangle)\|01\rangle$	$p(1-p)^2$
$(a\|110\rangle + b\|001\rangle)\|01\rangle$	$p^2(1-p)$
$(a\|101\rangle + b\|010\rangle)\|10\rangle$	$p^2(1-p)$
$(a\|011\rangle + b\|100\rangle)\|11\rangle$	$p^2(1-p)$
$(a\|111\rangle + b\|000\rangle)\|00\rangle$	p^3

(2)

Bob measures the two ancilla bits in the basis $\{|0\rangle, |1\rangle\}$. This gives him two classical bits of information. This information is called the *error syndrome*, since it helps to diagnose the errors in the received qubits. Bob's next action is as follows:

measured syndrome	action
00	do nothing
01	apply σ_x to third qubit
10	apply σ_x to second qubit
11	apply σ_x to first qubit

Suppose for example that Bob's measurements give 10 (i.e. the ancilla state is projected onto $|10\rangle$). Examining eq. (2), we see that the state of the received qubits must be either $a|010\rangle + b|101\rangle$ (probability $p(1-p)^2$) or $a|101\rangle + b|010\rangle$ (probability $p^2(1-p)$). Since the former is more likely, Bob corrects the state by applying a Pauli σ_x operator to the second qubit. He thus obtains either $a|000\rangle + b|111\rangle$ (most likely) or $a|111\rangle + b|000\rangle$. Finally, to extract the qubit which Alice sent, Bob applies controlled-NOT from the first qubit to the second and third, obtaining either $(a|0\rangle + b|1\rangle)|00\rangle$ or $(a|1\rangle + b|0\rangle)|00\rangle$. Therefore Bob has either the exact qubit sent by Alice, or Alice's qubit operated on by σ_x. Bob does not know which he has, but the important point is that the method has a probability of success greater than $1-p$. The correction is designed to succeed whenever either no or just one qubit is corrupted by the channel, which are the most likely possibilities. The failure probability is the probability that at least two qubits are corrupted by the channel, which is $3p^2(1-p) + p^3 = 3p^2 - 2p^3$, i.e. less than p (as long as $p < 1/2$).

To summarise, Alice communicates a single general qubit by expressing its state as a joint state of three qubits, which are then sent to Bob. Bob first applies error correction, then extracts a single qubit state. The probability that he fails to obtain Alice's original state is $O(p^2)$, whereas it would have been $O(p)$ if no error correction method had been used. We will see later that with more qubits the same basic ideas lead to much more powerful noise suppression, but it is worth noting that we already have quite an impressive result: by using just three times as many qubits, we reduce the error probability by a factor $\sim 1/3p$, i.e. a factor $\sim$30 for $p = 0.01$, $\sim$300 for $p = 0.001$, and so on.

3. Binary fields and discrete vector spaces

In order to generalise the above ideas, we will need to understand the theory of classical error correcting codes, and this section provides some mathematical preliminaries.

Classical error correction is concerned with classical bits, not quantum states. The mathematical treatment is based on the fact that linear algebraic operations such as addition and multiplication can be consistently defined using finite rather than infinite sets of integers, by using modular arithmetic. The simplest case, that of modulo 2 arithmetic, will cover almost everything in this article. The addition operation is defined by $0+0=0, 0+1=1+0=1, 1+1=0$. The set $\{0,1\}$ is a group under this operation, since 0 is the identity element, both elements are their own inverse and the operation is associative. The set $\{0,1\}$ is also a group under multiplication, with identity element 1. Furthermore, we can also define division (except division by zero) and subtraction, and the commutative and distributive laws hold. These properties together define a finite *field*, also called a Galois field. Thus the set $\{0,1\}$ is referred to as the field GF(2), where addition and multiplication are as defined.

A string of n bits is considered to be a vector of n components, for example 011 is the vector $(0,1,1)$. Vector addition is carried out by the standard method of adding components, for example $(0,1,1)+(1,0,1)=(0+1,1+0,1+1)=(1,1,0)$. It is easy to see that this operation is equivalent to the exclusive-or operation $\oplus$ carried out bitwise between the binary strings: $011 \oplus 101 = 110$. Note that $u+u \equiv 0$ and $u-v=u+v$ (prove by adding v to both sides).

We can define the inner product (or scalar product) by the standard rule of multiplying corresponding components, and summing the results: $(1,1,0,1)\cdot(1,0,0,1) = 1+0+0+1 = 0$. Note that all the arithmatic is done by the rules of the Galois field, so the final answer is only ever 0 or 1. The inner product is also called a *parity check* or *check sum* since it indicates whether the second vector 'satisfies' the parity check specified by the first vector (or equivalently whether the first vector satisfies the parity check specified by the second). To satisfy a parity check u, a vector v must have an even number of 1's at the positions (coordinates) specified by the 1's in u. If u and v are row vectors then $u\cdot v = uv^T$ where T is the transpose operation.

The number of non-zero components of a binary vector u is important in what follows, and is called the *weight* (or Hamming weight), written $\mathrm{wt}(u)$. For example, $\mathrm{wt}(0001101)=3$. The number of places (coordinates) where two vectors differ is called the *Hamming distance* between the vectors; the distance between u and v is equal to $\mathrm{wt}(u+v)$.

There are 2^n vectors of n components (stated in other language, there are 2^n n-bit words). This set of vectors forms a linear vector space, sometimes called Hamming space. It is a discrete vector space since vector components are are only ever equal to 0 or 1. The vectors point to the vertices of a square lattice in n dimensions. The space is spanned by any set of n linearly independent vectors. The most obvious set which spans the space is $\{1000\cdots00,\ 0100\cdots00,\ 0010\cdots00,\ \ldots,\ 0000\cdots01\}$.

There are subspaces within Hamming space. A *linear* subspace C is any set of vectors which is closed under addition, ie $u+v \in C\ \forall\ u,v \in C$. For example the set 0000,0011,1100,1111 is a 2^2 linear subspace of the 2^4 Hamming space. A linear subspace containing 2^k vectors is spanned by k linearly independent vectors (for example 0011 and 1100 in the case just given). Any linear subspace is thus completely specified by its *generator matrix* G, which is just the matrix whose k rows are any k vectors which span the space. We can always linearly combine rows to get an equivalent generator matrix, for example

$$G = \begin{pmatrix} 0011 \\ 1100 \end{pmatrix} = \begin{pmatrix} 0011 \\ 1111 \end{pmatrix} \tag{3}$$

The *minimum distance* d of a subspace is the smallest Hamming distance between any two members of the subspace. If the two closest vectors are u and v, then $d=\mathrm{wt}(u+v)$. For the case of a linear space, $w=u+v$ is also a member of the space. From this we deduce that the minimum distance of a linear space is equal to the smallest weight of a non-zero member of the space. This fact is useful in calculating

the value of d, since it is much easier to find the minimum weight than to evaluate all the distances in the space.

Now, if $u \cdot v = 0$ and $u \cdot w = 0$ then $u \cdot (v + w) = 0$. From this it follows, that if all the rows of a generator satisfy the parity check u, then all so do all the vectors in the subspace. Any given parity check u divides Hamming space exactly in half, into those vectors which satisfy u and those that do not. Therefore, the 2^k vectors of a linear subspace in 2^n Hamming space can satisfy at most $n-k$ linearly independent parity checks. These parity checks together form the *parity check matrix* H, which is another way to define the linear subspace. H has n columns and $n-k$ rows. For any given subspace, the check and generator matrices are related by

$$HG^T = \mathbf{0} \tag{4}$$

where G^T is the transpose of G, and $\mathbf{0}$ is the $(n-k) \times k$ zero matrix.

