Written comprehensive exam: Quantum LDPC Codes

Martin Leslie

May 1, 2012

1 Introduction

This document is an introduction to quantum error correction and quantum low density parity check (LDPC) codes.

Quantum computers promise to provide significant speedups over classical computers in certain problems. The most famous example is factoring integers using Shor's algorithm, where the quantum algorithm is exponentially faster than any known classical algorithm. For an introduction to quantum algorithms, which we will not discuss in this paper, see [NC].

However, quantum computers face errors: unwanted unitary evolution of a quantum state and unwanted measurements. To correct these errors we need quantum error correction. Quantum error correction is quite similar to classical error correction but with some important differences.

In classical error correction LDPC codes, first invented in the 1960's by Gallager and then reinvented in the 1990's by Mackay, come close to achieving the capacity of the binary symmetric channel by using a suboptimal iterative decoder on an ensemble of random linear codes. The hope is that quantum LDPC codes will come close to achieving the capacity of the analogous quantum depolarizing channel. This hope has not yet come to fruition but practical quantum LDPC codes do have good performance, as shown in [MMM04]. We provide the background to understand that paper and discuss the construction and simulation of Bicycle codes within it. We also discuss the decoding of quantum LDPC codes, for which [PC08] is a good reference.

Section 2 of this document gives an introduction to the finite dimensional quantum mechanics we need to understand quantum computing. Section 3 then gives a brief overview of the quantum computing model and quantum circuit diagrams. Next, section 4 discusses classical coding theory and information theory, giving both necessary mathematical tools and a classical analog for our quantum error correction to come. Section 5 discusses quantum operations and the depolarizing channel - our source of noise. Section 6 introduces the main class of codes we will be interested in, called stabilizer codes. Finally, section 7 introduces quantum LDPC codes, discusses their construction and decoding, as well as providing some performance bounds and simulations of codes.

To explain our notation: We will use Dirac 'bra-ket' notation (for a brief introduction see Section A.2). The finite field of order q is $\mathbb{F}_q$. In the cases that it matters log will mean $\log_2$. Other notations will be introduced as needed.

2 Quantum mechanics

This introduction to quantum mechanics for quantum computing follows [NC]. Our quantum computing model is based on *qubits* (named for quantum bits) although more general qudits (quantum digits) or other systems are possible. Choose a basis (called the *computational basis*)

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

for $\mathbb{C}^2$. Now a qubit is an element of

$$\mathcal{H}_1 = \{|\psi\rangle = a|0\rangle + b|1\rangle \colon a, b \in \mathbb{C} \text{ with } |a|^2 + |b|^2 = 1\}$$

where we consider two elements to be indistinguishable if they differ multiplicatively by a complex number. So really the space of qubits is

$$\mathbb{P}(\mathbb{C}^2) = \mathbb{C}^2 \setminus \{(0,0)\}/\sim$$

where $|\psi\rangle \sim \lambda|\psi\rangle$ for all $\lambda \in \mathbb{C} \setminus \{0\}$. However, as is conventional we will work with an equivalence class representative which is normalized and can ignore global phase factors when looking at the whole system.

This is the first example of a state space, which we can think of as we go through the postulates of quantum mechanics.

2.1 The postulates of quantum mechanics

(i) Associated to any isolated physical system is a complex Hilbert space, known as the state space.

(ii) The evolution of a closed quantum system is described by a unitary transformation. That is, the state $|\psi\rangle$ of the system at time t_1 is related to the state $|\psi'\rangle$ of the system at time t_2 by $U = U(t_1, t_2)$ via $|\psi'\rangle = U|\psi\rangle$.

(iii) A quantum measurement is described by a collection $\{M_m\}$ of measurement operators acting on the state space of the system (one operator for each outcome of the measurement). If the state of the system is $|\psi\rangle$ before the measurement then the probability of outcome m is

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle$$

and if outcome m occurs then the state of the system after the measurement is

$$\frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}.$$

The measurement operators satisfy $\sum_m M_m^\dagger M_m = I$.

(iv) The state space of a composite physical system is the tensor product of the state spaces of the component physical systems.

These postulates deserve some comments: they do not determine which state space a system has associated to it or which unitary transformations act on it. We are in the lucky position of building a system rather than describing one so we restrict ourselves to tensor products of qubits and we will describe some useful unitary transformations (called quantum gates) in the next section.

Let the state space of n-qubits be $\mathcal{H}_n = \mathcal{H}_1^{\otimes n}$ (to be more precise, we take the tensor product of the underlying vector space and then normalize the elements of the new vector space). This has basis $|i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle$ where $i_j \in \{0, 1\}$ so an element $|\psi\rangle \in \mathcal{H}_n$ can be written as

$$|\psi\rangle = \sum_{i \in \mathbb{F}_2^n} a_i |i\rangle$$

where $|i\rangle = |i_1 i_2 \dots i_n\rangle = |i_1\rangle|i_2\rangle \cdots |i_n\rangle = |i_1\rangle \otimes |i_2\rangle \otimes \cdots \otimes |i_n\rangle$ are all just different notations for the same thing. The normalization condition means $\sum |a_i|^2 = 1$ and once again we can ignore global phase factors.

Following [NC] we have given a more general measurement postulate than most non quantum-information physics texts. To see how our measurements relate to the more traditional model we can instead start with an *observable* M, a Hermitian operator on the state space, and speak of measuring M. The operator M has a spectral decomposition $M = \sum_m mP_m$ where P_m is the projector onto the eigenspace of M with eigenvalue m (so $P_m P'_m = \delta_{mm'} P_m$ i.e. the P_m are orthogonal projectors). Then the possible outcomes of the measurement are the eigenvalues m and the probability of getting outcome m is

$$p(m) = \langle\psi|P_m|\psi\rangle.$$

After the measurement with outcome m the system will be in state $P_m|\psi\rangle/\sqrt{p(m)}$. This is equivalent to measuring the operators $\{P_m\}$ in our measurement postulate.

A very useful result is that if two observables commute then they can be measured simultaneously. To explain this further, recall that if two Hermitian operators A, B commute then they are simultaneously diagonalizable, say

$$A = \sum_i \lambda_i |i\rangle\langle i| \text{ and } B = \sum_i \mu_i |i\rangle\langle i|,$$

If the state of the system before measurement is $|\psi\rangle = \sum_i a_i|i\rangle$ then measuring A gives result λ_i with probability $|a_i|^2$, leaving the system in the state $|i\rangle\langle i|$. Then measuring B gives result μ_i and the system remains in state $|i\rangle\langle i|$. But measuring in the other order, B then A, will give the same results and leave the system in the same state.

2.2 Density matrices

Density matrices allow us to deal with probabilistic ensembles of quantum states (*mixed states* as compared to the *pure states* we have been discussing so far). If

a system is in state $|\psi_i\rangle$ with probability p_i then we say it has *density matrix*

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|.$$

Now consider unitary evolution of this ensemble. After some period of time the state will be $U|\psi_i\rangle$ with probability p_i so the new density matrix is

$$\rho' = \sum_i p_i U|\psi_i\rangle\langle\psi_i|U^\dagger = U\left(\sum_i p_i|\psi_i\rangle\langle\psi_i|\right)U^\dagger.$$

In other words, unitary evolution of density matrices is carried out by

$$\rho \mapsto U\rho U^\dagger.$$

A similar argument, along with some elementary probability, shows that for measurement operators $\{M_m\}$ the probability of outcome m is

$$p(m) = \mathrm{tr}(M_m^\dagger M_m \rho).$$

If outcome m occurs then the new state is

$$\rho_m = \frac{M_m \rho M_m^\dagger}{\mathrm{tr}(M_m^\dagger M_m \rho)}.$$

Example 2.1. This example shows that the classical probability of mixed states is truly different to the probabilistic measurement of pure states. Consider the pure state $|\psi\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, with density matrix

$$\rho_1 = |\psi\rangle\langle\psi| = \frac{1}{2}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| + |1\rangle\langle 1|) = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Alternatively, consider a mixed state prepared with equal probability of being $|0\rangle$ or $|1\rangle$. This has density matrix

$$\rho_2 = \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1| = \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

If we measure these systems with respect to measurements $P_0 = |0\rangle\langle 0|$ and $P_1 = |1\rangle\langle 1|$ both give outcomes 0 or 1 each with probability half.

However we can instead measure with respect to $P_+ = |+\rangle\langle +|$ and $P_- = |-\rangle\langle -|$ where $|\pm\rangle = \frac{|0\rangle \pm |1\rangle}{\sqrt{2}}$. Then the first system will always be measured as a 0, while the second system still gives 0 and 1 with probability half.

A density matrix ρ is a linear operator on the quantum state space Q. The following proposition shows which linear operators are density matrices.

Proposition 2.2. *A matrix ρ is the density matrix of some ensemble $\{p_i, |\psi_i\rangle\}$ iff ρ is a positive, trace one operator.*

Proof. If

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$$

then

$$\mathrm{tr}(\rho) = \sum_i p_i \mathrm{tr}(|\psi_i\rangle\langle\psi_i|) = \sum_i p_i = 1.$$

Also, for any state $|\phi\rangle$ we have

$$\begin{aligned}\langle\phi|\rho|\phi\rangle &= \sum_i p_i \langle\phi|\psi_i\rangle\langle\psi_i|\phi\rangle \\ &= \sum_i p_i |\langle\phi|\psi_i\rangle|^2 \\ &\geq 0\end{aligned}$$

so ρ is positive.

Conversely, a positive operator is Hermitian and therefore has a spectral decomposition

$$\rho = \sum_i \lambda_i |i\rangle\langle i|$$

where $\lambda_i \geq 0$ and $|i\rangle$ are orthonormal. The trace one condition means that $\sum_i \lambda_i = 1$ and thus ρ is a density matrix for the ensemble $\{\lambda_i, |i\rangle\}$. □

Notice that the results of time evolution and of measurements only depends on the density matrix of a state, not on the details of the ensemble. Thus we can in fact reformulate the postulates of quantum mechanics in terms of density matrices. The next example shows that a density matrix can come from more than one ensemble of pure states.

Example 2.3. Consider first a mixed state prepared with equal probability of being $|0\rangle$ or $|1\rangle$. As seen earlier it has density matrix

$$\rho_1 = \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1| = \frac{1}{2}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}.$$

Next, consider a mixed state prepared with equal probability of being $|+\rangle$ or $|-\rangle$. It has density matrix

$$\rho_1 = \frac{1}{2}|+\rangle\langle +| + \frac{1}{2}|-\rangle\langle -| = \frac{1}{4}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix} + \frac{1}{4}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}.$$

3 Quantum computing

We use quantum gates and measurements for our quantum computing model, following the description in [NC]. A quantum gate is a unitary transformation. The *Pauli matrices* are unitary Hermitian operators that act on one qubit:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

In the context of error correction we can think of X as a bit flip

$$X(a|0\rangle + b|1\rangle) = a|1\rangle + b|0\rangle$$

Z as a phase flip

$$Z(a|0\rangle + b|1\rangle) = a|0\rangle - b|1\rangle$$

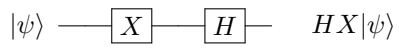
and $Y = iXZ$ as both (along with an insignificant global phase factor). The *Hadamard gate*

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

takes $|0\rangle$ and $|1\rangle$ to the *dual basis*

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \text{ and } |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

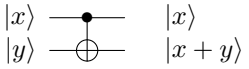
In quantum circuit diagrams we will represent a single qubit by a quantum wire (a line) and the action of a quantum gate by that gate inside a rectangle. In the following diagram a qubit $|\psi\rangle$ enters the circuit on the left and then has unitary operations X and H applied to it.