The simple error correction method described in the previous section is based around the very simple binary vector space $000, 111$. Its generator matrix is $G = (111)$ and the parity check matrix is

$$H = \begin{pmatrix} 110 \\ 101 \end{pmatrix} \tag{5}$$

A useful relationship enables us to derive each of H and G from the other. It is always possible to convert G to the form $G = (I_k, A)$ where I_k is the $k \times k$ identity matrix, and A is the rest of G (so A is $k \times n-k$). To do the conversion we can linearly combine rows of G, and if necessary swap columns of G. Once we have the right form for G, then H can be written down immediately, it is $H = (A^T, I_{n-k})$.

The last concept which we will need in what follows is that of the *dual*. The dual space $C^\perp$ is the set of all vectors u which have zero inner product with all vectors in C, $u \cdot v = 0 \; \forall v \in C$. It is simple to deduce that the parity check matrix of C is the generator matrix of $C^\perp$ and *vice versa*. If $H = G$ then $C = C^\perp$, such spaces are termed self-dual.

The notation (n, m, d) is a shorthand for a set of m n-bit vectors having minimum distance d. For linear vector spaces, the notation $[n, k, d]$ is used, where k is now the dimension of the vector space, so it contains 2^k vectors.

Let us conclude this section with another example of a linear binary vector space which will be important in what follows. It is a $[7, 4, 3]$ space discovered by Hamming [30]. The generator matrix is

$$G = \begin{pmatrix} 1010101 \\ 0110011 \\ 0001111 \\ 1110000 \end{pmatrix} \tag{6}$$

so the sixteen members of the space are

$$\begin{array}{cccc} 0000000 & 1010101 & 0110011 & 1100110 \\ 0001111 & 1011010 & 0111100 & 1101001 \\ 1110000 & 0100101 & 1000011 & 0010110 \\ 1111111 & 0101010 & 1001100 & 0011001 \end{array} \tag{7}$$

these have been written in the following order: first the zero vector, then the first row of G. Next add the second row of G to the two vectors so far obtained, then add the third row to the four vectors previously obtained, and so on. We can see at a glance that the minimum distance is 3 since the minimum non-zero weight is 3. The parity check matrix is

$$H = \begin{pmatrix} 1010101 \\ 0110011 \\ 0001111 \end{pmatrix}. \tag{8}$$

It is simple to confirm that $HG^T = \mathbf{0}$. Note also that since H is made of rows of G, this code contains its dual: $C^\perp \in C$.

4. Classical error correction

Classical error correction is a large subject, a full introduction may be found in many readily available textbooks [31, 32, 33, 34]. In order to keep the present discussion reasonably self-contained, a minimal set of ideas is given here. These will be sufficient to guide us in the construction of quantum error correcting codes.

Classical communication can be considered without loss of generality to consist in the communication of strings of binary digits, i.e. the binary vectors introduced in the previous section. A given binary vector, also called a binary *word*, which we wish to send from A to B, is called a *message*. A noisy communication channel will corrupt the message, but since the message u is a binary vector the only effect the noise can have is to change it to some other binary vector u'. The difference $e = u' - u$ is called the *error vector*. Error correction consists in deducing u from u'.

4.1. Error correcting code

A classical *error correcting code* is a set of words, that is, a binary vector space. It need not necessarily be linear, though we will be concerned almost exclusively with linear codes. Each error correcting code C allows correction of a certain set $\mathcal{S} \equiv \{e_i\}$ of error vectors. The correctable errors are those which satisfy

$$u + e_i \neq v + e_j \quad \forall u, v \in C \ (u \neq v) \tag{9}$$

The case of no error, $e = 0$, is included in $\mathcal{S}$, so that error-free messages are 'correctable'. To achieve error correction, we use the fact that each message u is corrupted to $u' = u + e$ for some $e \in \mathcal{S}$. However, the receiver can deduce u unambiguously from $u+e$ since by condition (9), no other message v could have been corrupted to $u + e$, as long as the channel only generates correctable error vectors. In practice a noisy channel causes both correctable and uncorrectable errors, and the problem is to match the code to the channel, so that the errors *most likely* to be generated by the channel are those the code can correct.

Let us consider two simple examples. First, suppose the channel is highly noisy, but noise occurs in bursts, always affecting pairs of bits rather than one bit at a time. In this case we use the simple code $C = \{00, 01\}$. Longer messages are sent bit by bit using the code, by sending 00 for 0 and 01 for 1. The possible error vectors (those which the channel can produce) are $\{00, 11\}$, and in this example this is also a set of correctable errors: the receiver interprets 00 or 11 as the message 00, and 01 or 10 as the message 01. Therefore error correction always works perfectly! This illustrates the fact that we can always take advantage of structure in the noise (here, pairs of bits being equally affected) in order to circumvent it. In the quantum case, the corresponding situation is called a "noise-free (or decoherence-free) subspace" [3], where parts of the state space of the quantum system are unaffected by environmental coupling, so they can be used to store information which will not be influenced by the noise.

Next suppose the noise affects each bit independently, with a fixed error probability $p < 1/2$. This noise has less structure than that we just considered, but it still has some predictable features, the most important being that the most likely error vectors are those with the smallest weight. We use the code $C = \{000, 111\}$. The errors which the channel can produce are, in order of decreasing probability, $\{000, 001, 010, 100, 011, 101, 011, 111\}$. With $n = 3$ and $m = 2$, the set of correctable errors can have at most $2^3/2 = 4$ members, and these are $\{000, 001, 010, 100\}$. This is the classical equivalent of the quantum code described in section 2.

4.2. Minimum distance coding

The noisy channel just described is called the *binary symmetric channel*. This is a binary channel (the only kind we are considering) in which the noise affects each bit sent down the channel independently. It is furthermore symmetric, meaning that the channel causes errors $0 \to 1$ and $1 \to 0$ with equal probability.

If n bits are sent down a binary symmetric channel, the probability that m of them are flipped ($0 \leftrightarrow 1$) is the probability for m independent events in n opportunities:

$$C(n,m)p^m(1-p)^{n-m} \tag{10}$$

where the binomial coefficient $C(n,m) = n!/(m!(n-m)!)$.

The binary symmetric channel is important because other types of noise can be treated as a combination of a structured component and a random component. The structured component can be tackled in more efficient ways, for example burst-error correcting codes, and then the remaining random component forms a problem equivalent to that of the binary symmetric channel.

To code for the binary symmetric channel, we clearly need a code in which error vectors of small weight are correctable, since these are the most likely ones. A code which corrects all error vectors of weight up to and including t is called a t-error correcting code. A simple but important observation is the following:

A code of minimum distance d can correct all error vectors of weight less than or equal to t if and only if $d > 2t$.

Proof: if $d > 2t$ then $\mathrm{wt}(u+v) > 2t$ for all u, v in the code, and therefore $\mathrm{wt}(u+v+e_1+e_2) \neq 0$ for all error vectors e_1, e_2 of weight $\leq t$. This implies $u + e_1 \neq v + e_2$ so condition (9) is satisfied. Also, if $d = \mathrm{wt}(u+v) \leq 2t$ then there exist error vectors e_1, e_2 of weight $\leq t$ such that $\mathrm{wt}(u+v+e_1+e_2) = 0$, which implies $u + e_1 = v + e_2$, so correction is impossible.

This argument shows that a good set of codes for the binary symmetric channel are those of high minimum distance.

4.3. Bounds on the size of codes

In order to communicate k bits of information using an error correcting code, $n > k$ bits must be sent down the channel. The ratio k/n, called the *rate* of the code, gives a measure of the cost of error correction. Various bounds on k/n can be deduced.