We can also use controlled gates which act on more than one qubit. The CNOT gate acts on two qubits by

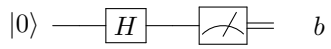
$$|00\rangle \mapsto |00\rangle$$
$$|01\rangle \mapsto |01\rangle$$
$$|10\rangle \mapsto |11\rangle$$
$$|11\rangle \mapsto |10\rangle$$

extended linearly. We represent it by the following circuit where the top wire is the control and the bottom wire is the target. The diagram demonstrates the result if $x, y \in \mathbb{F}_2$.



In our model we can only make measurements in the computational basis i.e. with respect to $P_0 = |0\rangle\langle 0|$ and $P_1 = |1\rangle\langle 1|$. We represent this measurement

with a meter symbol and then the result of a measurement is a classical bit which can travel over a classical wire, denoted by double lines. For example the following quantum circuit generates a random bit b which is either 0 or 1 with equal probability.



4 Classical coding theory

Classical information theory and coding theory were initiated by the works of Shannon and Hamming in 1948 and 1950 respectively. This section draws from a number of sources including [NC], [Mac03], [RU08] and [Gur10] to give an introduction to some of the ideas we need for quantum error correction.

4.1 Linear codes

Fix q a power of a prime (in all our examples q will be 2 or 4) and consider the vector space $\mathbb{F}_q^n$. This has a symmetric bilinear form (see Section A.1 for an introduction)

$$(x, y) \mapsto x \cdot y = \sum_i x_i y_i$$

which can easily be checked to be non-degenerate.

Define an $[n, k]_q$ *linear code* C to be a k-dimensional subspace of $\mathbb{F}_q^n$. We say that $G \in M_{n\times k}(\mathbb{F}_q)$ is a *generator matrix* for C if

$$C = \{Gx \colon x \in \mathbb{F}_q^k\}.$$

A matrix $H \in M_{m\times n}(\mathbb{F}_q)$ is called a *parity check matrix* for C if

$$C = \{x \in \mathbb{F}_q^n \colon Hx = 0\}.$$

From these definitions it follows that $\operatorname{rank}(G) = k$ as well as $\operatorname{rank}(H) = n-k$. To see the first equation we have

$$\dim(\operatorname{im}(G)) + \dim(\ker(G)) = \dim(\mathbb{F}_q^k)$$

but $\operatorname{im}(G) = C$ means that G is injective so has full rank k. Similarly

$$\dim(\operatorname{im}(H)) + \dim(\ker(H)) = \dim(\mathbb{F}_q^n)$$

which gives $\dim(\operatorname{im}(H)) = n-k$. Notice that with our definition G must be full rank and H doesn't have to be. We allow the parity check matrix to include $m \geq n-k$ conditions of which $m-(n-k)$ must be redundant. This is purely for convenience: many of our code constructions will be via specifying parity check matrices which may not necessarily be full rank.

Furthermore we have $HG = 0$ because columns of G are elements of C and H times an element of C is 0.

To form a generating matrix for a code C choose a basis for C and place these as columns of G. Then we can generate all the codewords by adding basis codewords i.e. multiplying G by some x.

To form a full rank parity check matrix for C we consider the orthogonal complement

$$C^\perp = \{y \in \mathbb{F}_q^n \colon y \cdot x = 0 \text{ for all } x \in C\},$$

called the *dual code* of C. One thing to note is that we do not necessarily have $C \cap C^\perp = \{0\}$ (for example if $q = 2$ then even weight codewords are orthogonal to themselves). However our bilinear form is nondegenerate so we do still have

$$\dim(C) + \dim(C^\perp) = \dim(\mathbb{F}_q^n)$$

and thus $\dim(C^\perp) = n - k$. Now choose a basis for $C^\perp$ and use this as the rows of H.

To show that H is a parity check matrix for C we need

$$C = \{x \in \mathbb{F}_q^n \colon Hx = 0\}.$$

If $x \in C$ then $Hx = 0$ because each row of H is orthogonal to all elements of C. If $Hx = 0$ then we know that x is orthogonal to a basis for $C^\perp$ and is thus orthogonal to $C^\perp$. Thus $x \in (C^\perp)^\perp = C$ (to see this last equality note that $C \subseteq (C^\perp)^\perp$ and that $\dim(C^\perp) + \dim((C^\perp)^\perp) = n$).

We now claim that $C^\perp$ is a $[n, n-k]_q$ code with generator matrix H^T and parity check matrix G^T. To see that H^T is a generator matrix notice that the rows of H are a basis for $C^\perp$ and thus any column vector in $C^\perp$ can be written as a sum of columns of H^T. For the parity check matrix: if $x \in C^\perp$ then x is orthogonal to all elements of C so is orthogonal to columns of G so $G^T x = 0$. Finally if $G^T x = 0$ then x is orthogonal to a basis of C so is orthogonal to C so $x \in C^\perp$.

We say that a code is *weakly self-dual* (or *self-orthogonal*) if $C \subseteq C^\perp$ and *strictly self-dual* (or just *self-dual*) if $C = C^\perp$. Being weakly self-dual is equivalent to $G^T G = 0$. To see this note that the elements of $G^T G$ are dot products of elements of C with each other. If $C \subseteq C^\perp$ then these all must be zero. Also if $G^T G = 0$ then all basis elements of C are orthogonal to each other so all elements of C must be orthogonal to each other and thus $C \subseteq C^\perp$.

4.2 Codes over $\mathbb{F}_4$

We write $\mathbb{F}_4 = \{0, 1, \omega, \omega + 1\}$ where $\omega^2 = \omega + 1$. Notice that $\mathbb{F}_4$ is a Galois extension of $\mathbb{F}_2$ (it's the splitting field of $x^4 + x$) with $\mathrm{Gal}(\mathbb{F}_4/\mathbb{F}_2) = \{\mathrm{id}, \overline{}\}$ where $\overline{}$ takes ω to $\omega + 1$ and vice versa. We can also write $\overline{x} = x^2$. Then the trace from $\mathbb{F}_4$ to $\mathbb{F}_2$ is given by $\mathrm{tr}(x) = x + \overline{x}$. A table of these operations is given below.

x	$\overline{x}$	$\mathrm{tr}(x)$
0	0	0
1	1	0
ω	$\omega + 1$	1
$\omega + 1$	ω	1

We have the *Hermitian bilinear form* on $\mathbb{F}_4^n$ as an $\mathbb{F}_4$-vector space given by

$$(x, y) \mapsto \overline{x} \cdot y = \sum_i \overline{x_i} y_i.$$

We also have the *trace Hermitian bilinear form* on $\mathbb{F}_4^n$ as an $\mathbb{F}_2$-vector space given by

$$(x, y) \mapsto \operatorname{tr}(\overline{x} \cdot y) = \sum_i \operatorname{tr}(\overline{x_i} y_i).$$

Both of these forms can be checked to be symmetric and nondegenerate.

Define the *Hermitian dual* of an $[n, k]_4$ code to be

$$C^{\perp_H} = \{y \in \mathbb{F}_4^n : \overline{y} \cdot x = 0 \text{ for all } x \in C\}.$$

Codes that are Hermitian self-dual have been extensively studied in algebraic coding theory. However in our application of this theory we are actually more interested in codes that are self-dual with respect to the trace Hermitian form. Clearly codes self-dual with respect to the Hermitian inner form are a subset of codes self-dual with respect to the trace Hermitian form.

4.3 Performance of binary codes

The codes we defined previously in this section are not just algebraic or combinatorial curiosities. In particular, the codes over $\mathbb{F}_2$ can be used to correct errors coming from the binary symmetric channel (BSC). The *binary symmetric channel* with *crossover probability* p takes one bit as an input and then gives an output which is the input bit with probability $1 - p$ and that bit flipped with probability p. We assume $p \leq 1/2$ (otherwise we should just swap 0's and 1's at the output).

Assume we have a $[n, k]_2$ code. To transmit $u \in \mathbb{F}_2^k$ we send $x = Gu$ over the BSC. The BSC acts independently on each bit and we receive $y = x + r$ where $r \in \mathbb{F}_2^n$. Then the optimal decoding algorithm is to decide that the sent codeword is codeword $x^* \in C$ which is closest to y. Here closest is with respect to the Hamming distance

$$d_H(a, b) = \text{number of elements in which } a \text{ and } b \text{ differ.}$$

Then to recover the information we decode to the unique $u^* \in \mathbb{F}_2^k$ which corresponds to x^*.

With this in mind, when designing a linear code we would like codewords to be far apart. The *minimum distance* d of a code is the minimum of $d_H(a, b)$ for all $a \neq b \in C$. For linear codes this minimum distance is also the minimum weight of nonzero codewords where the weight of a codeword is

$$\operatorname{wt}(a) = d_H(a, 0) = \text{number of 1's in } a.$$

To see this, we have

$$d = \min_{a \neq b \in C} d_H(a, b) = \min_{a \neq b \in C} \operatorname{wt}(a + b) = \min_{c \in C \setminus \{0\}} \operatorname{wt}(c).$$

We will refer to such a code as an $[n, k, d]_2$-code. Much work has been done in finding, for a given n and d, the largest possible k.

However, for practical applications what we really want is a code with largest information rate $R = k/n$ which can send information reliably over a channel. Rather than explain the theory of noisy channel capacity we will merely give a method for calculating capacity when the channel is symmetric in a certain sense. For details see Chapter 7 of [CT06].

Define the *entropy* of a probability distribution (i.e. $\{p_i\}$ with $\sum p_i = 1$) to be

$$H(\{p_i\}) = -\sum p_i \log(p_i).$$

Consider a *discrete memoryless channel* which takes input from one finite alphabet $\mathcal{X}$ and gives output from a finite alphabet $\mathcal{Y}$ where the probability of an output given an input is given by the *transition matrix* $p(y \mid x)$. If each row of the transition matrix is a permutation of every other row and all column sums are equal then we call the channel *weakly symmetric*.

Theorem 4.1. *A weakly symmetric discrete memoryless channel has capacity*

$$C = \log |\mathcal{Y}| - H(\textit{row of transition matrix}).$$

This means that if we fix an $R < C$ then for all $\epsilon > 0$ there exists a sufficiently large n and a code with information rate $k/n \geq R$ and probability of decoding a block incorrectly less than ϵ.