In a Hamming space of n dimensions, that is, one consisting of n-bit vectors, there are $C(n,t)$ error vectors of weight t. Any member of a t-error correcting code has $\sum_{i=0}^{t} C(n,i)$ erroneous versions (including the error-free version) which must all be distinct from the other members of the code and their erroneous versions if correction is to be possible. They are also distinct from each other because

$$e_1 \neq e_2 \Rightarrow u + e_1 \neq u + e_2. \tag{11}$$

However, there are only 2^n possible vectors in the whole Hamming space, therefore the number of vectors m in n-bit t-error correcting codes is limited by the *Hamming bound* [30, 31]

$$m \sum_{i=0}^{t} C(n,i) \leq 2^n \tag{12}$$

This is also called the *sphere packing bound* because the problem is equivalent to that of packing as many spheres as possible in the n-dimensional rectangular space, where each sphere has radius t. For linear codes $m = 2^k$ so the Hamming bound becomes

$$k \leq n - \log_2 \left(\sum_{i=0}^{t} C(n,i) \right). \tag{13}$$

From this one may deduce in the limit of large n, k, t:

$$\frac{k}{n} \leq \left(1 - H\left(\frac{t}{n}\right)\right)(1-\eta) \tag{14}$$

where $\eta \to 0$ as $n \to \infty$, and $H(x)$ is the entropy function

$$H(x) \equiv -x \log_2 x - (1-x)\log_2(1-x). \tag{15}$$

The Hamming bound makes precise the intuitively obvious fact that error correcting codes can not achieve arbitrarily high rate k/n while still retaining their correction ability. As yet we have no definite guide as to whether good codes exist in general. A very useful result is the Gilbert-Varshamov bound: it can be shown [32] that for given n, d, there exists a linear $[n, k, d]$ code provided

$$2^k \sum_{i=0}^{d-2} C(n-1, i) < 2^n \tag{16}$$

In the limit of large n, k, d, and putting $t = d/2$, this becomes

$$\frac{k}{n} \geq \left(1 - H\left(\frac{2t}{n}\right)\right)(1-\eta) \tag{17}$$

where $\eta \to 0$ as $n \to \infty$. It can be shown that there exists an infinite sequence of $[n, k, d]$ codes satisfying (17) with $d/n \geq \delta$ if $0 \leq \delta < 1/2$. The Gilbert-Varshamov bound necessarily lies below the Hamming bound, but it is an important result because it shows that error correction can be very powerful. In the binary symmetric channel, for large n the probability distribution of all the possible error vectors is strongly peaked around error vectors of weight close to the mean np (see eq. (10)), where p is the error probability per bit. This is an example of the law of large numbers. Therefore as long as $t > np(1+\eta)$, where $\eta << 1$, error correction is almost certain to succeed in the limit $n \to \infty$. The Gilbert-Varshamov bound tells us that this can be achieved without the need for codes of vanishingly small rate.

Another result on coding limitations is Shannon's theorem [35, 31, 32], which states that codes exist whose average performance is close to that of the Hamming bound:

Shannon's theorem: If the rate k/n is less than the channel capacity, and n is sufficiently large, then there exists a binary code allowing transmission with arbitrarily low failure probability.

The failure probability is the probability that an uncorrectable error will occur; the capacity of the binary symmetric channel is $1 - H(p)$. Shannon's theorem is 'close to' the Hamming bound in the sense that we would expect a correction ability $t > np$ to be required to ensure success in a binary symmetric channel, which implies $k/n < 1 - H(p)$ in the Hamming bound; Shannon's theorem assures us that codes exist which get arbitrarily close to this. The codes whose existence is implied by Shannon's theorem are not necessarily convenient to use, however.

4.4. Linear codes, error syndrome

The importance of linear codes is chiefly in their convenience, especially the speed with which any erroneous received word can be corrected. They are also highly significant when we come to generalise classical error correction to quantum error correction.

So far the only error correction method we have mentioned is the simple idea of a look-up table, in which a received vector w is compared with all the code vectors $u \in C$ and their erroneous versions $u + e$, $e \in \mathcal{S}$, until a match $w = u + e$ is found, in which case the vector is corrected to u. This method makes inefficient use of either time or memory resources since there are 2^n vectors $u + e$.

For linear codes, a great improvement is to calculate the *error syndrome* s given by

$$s = Hw^T. \tag{18}$$

where H is the parity check matrix. Since H is a $(n-k) \times n$ matrix, and w is an n bit row vector, the syndrome s is an $n-k$ bit column vector. The transpose of w is needed to allow the normal rules of matrix multiplication, though the notation $H \cdot w$ is sometimes also used. Consider

$$s = H(u+e)^T = Hu^T + He^T = He^T \tag{19}$$

where we used eq. (4) for the second step. This shows that *the syndrome depends only on the error vector, not on the transmitted word.* If we could deduce the error from the syndrome, which will be shown next, then we only have 2^{n-k} syndromes to look up, instead of 2^n erroneous words. Furthermore, many codes

can be constructed in such a way that the error vector can be deduced from the syndrome by analytical methods, such as the solution of a set of simultaneous equations.

Proof that the error can be deduced from the syndrome. The parity check matrix consists of $n-k$ linearly independent parity check vectors. Each check vector divides Hamming space in half, into those vectors which satisfy the check, and those which do not. Hence, there are exactly 2^k vectors in Hamming space which have any given syndrome s. Using eq. (19), these vectors must be the vectors $u+e_s$ where $u \in C$ is one of the 2^k members of the code C. Hence the only error vectors which give the syndrome s are the members of the set $\{e_s+u\}, u \in C$. Such a set is called a coset. The syndrome cannot distinguish among errors in a given coset. In choosing which errors will be correctable by the code, we can choose at most one from each coset. Then, we have proved that each correctable error is uniquely determined by its syndrome.

To conclude, let us consider an example using the $[7,4,3]$ Hamming code given at the end of section 3. Since this code has minimum distance 3, it is a single error correcting code. The number of code vectors is limited by the Hamming bound to $2^7/(C(7,0)+C(7,1)) = 2^7/(1+7) = 2^4$. Since there are indeed 2^4 vectors the code saturates the bound; such codes are called *perfect*. The set of correctable errors is $\{0000000, 0000001, 0000010, 0000100, 0001000, 0010000, 0100000, 1000000\}$. Suppose the message is 0110011 and the error is 0100000. The received word is $0110011 + 0100000 = 0010011$. The syndrome, using eq. (8), is

$$H(0010011)^T = (010)^T \tag{20}$$

The only word of weight ≤ 1 which fails the second parity check and passes the others is 0100000, so this is the deduced error. We thus deduce the message to be $0010011 - 0100000 = 0110011$.

5. Quantum error correction

The introductory example of quantum error correction (QEC) that was given in section 2 was in fact based on the most simple classical error correcting code, the $[3,1,3]$ repetition code, whose parity check matrix has two rows 110 and 011. It illustrates the basic idea of using parity checks to acquire an error syndrome in the quantum case, but it cannot correct the more general type of noise which can affect qubits. In order to generalise to full quantum error correction, we will need to introduce further concepts. The simplest aspects of this more general theory will be summarised in this section.

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.

5.1. 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 in the form:

$$|\phi\rangle\,|\psi_0\rangle_e \rightarrow \sum_i (E_i\,|\phi\rangle)\,|\psi_i\rangle_e \tag{21}$$

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 $|\psi_i\rangle_e$ 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 $X \equiv \sigma_x$, $Z \equiv \sigma_z$, $Y \equiv -i\sigma_y = XZ$.

It is worth mulling over equation (21). The statement is true because the Pauli matrices are a complete set: they can describe any transformation of the qubits, and because we allow any sort of entanglement

with the environment. You may want to take a moment to write your favourite type of noise process in this way, to familiarise yourself with the approach. Physicists often consider coupling to the environment by starting from a Hamiltonian, or through a density matrix approach; it is important to see that we are being completely general here, so those other methods can be completely encompassed by (21).

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,

$$X \otimes I \otimes Z \otimes Y \otimes X \equiv X_{10011} Z_{00110}. \tag{22}$$

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 $X_u |\phi\rangle$. Putting all this together, we discover the highly significant fact that to correct *the most general possible* noise (eq. (21)), it is sufficient to correct just X and Z errors. See eq. (28) and following for a proof. It may look like we are only correcting unitary 'bit flip' and 'phase flip' rotations, but this is a false impression: actually we will correct everything, including non-unitary relaxation processes!