Proof. See [CT06]. Another way of proving this, at least in the BSC case, is showing that, for sufficiently large n, linear codes specified by random parity check matrices achieve this on average, so there must be a specific code that achieves this performance. See Chapter 14 of [Mac03] for details. □

Example 4.2. One example that we will need when we discuss quantum error correction is the 4-ary symmetric channel with error probability p. This has transition matrix

$$p(y \mid x) = \begin{bmatrix} 1-p & p/3 & p/3 & p/3 \\ p/3 & 1-p & p/3 & p/3 \\ p/3 & p/3 & 1-p & p/3 \\ p/3 & p/3 & p/3 & 1-p \end{bmatrix}$$

which is weakly symmetric and thus the channel has capacity

$$\begin{aligned} C &= \log 4 - H(1-p, p/3, p/3, p/3) \\ &= 2 + (1-p)\log(1-p) + 3(p/3)\log(p/3) \\ &= 2 + (1-p)\log(1-p) + p\log p - p\log 3 \\ &= 2 - h_2(p) - p\log 3 \end{aligned}$$

where $h_2(p) = H(p, 1-p) = -p\log p - (1-p)\log(1-p)$ is the *binary entropy function*.

Example 4.3. More in the spirit of this section, consider the BSC with crossover probability p. This has transition matrix

$$p(y \mid x) = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}$$

which is weakly symmetric and thus the channel has capacity

$$\begin{aligned} C &= \log 2 - H(p, 1-p) \\ &= 1 - h_2(p). \end{aligned}$$

We interpret this as meaning that we can send on average $1 - h_2(p)$ bits of information through the BSC per channel use.

Interestingly, the decoding scheme suggested by the idea of minimum distance does not achieve the capacity of the binary symmetric channel. If we use a *bounded distance decoder* which will only decode codewords in radius d spheres around each codeword then it is conjectured that the best we can achieve is the Gilbert-Varshamov bound

$$R_{GV} = 1 - h_2(2p).$$

The next two results show that we can achieve performance at least this good using bounded-distance decoding.

Proposition 4.4. *For any d there exists a sufficiently large n and an $[n, k, d]_2$ code with $k/n \geq 1 - h_2((d-1)/n)$.*

Proof. We assemble some results from Lecture 2 of [Gur10]. By choosing a minimum-distance d linear code greedily there exists an $[n, k, d]_2$ code with

$$k \geq n - \log_2 \left(\sum_{j=0}^{d-1} \binom{n}{j} \right).$$

Now we use an estimate on volume of Hamming balls

$$\sum_{j=0}^{d-1} \binom{n}{j} \leq 2^{h_2((d-1)/n)n}$$

to get

$$k/n \geq 1 - h_2 \left(\frac{d-1}{n} \right).$$

□

Proposition 4.5. *Fix $p < 1/4$ and $R < R_{GV}$. Then for all $\epsilon > 0$ there exists a sufficiently large n and an $[n, k, d]_2$ code with $k/n > R$ which can be bounded-distance decoded with probability of failure less than ϵ.*

Proof. The following argument is based on one given in [RU08]. First notice that $R_{GV}(p)$ is a decreasing function of p that goes from 1 to 0 as p goes from 0 to 1/4. So since $R < R_{GV}$ we must have $R = 1 - h_2(2p + w)$ for some $w > 0$.

Let $d = \lfloor (2p + w)n \rfloor + 1$. Notice that

$$\frac{d-1}{n} = \frac{\lfloor (2p+w)n \rfloor}{n} \leq 2p + w$$

so by the previous proposition and the fact that $1-h_2(p)$ is a decreasing function of p there exists an $[n, k, d]_2$ code with

$$\frac{k}{n} \geq 1 - h_2\left(\frac{d-1}{n}\right) \geq 1 - h_2(2p + w).$$

Let the random variable X be the weight of an error of length n created by a BSC of crossover probability p. This variable is binomially distributed with $X \sim B(n, p)$ so $E[X] = pn$ and $\mathrm{Var}[X] = np(1 - p)$. Then by Chebyshev's inequality we have

$$\begin{aligned}
&P(X \geq pn + c\sqrt{n}) \\
=&P(X - pn \geq c\sqrt{n}) \\
\leq&P(|X - pn| \geq c\sqrt{n}) \\
\leq&E[(X - pn)^2]/(c\sqrt{n})^2 \\
=&p(1 - p)/c^2.
\end{aligned}$$

Now choose c sufficiently large so that $p(1-p)/c^2 < \epsilon$. Then choose n sufficiently large (to be precise $n > (2c/w)^2$) such that $pn + wn/2 > pn + c\sqrt{n}$.

So now notice

$$\frac{d}{2} = \frac{\lfloor (2p+w)n \rfloor + 1}{2} \geq (p + w/2)n > pn + c\sqrt{n}$$

and thus

$$\begin{aligned}
&P(\text{bounded distance decoder fails}) \\
=&P\left(X \geq \frac{d}{2}\right) \\
\leq&P(X \geq pn + c\sqrt{n}) \\
<&\epsilon.
\end{aligned}$$

□

The upshot of these results is that to achieve Shannon capacity we need to decode beyond the minimum distance.

4.4 Classical LDPC codes

Inspired by the kind of discussion in the previous section, LDPC codes use random or random-like parity check matrices that have been chosen to be low-density i.e. have a low proportion of ones. These kinds of codes can be proven to be good in the sense that families of such codes can achieve the capacity of the BSC with optimal decoding. However optimal decoding is computationally infeasible so the exciting feature of LDPC codes is that for low density parity check matrices there exists a suboptimal decoder which can run in reasonable time.