5.2. 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 [17] and Calderbank *et. al.* [21, 22]. This introduced the concept of the stabilizer, which yielded important insights into the first discoveries, and enabled significant generalisations to be written down simply.

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, sections 3,4.

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 simultaneous eigenstates. Let $\mathcal{C} = \{|u\rangle\}$ be an orthonormal set of simultaneous eigenstates all having eigenvalue $+1$:

$$M |u\rangle = |u\rangle \quad \forall u \in \mathcal{C},\ \forall M \in \mathcal{H}. \tag{23}$$

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:

$$|\phi\rangle_L = \sum_{u \in \mathcal{C}} a_u |u\rangle \tag{24}$$

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 (24). 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 [3] of the system, cf section 4.1.

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 $E_1 E_2$ of two members is either in $\mathcal{H}$, or anticommutes with a member of $\mathcal{H}$. To see this, take the second case first:

$$E_1 E_2 M = -M E_1 E_2 \;\text{ for some } M \in \mathcal{H}. \tag{25}$$

We say that the combined error operator $E_1 E_2$ is *detectable*. This can only happen if

$$\begin{aligned} &\text{either} \quad \{ME_1 = -E_1 M, \;\; ME_2 = E_2 M\} \\ &\text{or} \quad \{ME_1 \;\;\;\; = E_1 M, \;\; ME_2 = -E_2 M\}. \end{aligned} \tag{26}$$

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 (26) 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 $E\left|\phi\right\rangle_L$, (26) means that 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

$$\sum_i \left(E_i \left|\phi\right\rangle_L\right) \left|\psi_i\right\rangle_e \; . \tag{27}$$

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 (discussed in section 7), 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:

$$\left|0\right\rangle_a \sum_i \left(E_i \left|\phi\right\rangle_L\right) \left|\psi_i\right\rangle_e \rightarrow \sum_i \left|s_i\right\rangle_a \left(E_i \left|\phi\right\rangle_L\right) \left|\psi_i\right\rangle_e . \tag{28}$$

The syndromes s_i are $(n-k)$-bit binary strings. So far the treatment is completely general.

Now suppose the E_i all have different syndromes. Then 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_i as the measurement result. Since there is only one E_i with this syndrome, we can deduce the operator E_i

‡ This is not strictly necessary, see [13], but is convenient.

which should now be applied to correct the error. The system is therefore "magically" disentangled from its environment, and perfectly restored to $|\phi\rangle_L$!

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.

In practice the supposition that all errors generated by the noise have different syndromes will not be true. Usually there will be more than one error E having a given syndrome s_i, just as in the classical case, and we must choose one error per coset (i.e. set having the same syndrome) to be correctable, except in a special case. This is the case mentioned just before eq. (25), namely when

$$E_1 E_2 \in \mathcal{H}. \tag{29}$$

In this case E_1 and E_2 will have the same syndrome, so are indistinguishable in the syndrome extraction process, but both are correctable because we may 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 occurred, this is obviously fine, while if in fact E_2 occurred, 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 (32)§.

5.3. Conditions for quantum error correcting codes

The discussion based on the stabilizer is useful because it focuses 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 [14, 15]

$$\langle u| E_1 E_2 |v\rangle = 0 \tag{30}$$

$$\langle u| E_1 E_2 |u\rangle = \langle v| E_1 E_2 |v\rangle \tag{31}$$

for all $E_1, E_2 \in \mathcal{S}$ and $|u\rangle, |v\rangle \in \mathcal{C}, |u\rangle \neq |v\rangle$. These are the requirements for quantum codewords which correspond to the requirement (9) for classical codes. 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 (see next section) must be satisfied.

The first condition says that to be correctable, an error acting on one codeword must not produce a state which overlaps with another codeword, or with an erroneous version of another codeword. This is what we would expect intuitively. *Proof.* Let the unitary operator $\mathcal{R}$ describe the whole recovery operation (e.g. syndrome extraction followed by correction), then

$$\left.\begin{array}{rcl} \mathcal{R}|\alpha_j\rangle E_j |v\rangle & = & |\alpha'_j\rangle |v\rangle \\ \mathcal{R}|\alpha_i\rangle E_i |u\rangle & = & |\alpha'_i\rangle |u\rangle \end{array}\right\}$$

$$\begin{aligned} \Rightarrow \langle u| E_i^\dagger \langle\alpha_i| \mathcal{R}^\dagger \mathcal{R} |\alpha_j\rangle E_j |v\rangle &= \langle \alpha'_i | \alpha'_j \rangle \langle u | v \rangle \\ \Rightarrow \langle u| E_i^\dagger E_j |v\rangle &= 0. \end{aligned}$$

where $|\alpha\rangle$ are states of the ancilla, environment and any other systems used during recovery. The last step uses the fact that the recovery must work when $|\alpha_i\rangle = |\alpha_j\rangle$ (as well as when $|\alpha_i\rangle \neq |\alpha_j\rangle$).

The second condition, (31) is surprising since it permits one erroneous version of a codeword $|u\rangle$ to overlap with another, as long as all the other codewords when subject to the same error have the same overlap

§ However, codes which break the bound are rare, and in any case they can only go slightly beyond it, c.f. [37]

with their respective other erroneous version. This is not possible in classical error correction because of eq. (11). *Proof of (31)*:

$$\begin{aligned} \mathcal{R}\left|\alpha\right\rangle E_i\left|u\right\rangle &= \left|\alpha_i'\right\rangle\left|u\right\rangle \\ \Rightarrow \left\langle u\right| E_i^\dagger \left\langle\alpha\right| \mathcal{R}^\dagger\mathcal{R}\left|\alpha\right\rangle E_j\left|u\right\rangle &= \left\langle\alpha_i'\right|\left.\alpha_j'\right\rangle \\ \Rightarrow \left\langle u\right| E_i^\dagger E_j\left|u\right\rangle &= \left\langle\alpha_i'\right|\left.\alpha_j'\right\rangle. \end{aligned}$$

The same result is obtained starting from $|v\rangle$, from which (31) is derived. For further information see [3].

5.4. Quantum Hamming bound

A t-error correcting quantum code is defined to be a code for which all errors of weight less than or equal to t are correctable. Since there are 3 possible single-qubit errors, the number of error operators of weight t acting on n qubits is $3^tC(n,t)$. Therefore a t-error correcting code must be able to correct $\sum_{i=0}^{t} 3^iC(n,i)$ error operators. For nondegenerate codes, $\langle u| E_1E_2\left|u\right\rangle = 0$ in (31), every codeword $|u\rangle$ and all its erroneous versions $M\left|u\right\rangle$ must be orthogonal to every other codeword and all its erroneous versions. All these orthogonal states can only fit into the 2^n-dimensional Hilbert space of n qubits if

$$m\sum_{i=0}^{t} 3^iC(n,i) \leq 2^n. \tag{32}$$

This bound is known as the quantum Hamming bound [14, 16, 15, 17]. For $m = 2^k$ and large n, t it becomes

$$\frac{k}{n} \leq 1 - \frac{t}{n}\log_2 3 - H\left(t/n\right). \tag{33}$$

This result is shown in fig. 2. The rate k/n falls to zero at $t/n \simeq 0.18929$.

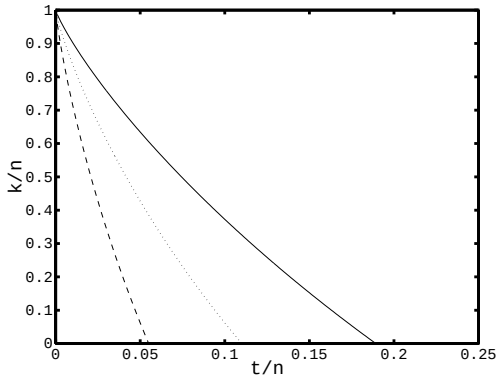

Figure 2. Bounds on code rates. The curves show the code rate k/n as a function of t/n for t-error correcting codes, in the limit $n \to \infty$. Full curve: quantum Hamming bound (eq. (33), this is an upper limit for orthogonal codes, in which every correctable error is associated with an orthogonal syndrome state. Dashed curve: Gilbert-Varshamov type bound (eq. (40)) for CSS codes: codes exist with at least this rate. Dotted curve: Hamming bound for CSS codes $k/n \leq 1 - 2H(t/n)$.

What is the smallest single-error correcting orthogonal quantum code? A code with $m = 2$ codewords represents a Hilbert space of one qubit (it 'encodes' a single qubit). Putting $m = 2$ and $t = 1$ in the quantum Hamming bound, we have $1 + 3n \leq 2^{n-1}$ which is saturated by $n = 5$, and indeed a 5-qubit code exists, see (43).

6. 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. We will first present a general description of stabilizer codes; this will connect the mathematics of error operators to that of binary vector spaces, which has been studied in classical coding theory. The subsequent sections will then examine examples in more detail.

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

$$u \cdot v' + v \cdot u' = 0 \tag{34}$$

The stabilizer is completely specified by writing down any $n - k$ 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

$$H = (H_x | \, H_z) \tag{35}$$

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

$$H_x H_z^T + H_z H_x^T = 0 \tag{36}$$

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

$$H_x G_z^T + H_z G_x^T = 0. \tag{37}$$

In other words, H and G are duals with respect to the inner product defined by (34). 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 (37) and (36) 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 (37), 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 (29) (a degenerate code). The weight d is called the minimum distance of the code. We use the notation $[[n, k, d]]$ to indicate the main properties of the quantum code.

The problem of code construction is thus reduced to a problem of finding binary matrices H which satisfy (36), and whose duals G, defined by (37), have large weights. We will now write down such a code by combining well-chosen classical binary error correcting codes:

$$H = \left(\begin{array}{c|c} H_2 & 0 \\ 0 & H_1 \end{array} \right), \quad G = \left(\begin{array}{c|c} G_1 & 0 \\ 0 & G_2 \end{array} \right). \tag{38}$$

Here H_i, $i = 1, 2$, is the check matrix of the classical code C_i generated by G_i. Therefore $H_i G_i^T = 0$ (eq. (4)) and (37) is satisfied. To satisfy commutativity, (36), we force $H_1 H_2^T = 0$, in other words, $C_2^\perp \subset C_1$. These are the CSS (Calderbank Shor Steane) codes. Their significance is first that they can be efficient (see below), and second that they are useful in fault-tolerant computing.

By construction, if the classical codes underlying a CSS code have parameters $[n, k_1, d_1]$, $[n, k_2, d_2]$ then the quantum code has size $k = k_1 + k_2 - n$. Also, the (non-zero) members of $\mathcal{G}$ have weight at least $\min(d_1, d_2)$ so we have a quantum code of minimum distance $d = \min(d_1, d_2)$. One way to form the quantum codewords is

$$|u\rangle_L = \sum_{x \in C_2^\perp} |x + u \cdot D\rangle \tag{39}$$

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. We can understand the structure of these codewords as follows. Start with the case $u = 0$: $|0\rangle_L$ is an equal superposition of all the members of $C_2^\perp$. The next encoded state is found by displacing all the members of $C_2^\perp$ by the same vector (the first row of D): in other words we have a superposition of all the members of a coset. We choose the vector (the coset leader) so that this coset is still in C_1. The other quantum codewords are formed similarly by further cosets of $C_2^\perp$, all within C_1. Bit flip correction follows from the use of product states all in C_1. Phase flip correction follows from the fact that the Hadamard transform of an equal superposition of members of $C_2^\perp$, or of one of its cosets, will only produce members of C_2 (this is discussed in section 7.2).

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 [12]. This is highly significant: it means there exist codes of useful rate and large minimum distance, and therefore QEC can be a very powerful method to suppress noise, because all the most likely errors from random noise can be in the correctable set (see section 8 for more information). Specifically, it can be shown [12] that there exists an infinite sequence of classical codes which contain their dual and satisfy the Gilbert-Varshamov bound (16). Therefore there exists an infinite sequence of quantum $[[n, K, d]]$ CSS codes provided (16) is satisfied with $k = (K + n)/2$. In the limit of large n, k, t this becomes [12, 13]

$$\frac{K}{n} \geq 1 - 2H\left(2t/n\right) \tag{40}$$

where the usual factor $(1 - \eta)$ has been suppressed. The bound (40) is indicated on fig. 2.

6.1. Some example codes

The simplest CSS code is the 7-bit code discovered by Steane. It uses the CSS construction (38) with $H_1 = H_2 = H$ as given in (8). This is the $[7, 4, 3]$ Hamming code, it is single-error correcting and contains its dual, so leads to a single-error correcting quantum code of parameters $[[n, k, d]] = [[7, 1, 3]]$, i.e. a single qubit stored in 7, with minimum distance 3. The two codewords are

$$\begin{aligned} |0\rangle_L &= |0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle \\ &\quad + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle\,, \end{aligned} \tag{41}$$

$$|1\rangle_L = X_{1111111}\,|0\rangle_L\,. \tag{42}$$

The syndrome extraction operation for this code is illustrated in fig. 3.

There exist QECCs more efficient than CSS codes. Good codes can be found by extending CSS codes, and by other methods. The $[[n, k, d]] = [[5, 1, 3]]$ perfect code encodes a single qubit ($k = 1$), and corrects

all errors of weight 1 (since $(d-1)/1 = 1$). The stabilizer and generator are given by

$$H = \left(\begin{array}{c|c} 11000 & 00101 \\ 01100 & 10010 \\ 00110 & 01001 \\ 00011 & 10100 \end{array}\right), \quad G = \left(\begin{array}{c|c} H_x & H_z \\ 11111 & 00000 \\ 00000 & 11111 \end{array}\right). \tag{43}$$

One possible choice of the two codewords is

$$\begin{aligned} |0\rangle_L &= |00000\rangle + |11000\rangle + |01100\rangle - |10100\rangle \\ &+ |00110\rangle - |11110\rangle - |01010\rangle - |10010\rangle \\ &+ |00011\rangle - |11011\rangle - |01111\rangle - |10111\rangle \\ &- |00101\rangle - |11101\rangle - |01001\rangle + |10001\rangle \end{aligned}$$

and

$$|1\rangle_L = X_{11111}\,|0\rangle_L\,.$$

The expression for $|0\rangle_L$ follows from combining the rows H. The combination $|00000\rangle + |11000\rangle$ is obviously unaffected by the operator $X_{11000}Z_{00101}$ (1st row of H) since this operator converts each of the two product states into the other. We then add the state $|01100\rangle$ with the 2nd row of H in mind, and its partner must be $-|10100\rangle$ so that the result is still an eigenstate of the 1st row of H. It is seen that the combination is then also an eigenstate of $X_{01100}Z_{10010}$ (2nd row of H). The rest follows similarly. The final two rows of G can be regarded as logical X and Z operators in the encoded subspace, and hence we obtain the expression for $|1\rangle_L$. (The choice is not unique: we could equally regard X_{11111} as logical Z and Z_{11111} as logical X, and then the expression for $|1\rangle_L$ would be $Z_{11111}\,|0\rangle_L$.)

To bring out the cyclic structure evident in the matrices, we can also write the state

$$\begin{aligned} |0\rangle_L &= |00000\rangle \\ &+ |11000\rangle + |01100\rangle + |00110\rangle + |00011\rangle + |10001\rangle \\ &- |10100\rangle - |01010\rangle - |00101\rangle - |10010\rangle - |01001\rangle \\ &- |11110\rangle - |01111\rangle - |10111\rangle - |11011\rangle - |11101\rangle\,. \end{aligned}$$

By construction, the codewords are eigenstates of the stabilizer. It is also simple to check this by explicit calculation. To show that the code is a single error correcting code, one can either [20] examine the syndrome for every error vector of weight ≤ 1, or confirm that G generates operators of minimum weight 3.