Consider the parity check matrix to be an adjacency matrix of a bipartite graph, called the *Tanner graph*. Each column of the parity check matrix represents a bit of the received string, called a variable node, and each row represents a constraint, called a parity check. Then the decoder, called *belief propagation* (BP), works by sending messages along the edges of the Tanner graph representing probabilities of a bit being a certain value or a check being satisfied. This decoder runs iteratively, sending messages back and forth between checks and variables until either the variable nodes beliefs converge to a codeword or until a fixed number of iterations, when a decoding failure is declared.

We will not give details of classical LDPC codes now, but will derive the required equations for quantum LDPC codes later. To give some support to the idea that classical LDPC codes are a correct answer to the problem of communicating reliably over the BSC we include a figure from [Mac03].

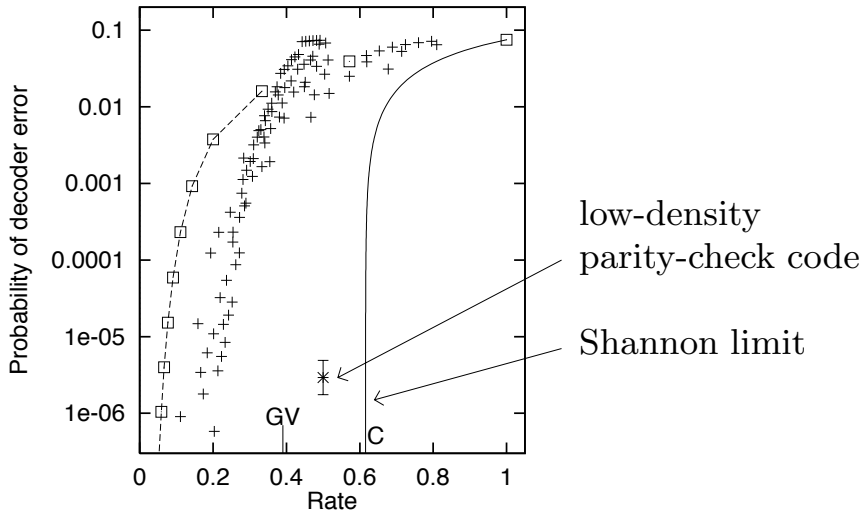


Figure 1: A plot of probability of decoder error against code rate from [Mac03]. The non-labelled data points are various algebraic codes: repetition, Hamming, Reed-Muller, BCH. The channel is the BSC with $p = 0.075$. The LDPC code is specified by a 10000×20000 parity check matrix with three ones per column.

5 Quantum operations

Quantum operations are a formalism that unifies unitary evolution and quantum measurement and allow us to introduce quantum channels. We give the axiomatic approach to quantum operations, mainly following [NC].

Let $L(Q)$ denote the set of linear operators on Q. If Q_1 and Q_2 are state spaces of two quantum systems consider a function $\mathcal{E}\colon L(Q_1) \to L(Q_2)$. We say $\mathcal{E}$ is *positive* if it takes positive operators to positive operators. We say it is *completely positive* if for any extra system R it is true that $(\mathcal{I} \otimes \mathcal{E})(A)$ is positive for any positive operator A on the composite system RQ_1, where $\mathcal{I}$ is the identity operator on $L(R)$.

A *quantum operation* is an operator $\mathcal{E}\colon L(Q_1) \to L(Q_2)$ satisfying the following axioms:

(i) For any density matrix ρ on Q_1 we have $0 \leq \mathrm{tr}[\mathcal{E}(\rho)] \leq 1$,

(ii) $\mathcal{E}$ is linear, and

(iii) $\mathcal{E}$ is a completely positive map.

Note that we depart from [NC] in insisting that our quantum operations are defined on all linear operators, not just density matrices (also, in that text the second axiom is that $\mathcal{E}$ is convex-linear). This change is not strictly required because, as is shown in [Ryc12], any convex-linear operator defined on density matrices can be extended uniquely to a linear operator on all matrices. We make our definition for convenience, as the proof of this fact is somewhat involved and the examples of quantum operations we are interested in are clearly linear.

We now explain why these axioms are reasonable. We can interpret $\mathrm{tr}[\mathcal{E}(\rho)]$ as the probability of operation $\mathcal{E}$ occurring when the system began in state ρ. If the probability is one for all ρ we say the operation is *trace preserving*. The second axiom (in its convex-linear form) can be understood as saying that if we randomly select an initial state from an ensemble of states then we get the expected result chosen from the ensemble of resulting states. The third axiom comes from the fact that if we begin with a density matrix on a composite system RQ_1 then it should remain a valid density matrix upon applying our operator only to the Q_1 part.

We will prove only the easy direction of the following characterization of quantum operations. For the remainder of the proof see [NC].

Proposition 5.1. *The map $\mathcal{E}\colon L(Q_1) \to L(Q_2)$ is a quantum operation iff*

$$\mathcal{E}(A) = \sum_i E_i A E_i^\dagger$$

for some set of operators E_i with $\sum_i E_i^\dagger E_i \leq I$.

Proof. We show that if $\sum_i E_i^\dagger E_i \leq I$ then $\mathcal{E}(A) = \sum_i E_i A E_i^\dagger$ is a quantum operation. The linearity of $\mathcal{E}$ is clear. For the trace condition notice that

$$\begin{aligned}
\operatorname{tr}(\mathcal{E}(\rho)) &= \operatorname{tr}\left(\sum_i E_i \rho E_i^\dagger\right) \\
&= \operatorname{tr}\left(\sqrt{\rho}\sum_i E_i^\dagger E_i \sqrt{\rho}\right) \\
&= \sum_j \langle j|\sqrt{\rho}\sum_i E_i^\dagger E_i \sqrt{\rho}|j\rangle \\
&\leq \sum_j \langle j|\sqrt{\rho}\sqrt{\rho}|j\rangle \\
&= \sum_j \langle j|\rho|j\rangle \\
&= \operatorname{tr}(\rho) \\
&= 1
\end{aligned}$$

where the inequality follows from the assumption $\sum_i E_i^\dagger E_i \leq I$.

For complete positivity, we need $\langle\phi|(\mathcal{I}\otimes\mathcal{E})(A)|\phi\rangle \geq 0$ for all positive $A \in L(RQ_1)$. But

$$\begin{aligned}
&\langle\phi|(\mathcal{I}\otimes\mathcal{E})(A)|\phi\rangle \\
=&\langle\phi|\sum_i (I_R\otimes E_i)A(I_R\otimes E_i^\dagger)|\phi\rangle \\
=&|\sum_i \langle\phi|(I_R\otimes E_i)A(I_R\otimes E_i)^\dagger|\phi\rangle \\
\geq&0
\end{aligned}$$

where the inequality follows from A being positive. □

Examples of quantum operations that are of this *operator sum* form include unitary evolution $\mathcal{E}(\rho) = U\rho U^\dagger$ and measurements with a particular outcome $\mathcal{E}_m(\rho) = M_m \rho M_m^\dagger$.

5.1 Depolarizing channel

The depolarizing channel acting on one qubit

$$\mathcal{E}\colon L(\mathcal{H}_1) \to L(\mathcal{H}_1)$$

is an example of a quantum operation that we will think of as being the source of errors for our quantum error correcting codes. We can think of it in two different ways. First we think of it as an operation that with probability $1-q$

leaves the qubit untouched and with probability q replaces it by the completely mixed state $I/2$. So

$$\mathcal{E}(A) = q\frac{I}{2}\mathrm{tr}(A) + (1-q)A$$

which for density matrices ρ simplifies to

$$\mathcal{E}(\rho) = q\frac{I}{2} + (1-q)\rho$$

For the second way to look at the depolarizing channel consider the function

$$\mathcal{D}(A) = \frac{IAI + XAX + YAY + ZAZ}{4}.$$

Notice that $\mathcal{D}(I) = I$ and $\mathcal{D}(X) = \mathcal{D}(Y) = \mathcal{D}(Z) = 0$ (to see the latter three notice that conjugating one of X, Y, Z by one of the other two non-identity Pauli matrices always results in the negative of the original). Now using the fact that $\mathcal{D}$ is linear and the fact that I, X, Y, Z are a basis for all 2×2 complex matrices with X, Y, Z all trace zero we get that $\mathcal{D}(A) = \mathrm{tr}(A)I/2$ for any A. From this we have that the depolarizing channel is given by

$$\begin{aligned}\mathcal{E}(\rho) &= q\frac{I\rho I + X\rho X + Y\rho Y + Z\rho Z}{4} + (1-q)\rho \\ &= \left(1 - \frac{3q}{4}\right)\rho + \frac{q}{4}(X\rho X + Y\rho Y + Z\rho Z).\end{aligned}$$

Defining $p = 3q/4$ we can also write this as

$$\mathcal{E}(\rho) = (1-p)\,\rho + \frac{p}{3}(X\rho X + Y\rho Y + Z\rho Z)$$

where now p is the chance of replacing ρ by the action of X, Y or Z on ρ (before q was the chance of replacing ρ by $I/2$). In what follows we will usually take this second viewpoint of the depolarizing channel and call p the *depolarizing strength*.

6 Stabilizer Codes

The stabilizer code formalism, first introduced by Gottesman in [Got97], is a way of describing quantum codes somewhat analogous to linear codes in the classical setting. Our discussion in this section mainly follows [NC] and, in parts, [MMM04].

6.1 Definitions and error correction

Recall the definitions of the Pauli matrices

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

They can be easily checked to satisfy the following relations:

$$X^2 = Y^2 = Z^2 = I,$$

$$XY = iZ \quad ZX = iY \quad YZ = iX,$$

$$YX = -iZ \quad XZ = -iY \quad ZY = -iX.$$

Let $U(\mathcal{H}_n)$ be the group of unitary operators on the space of n qubits. Define the *Pauli group* to be the group G_n inside $U(\mathcal{H}_n)$ generated by operators of the form $A_1 \otimes \cdots \otimes A_n$ where each $A_i \in \{I, X, Y, Z\}$. We will sometimes use notation where we omit the tensor signs or include only the non-identity operators. For example $XIY = X_1 Y_3 \in G_3$ is shorthand for $X \otimes I \otimes Y$.

Using this notation we have $X_1 Y_1 Z_1 = iI \in G_n$. Thus the group G_n must contain $\{\pm I, \pm iI\}$. But this is all we need: if $c, d \in \{\pm 1, \pm i\}$ and $A_i, B_i \in \{I, X, Y, Z\}$ then we have

$$\left(c \bigotimes_{i=1}^{n} A_i \right) \left(d \bigotimes_{i=1}^{n} B_i \right) = (cd) \bigotimes_{i=1}^{n} (A_i B_i)$$

and thus

$$G_n = \left\{ c \bigotimes_{i=1}^{n} A_i \colon c \in \{\pm 1, \pm i\}, A_i \in \{I, X, Y, Z\} \right\}.$$

We will call elements of the Pauli group *Pauli operators*. The following facts follows easily from our multiplication rule above and the relations among the Pauli matrices.

Proposition 6.1. *(i) Pauli operators commute iff they have an even number of places with different non-identity matrices. If they do not commute then they anti-commute.*

(ii) Squaring a Pauli operator gives $\pm I$

(iii) A Pauli operator $c \bigotimes_{i=1}^{n} A_i$ is Hermitian iff $c = \pm 1$.