An example degenerate code is the 9-bit code discovered by Shor (this was in fact the first QECC to be discovered):

$$\begin{aligned} |0\rangle_L &= (|000\rangle + |111\rangle)\,(|000\rangle + |111\rangle)\,(|000\rangle + |111\rangle)\,, \\ |1\rangle_L &= (|000\rangle - |111\rangle)\,(|000\rangle - |111\rangle)\,(|000\rangle - |111\rangle)\,. \end{aligned}$$

The stabilizer and generator are

$$H = \left(\begin{array}{c|c} 111111000 & 000000000 \\ 111000111 & 000000000 \\ 000000000 & 110000000 \\ 000000000 & 101000000 \\ 000000000 & 000110000 \\ 000000000 & 000101000 \\ 000000000 & 000000110 \\ 000000000 & 000000101 \end{array}\right), \quad G = \left(\begin{array}{c|c} H_x & H_z \\ 111111111 & 000000000 \\ 000000000 & 111111111 \end{array}\right). \tag{44}$$

The degeneracy of the code is evidenced by that fact that it corrects all single-qubit errors, even though some different single-qubit errors have the same syndrome (for example, a Z error on any of the first three bits). The correction still works because the product of any pair of such errors is in the stabilizer. The degenerate nature of the code is also seen by the fact that the minimum distance of $\mathcal{G} \setminus \mathcal{H}$ is greater than the minimum distance of $\mathcal{G}$.

The 7-bit and 9-bit code are both CSS codes, the 5-bit code is not. A way to enlarge many CSS codes, producing a code of higher rate which is not CSS, is described in [24].

Finally, a $[[8, 3, 3]]$ code [17, 21, 36, 23] is given by

$$H = \left(\begin{array}{c|c} 11111111 & 00000000 \\ 00000000 & 11111111 \\ 00001111 & 00110011 \\ 00110011 & 01010101 \\ 01010101 & 00111100 \end{array}\right), \quad G = \left(\begin{array}{c|c} 11111111 & 00000000 \\ 00001111 & 00000000 \\ 00110011 & 00000000 \\ 01010101 & 00000000 \\ 00000000 & 11111111 \\ 00000000 & 00001111 \\ 00000000 & 00110011 \\ 00000000 & 01010101 \\ 00000011 & 00000101 \\ 00000101 & 00010001 \\ 00010001 & 00000110 \end{array}\right). \tag{45}$$

Each codeword can be written as a superposition of 16 states, further details are given by Gottesman [17] and Steane [36, 23]. This code is the first of an infinite set of quantum codes based on classical Reed-Muller codes [23, 32].

7. Further insights into coding and syndrome extraction

We will now examine some of the simplest codes in more detail, with the aim of making both the form of the codewords and the extraction of the syndrome physically intuitive.

7.1. Quasi-classical codes

Suppose first of all that the noise only includes error operators composed of tensor products of I and X, so $E_i = X_e$. This mimics the errors occurring in the binary vector spaces of classical error correction. A suitable quantum error correcting code in this situation is the set of codewords $\{|u \in C\rangle\}$ where C is a classical t-error correcting code, and we use the standard notation $|00101\rangle \equiv |0\rangle\,|0\rangle\,|1\rangle\,|0\rangle\,|1\rangle$. It is easy to see that the conditions (31) are satisfied for all error operators of weight $\leq t$, since $X_e\,|u\rangle = |u + e\rangle$, in which u, e and $u + e$ are binary vectors.

In section 5.2 we described the general method to extract syndromes as essentially a measurement of the members M of the stabilizer, using ancillary bits and controlled gates where the ancilla is control bit. The same measurement can also be understood as a measurement of parity checks, with the ancillary bit acting as target. This is the way of extracting the syndrome which was illustrated in section 2. It generalises as follows. Suppose C is an $[n, k, d]$ linear code. We introduce an ancilla of $n - k$ qubits, prepared in $|0\rangle$. To evaluate each classical parity check we perform a sequence of controlled-NOT ($\equiv$ XOR) operations from qubits in q to a single qubit in a. The single qubit in a is the target, and the qubits in q which act as control bits are those specified by the 1's in the n-bit parity check vector. The $n - k$ parity checks specified by the classical parity check matrix H are thus evaluated on the $n - k$ qubits in a. The notation $\mathrm{XOR}_{q,a}^{(H)}$ will be used for this unitary interaction between q and a, an example is shown in fig. 3.

The syndrome extraction operation is

$$\text{XOR}_{q,a}^{(H)} \left(|0\rangle_a |u+e\rangle \right) = \left| He^T \right\rangle_a |u+e\rangle \tag{46}$$

where we used (19). The fact that the classical syndrome is independent of the message is now highly significant, for if the initial state of q is any superposition $\sum_{u\in C} a_u |u\rangle$, the syndrome extraction does not leave q and a entangled:

$$\text{XOR}_{q,a}^{(H)} |0\rangle_a X_e \sum_{u\in C} a_u |u\rangle = \left| He^T \right\rangle_a \sum_{u\in C} a_u |u+e\rangle \tag{47}$$

Recovery is completed by measuring a, deducing e from He^T, and applying $X_e^{-1} = X_e$ to q.

7.1.1. Phase decoherence The noise process just considered is quite unusual in practice. However, it is closely related to a common type of noise, namely phase decoherence. This is the situation where the channel generates random rotations of the qubits about the z axis. Such a rotation is given by the operator

$$P(\epsilon\phi) = \begin{pmatrix} e^{i\epsilon\phi} & 0 \\ 0 & e^{-i\epsilon\phi} \end{pmatrix} \equiv \cos(\epsilon\phi) I + i \sin(\epsilon\phi) Z \tag{48}$$

where I is the identity, ϵ is a fixed quantity indicating the typical size of the rotations, and $-\pi < \phi < \pi$ is a random angle. We may understand such an error as a combination of no error (I) and a phase flip error (Z). Therefore phase decoherence takes the form (21) with error operators consisting of tensor products of I and Z, so $E_i = Z_e$.

The quantum error correcting code for this case is now simple to find because $Z = RXR$, where R is the Hadamard or basis change operation

$$R = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{49}$$

The letter H is often used for this operator, but R is adopted here to avoid confusion with the parity check matrix. We will use $\mathbf{R} = R_1 R_2 \ldots R_n$ to denote Hadamard rotation of all the n qubits in q. Apply $\mathbf{R}$ at the two ends of the channel. The combined effect of phase error and these Hadamard rotations on any given qubit is

$$RPR = \cos(\epsilon\phi) I + i \sin(\epsilon\phi) X \tag{50}$$

Thus we can convert a channel creating phase noise to one creating bit flip noise.

The formal analysis is as follows. The quantum codewords are $|c_u\rangle = \mathbf{R} |u \in C\rangle$ where C is a classical error correcting code. An ancilla is prepared in the state $|0\rangle_a$, and the syndrome extraction operation is $\mathbf{R}\,\text{XOR}_{q,a}^{(H)}\mathbf{R}$ where H is the parity check matrix of C. This can be understood as exactly the same code and extraction as the previous example, only now all operations are carried out in the basis $\{R|0\rangle, R|1\rangle\}$ instead of $\{|0\rangle, |1\rangle\}$. An error acting on a codeword produces

$$Z_e \left(\mathbf{R} |u\rangle \right) = \mathbf{R} X_e |u\rangle = \mathbf{R} |u+e\rangle . \tag{51}$$

Now introduce the ancilla and perform syndrome extraction:

$$\left(\mathbf{R}\,\text{XOR}_{q,a}^{(H)}\mathbf{R} \right) |0\rangle_a \mathbf{R} |u+e\rangle = \mathbf{R}\,\text{XOR}_{q,a}^{(H)} |0\rangle_a |u+e\rangle \tag{52}$$

$$= \left| He^T \right\rangle_a \mathbf{R} |u+e\rangle \tag{53}$$

where we use the fact that $\mathbf{R}$ does not operate on a. The error vector e is deduced from He^T, and the corrective operation Z_e is applied, returning q to $\mathbf{R}|u\rangle$.

The simplest example of a phase error correcting code is a single error correcting code using three qubits. The two quantum codewords are

$$\mathbf{R}|000\rangle = |000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle ,$$
$$\mathbf{R}|111\rangle = |000\rangle - |001\rangle - |010\rangle + |011\rangle - |100\rangle + |101\rangle + |110\rangle - |111\rangle .$$

where the normalisation factor $1/\sqrt{8}$ has been omitted. An equivalent code is one using codewords $\mathbf{R}(|000\rangle \pm |111\rangle)$, which are slightly simpler (they are the sets of even and odd parity words).

Let us now find the fidelity for this 3-bit code example. In the channel under discussion, instead of a X 'bit flip' being applied randomly with probability p to each qubit, every qubit certainly experiences an error which is a combination of the identity and Z. An analysis as in section 2 (c.f. figure 1), modelling the measurements on the ancilla qubits as standard Von-Neumann projective measurements, gives an outcome exactly as before, with the quantity p equal to the average of the squared coefficients of the terms in the quantum state which have single bit-flip errors (before correction). This average is over the random variable ϕ, thus $p = \left\langle \sin^2 \epsilon\phi \right\rangle \simeq (\pi\epsilon)^2/3$ for $\epsilon \ll 1$, where we have considered the worst case, in which the states being transmitted are taken to orthogonal states by the action of Z. The fidelity of the final corrected state is found to be $f \simeq 1 - 3p^2$ for small p. For further details see [13].

7.1.2. Projective errors It is a familiar feature of quantum mechanics that a set of particles where each is in an equal superposition of two states $(|0\rangle + \exp(i\phi)\,|1\rangle)/\sqrt{2}$, with the relative phase ϕ of the two terms random, is indistinguishable from a statistical mixture, i.e. a set of particles where each is either in the state $|0\rangle$ or in $|1\rangle$, randomly. This follows immediately from the fact that these two cases have the same density matrix. With this in mind, it should not be surprising that another type of error which a quasi-classical code such as the three-bit code can correct is an error process that projects the qubit onto the $|0\rangle, |1\rangle$ basis. To be specific, imagine the channel acts as follows: for each qubit passing through, either no change occurs (probability $1-2p$) or a projection onto the $|0\rangle, |1\rangle$ basis occurs (probability $2p$). The projection $|0\rangle\langle 0| = (I+Z)/2$ and $|1\rangle\langle 1| = (I-Z)/2$. Such errors are identical to phase errors (equation(48)), except for the absence of the factor i before the Z term. However, this factor does not affect the argument, and once again the analysis in equation (2) applies (once we have used the Hadamard 'trick' to convert phase errors to bit-flip errors) in the limit of small p. Note that we define p in terms of the effect of the noise on the state: it is not the probability that a projection occurs, but the probability that a Z error is produced in the state of any single qubit when quantum codewords are sent down the channel.

Another way of modelling projective errors is to consider that the projected qubit is first coupled to some other system, and then we ignore the state of the other system. Let such another system have, among its possible states, two states $|\alpha\rangle_e$ and $|\beta\rangle_e$ which are close to one another, $\langle\alpha\,|\,\beta\rangle = 1-\epsilon$. The error consists in the following coupling between qubit and extra system:

$$(a\,|0\rangle + b\,|1\rangle)\,|\alpha\rangle_e \rightarrow a\,|0\rangle\,|\alpha\rangle_e + b\,|1\rangle\,|\beta\rangle_e \tag{54}$$

this is an entanglement between the qubit and the extra system. A useful insight is to express the entangled state in the following way:

$$a\,|0\rangle\,|\alpha\rangle_e + b\,|1\rangle\,|\beta\rangle_e \equiv 2^{-1/2}\left[(a\,|0\rangle + b\,|1\rangle)\,|+\rangle_e + (a\,|0\rangle - b\,|1\rangle)\,|-\rangle_e\right]$$

where $|\pm\rangle_e = (|\alpha\rangle_e \pm |\beta\rangle_e)/\sqrt{2}$. Hence the error on the qubit is seen to be a combination of identity and Z, and is correctable as before. The probability that the Z error is produced is calculated by finding the weights of the different possibilities in equation (2) after tracing over the extra system. We thus obtain $p = (1 - \mathrm{Re}(\langle\alpha\,|\,\beta\rangle))^2/2 = \epsilon^2/2$.

7.2. CSS codes

We now turn to quite general types of noise, where the error operators include X, Y and Z terms. The code construction and correction method discovered by Calderbank, Shor [12] and Steane [11, 13] works by separately correcting the X and Z errors contained in a general error operator $E_s = X_x Z_z$.

The key to the code construction is the "dual code theorem" [11]:

$$\mathbf{R}\sum_{i\in C}|i\rangle = \sum_{i\in C^\perp}|i\rangle\,. \tag{55}$$

The normalisation has been omitted from this expression in order to make apparent the significant features; normalisation factors will be dropped hereafter since they do not affect the argument. The content of (55) is that if we form a state by superposing all the members of a linear classical code, then the Hadamard transformed state is a superposition of all the members of the dual code. We can correct both X and Z errors by using states like those in (55), as long as both C and $C^\perp$ have good error correction abilities.

The codewords of a Calderbank, Shor and Steane (CSS) code can be chosen as in (39). For brevity we will consider $C_2 = C_1 = C$, with $C^\perp \subset C$; the generalisation to $C_1 \neq C_2$ is straightforward. If $C = [n, k, d]$ then $C^\perp = [n, n-k, d^\perp]$. The number of linearly independent u which generate new states is therefore $k - (n-k) = 2k - n$. We can construct 2^{2k-n} orthonormal quantum codewords, which represents a Hilbert space large enough to store $2k - n$ qubits. We will show that the resulting code can correct all errors of weight less than $d/2$. The parameters of the quantum code are thus $[[n, 2k-n, d]]$.

To correct a CSS code obtained from a classical code $C = [n, k, d]$, $C^\perp \subset C$ we introduce two ancillas $a(x)$ and $a(z)$, each of $n - k$ qubits, prepared in the state $|0\rangle$. The syndrome extraction operation is

$$\left(\mathbf{R}\,\mathrm{XOR}^{(H)}_{q,a(z)}\mathbf{R}\right)\mathrm{XOR}^{(H)}_{q,a(x)} \tag{56}$$

where H is the check matrix of C. The proof that this works correctly for all errors $M_s = X_x Z_z$ of weight less than $d/2$ is left as an exercise for the reader. It is straightforward through use of the relations

$$X_x Z_z = (-1)^{x\cdot z} Z_z X_z \tag{57}$$

$$\mathrm{XOR}^{(H)}_{q,a} Z_z = Z_z \mathrm{XOR}^{(H)}_{q,a} \tag{58}$$

$$\mathbf{R} X_s = Z_s \mathbf{R} \tag{59}$$

$$\mathbf{R} \sum_{i\in C^\perp} |i+u\rangle = \sum_{i\in C} (-1)^{i\cdot u} |i\rangle \tag{60}$$

where the latter follows from (55) and (59).

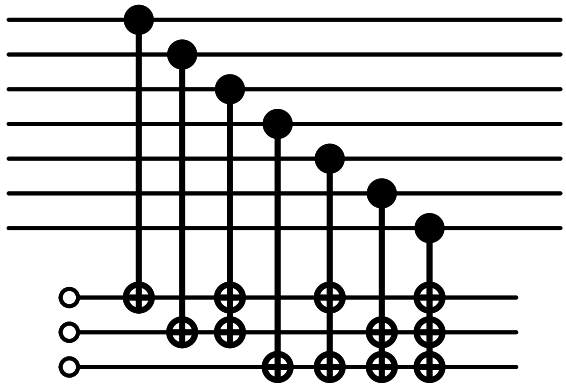

Figure 3. Syndrome extraction operation for $[[7, 1, 3]]$ CSS code. A control with several NOT's represents several controlled NOT operations with the same control qubit. Each of the three qubits of the ancilla begin in $|0\rangle$ and finish in $|0\rangle$ or $|1\rangle$ according as the relevant parity check in H (eq. (8)) is or is not satisfied. The ancilla can then be measured. A further ancilla of three more qubits is required to complete the syndrome, using a similar network together with R operations, see eq. (56).

The syndrome extraction operation is illustrated in fig. 3 for the 7-bit CSS code given in (41).

For a general stabilizer code (not necessarily CSS) the syndrome extraction can always be carried out by a network of the form

$$\left(\mathbf{R}\,\mathrm{XOR}^{(H_x)}_{q,a}\mathbf{R}\right)\mathrm{XOR}^{(H_z)}_{q,a} \tag{61}$$

where we use an $n - K$ bit ancilla prepared in $|0\rangle_a$. The relationship between this network, and the one described after (27) based on the concept of measuring eigenvalues of operators in the stabilizer, is simply in the direction of the controlled-not operations: we can always replace a controlled-not from system to ancilla by a controlled-not going in the other direction and surrounded by Hadamard rotations.

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

$$H_I = \sum_i E_i \otimes H_i^e \tag{62}$$

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 [14, 26, 3]. 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 shorthand 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 (62) into a sum of terms having error operators of different weight:

$$H_I = \sum_{\mathrm{wt}(E)=1} E \otimes H_E^e + \sum_{\mathrm{wt}(E)=2} E \otimes H_E^e + \sum_{\mathrm{wt}(E)=3} E \otimes H_E^e + \dots . \tag{63}$$

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 approximately

$$P(t+1) \simeq \left(3^{t+1} \binom{n}{t+1} \epsilon^{t+1} \right)^2 \tag{64}$$

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

$$P(t+1) \simeq 3^{t+1} \binom{n}{t+1} \epsilon^{2(t+1)} \tag{65}$$

when the errors add incoherently (i.e. either separate environments, or a common environment with couplings of randomly changing phase). The significance of (64) and (65) is that they imply QEC works extremely well when t is large and $\epsilon^2 < t/3n$. Since good codes exist, t can 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 (64) and (65) remain approximately unchanged, if and only if the coupling constants in (63) 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 stabilizer 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.

The obvious approach to take in practice is a combined one, in which we first discover the correlated contributions to the noise in our system, and design a first layer of encoding accordingly, and then overlay a second layer optimized for minimum-distance coding. Such ideas have been studied in classical coding theory, where sometimes many layers of encoding are combined, including tricks such as not placing adjacent physical bits in the same logical block when the code is not designed for burst errors.

The process of encoding one bit in several, and then encoding each of those bits, and so on, is referred to as code *concatenation*. When used recursively, we obtain a powerful code whose behaviour is relatively simple to analyse. This is the structure which underlies the "threshold result", which states that arbitrarily long quantum computations can be made reliable by introducing more and more layers of concatenation, conditioned only that the level of noise per time step and per elementary operation on the physical hardware is below a finite threshold. In other words, we don't require a more and more precise and noiseless computer if we want to evolve longer and longer computations. However, we do need a bigger one.

[1] Preskill J., *Proc. Roy. Soc. Lond. A*, **454** (1998) 385, (quant-ph/9705031).
[2] Lo H.-K., Popescu S., Spiller T. (Editors), *Introduction to quantum computation and information*, (World Scientific, Singapore) 1998.
[3] Nielsen M. A. and Chuang I. L., *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge) 2000.
[4] Shor P. W., in *Proc. 37th Symp. on Foundations of Computer Science*, (IEEE Computer Society Press, Los Alamitos) 1996, pp.15-65.
[5] Unruh W. G., *Phys. Rev. A*, **51** (1995) 992-997.
[6] Palma G. M., Suominen K.-A. and Ekert A. K., *Proc. Roy. Soc. Lond. A*, **452** (1996) 567-584.
[7] Plenio M. B. and Knight P. L., *Proc. R. Soc. Lond. A*, **453** (1997) 2017.
[8] Chuang I. L., Laflamme R., Shor P. W. and Zurek W. H., *Science*, **270** (1995) 1633.
[9] Steane A. M., *Rep. Prog. Phys.*, **61** (1998) 117-173, (quant-ph/9708022).
[10] Shor P. W., *Phys. Rev. A*, **52** (1995) R2493-R2496.
[11] Steane A. M., *Phys. Rev. Lett.*, **77** (1996) 793-767.
[12] Calderbank A. R. and Shor P. W., *Phys. Rev. A*, **54** (1996) 1098-1105.
[13] Steane A. M., *Proc. Roy. Soc. Lond. A*, **452** (1996) 2551-2577.
[14] Knill E. and Laflamme R., *Phys. Rev. A*, **55** (1997) 900-911.
[15] Bennett C. H., DiVincenzo D. P., Smolin J. A. and Wootters W. K., *Phys. Rev. A*, **54** (1996) 3824.
[16] Ekert A. and Macchiavello C., *Phys. Rev. Lett.*, **77** (1996) 2585-2588.
[17] Gottesman D., *Phys. Rev. A*, **54** (1996) 1862-1868.
[18] Bennett C. H., Brassard G., Popescu S., Schumacher B., Smolin J. A. and Wootters W. K., *Phys. Rev. Lett.*, **76** (1996) 722-725.
[19] Deutsch D., Ekert A., Jozsa R., Macchiavello C., Popescu S., and Sanpera A., *Phys. Rev. Lett.*, **77** (1996) 2818.
[20] Laflamme R., Miquel C., Paz J. P. and Zurek W. H., *Phys. Rev. Lett.*, **77** (1996) 198.
[21] Calderbank A. R., Rains E. M., Shor P. W. and Sloane N. J. A., *Phys. Rev. Lett.*, **78** (1997) 405.
[22] Calderbank A. R., Rains E. M., Shor P. W. and Sloane N. J. A., *IEEE Trans. Information Theory*, **44** (1998) 1369-1387.
[23] Steane A. M., *IEEE Trans. Inf. Theory*, **45** (1999) 1701-1703; (quant-ph/9608026).
[24] Steane A. M., *IEEE Trans. Inf. Theory*, **45** (1999) 2492-2495.
[25] Shor P. W. and Laflamme R., *Phys. Rev. Lett.*, **78** (1997) 1600.
[26] Knill E. and Laflamme R., (quant-ph/9608012).
[27] Aharonov D. and Ben-Or M., in *Proc. 29th Ann. ACM Symp. on Theory of Computing*, (ACM, New York) 1998, pp.176; see also quant-ph/9906129, quant-ph/9611025.
[28] van Enk S. J., Cirac J. I. and Zoller P., *Phys. Rev. Lett.*, **78** (1997) 4293.
[29] Bennett C. H. and Shor P. W., *Science*, **303** (2004) 1784-1787.
[30] Hamming R. W., *Bell Syst. Tech. J.*, **29** (1950) 147.
[31] Hamming R. W., *Coding and information theory*, 2nd ed, (Prentice-Hall, Englewood Cliffs) 1986.

[32] MacWilliams F. J. and Sloane N. J. A. , *The theory of error correcting codes*, (Elsevier Science, Amsterdam) 1977.
[33] Jones D. S., *Elementary information theory* (Clarendon Press, Oxford) 1979.
[34] Hill R., *A first course in coding theory* (Clarendon Press, Oxford) 1986.
[35] Shannon C. E., *Bell Syst. Tech. J.*, **27** (1948) 379; also p. 623.
[36] Steane A. M., *Phys. Rev. A*, **54** (1996) 4741-4751.
[37] Shor P. W. and Smolin J. A., (quant-ph/9604006).