For $S \leq G_n$ define $V_S \subseteq \mathcal{H}_n$ to be the set of vectors stabilized by S i.e.

$$V_S = \{|\psi\rangle \colon s|\psi\rangle = |\psi\rangle \text{ for all } s \in S\}.$$

It is easy to check that V_S is a subspace of $\mathcal{H}_n$ and that we can write

$$V_S = \cap_{s \in S} V_{\{s\}}.$$

Proposition 6.2. *The subspace $V_S \neq 0$ only if S is abelian and $-I \notin S$. In this case we also have $\pm iI \notin S$ and the relations $g^2 = I$ and $g^\dagger = g$ for all $g \in S$.*

Proof. If S is not abelian then there exists $M, N \in S$ such that $MN = -NM$. Then for any $|\psi\rangle \in V_S$ we have $|\psi\rangle = MN|\psi\rangle = -NM|\psi\rangle = -|\psi\rangle$ so $|\psi\rangle = 0$. Similarly, if $-I \in S$ then $|\psi\rangle = 0$ for all $|\psi\rangle \in V_S$.

If $\pm iI \in S$ then $(\pm iI)^2 = -I \in S$. Also for any $g \in S$ we have $g^2 = \pm I$. So $-I \notin S$ implies $\pm iI \notin S$ and $g^2 = I$ for all $g \in S$. Then $g^2 = I$ implies $g^\dagger = g$ because g is unitary. □

Define a *stabilizer group* to be a abelian subgroup $S \leq G_n$ with $-I \notin S$. The *stabilizer code* given by S is V_S.

Recall the definitions of the normalizers and centralizer of a subgroup: the normalizer of the stabilizer group S in G_n is

$$N(S) = \{E \in G_n \colon EgE^\dagger \in S \text{ for all } g \in S\}$$

and the centralizer is

$$C(S) = \{E \in G_n \colon EgE^\dagger = g \text{ for all } g \in S\}.$$

Clearly $C(S) \subseteq N(S)$ but in this case the inclusion is true in the opposite direction also: if $E \in N$ then $EgE^\dagger = \pm gEE^\dagger = \pm g$. So since $EgE^\dagger \in S$ we must have plus not minus and $EgE^\dagger = g$.

We will often work with generators for $S = \langle g_1, \ldots, g_m \rangle$. Note that E commutes with all elements of S iff it commutes with all the generators. To prove this: using the fact that the g_l commute and square to one, a general $g \in S$ can be written as $g = g_1^{\epsilon_1} \cdots g_m^{\epsilon_m}$ with $\epsilon_l \in \{0, 1\}$. Then if E commutes with each g_l it commutes with g.

We now describe the error correction process. Inspired by the depolarizing channel and similar channels our errors are elements of G_n. We start with an encoded state $|\psi\rangle$ then after error E occurs the system is in state $E|\psi\rangle$. If $S = \langle g_1, \ldots, g_m \rangle$ then the *syndrome* of an error operator E is

$$\beta = \beta(E) = (\beta_1 \ldots, \beta_m)$$

where $\beta_l \in \{0, 1\}$ is defined by the equation

$$Eg_l = (-1)^{\beta_l} g_l E.$$

Now the stabilizer generators g_l are commuting Hermitian operators so are observables that can be measured simultaneously. Each observable g_l has eigenvalues ± 1 (because $g_l^2 = I$) so can be decomposed as

$$g_l = (+1)\frac{I + g_l}{2} + (-1)\frac{I - g_l}{2}.$$

If β_l is the syndrome of g_l then measuring g_l gives result + with probability

$$\begin{aligned} p(+) &= \langle\psi|E^\dagger \left(\frac{I + g_l}{2}\right) E|\psi\rangle \\ &= \frac{1}{2}\langle\psi|\psi\rangle + \frac{1}{2}\langle\psi|E^\dagger g_l E|\psi\rangle \\ &= \frac{1}{2} + \frac{1}{2}(-1)^{\beta_l}\langle\psi|g_l E^\dagger E|\psi\rangle \\ &= \frac{1}{2} + \frac{1}{2}(-1)^{\beta_l}\langle\psi|\psi\rangle \\ &= \frac{1}{2} + \frac{1}{2}(-1)^{\beta_l} \\ &= 1 - \beta_l. \end{aligned}$$

Thus the outcome of the measurement is deterministic and depends only on the syndrome of the error, not the state $|\psi\rangle$.

If we have a collection of errors with distinct syndromes then we can correct the errors. In fact more as true as we will now see.

Theorem 6.3. *Suppose $\{E_j\}$ is a set of error operators such that $E_j^\dagger E_k \notin N(S) \setminus S$ for all j, k. Then $\{E_j\}$ is correctable.*

Proof. First note that E having syndrome β means $Eg_l = (-1)^{\beta_l} g_l E$ which can be rewritten as $Eg_l E^\dagger = (-1)^{\beta_l} g_l$ which can be rewritten as $E^\dagger g_l E = (-1)^{\beta_l} g_l$. So in particular $E^\dagger$ has the same syndrome as E. Now we claim that errors E_j and E_k having the same syndrome is equivalent to $E_j^\dagger E_k \in N(S)$. First, if E_j and E_k have the same syndrome then for each g_l we have

$$\begin{aligned} E_j^\dagger E_k g_l &= E_j^\dagger (-1)^{\beta_l} g_l E_k \\ &= ((-1)^{\beta_l})^2 g_l E_j^\dagger E_k \\ &= g_l E_j^\dagger E_k \end{aligned}$$

so $E_j^\dagger E_k \in N(S)$. Conversely, if $E_j^\dagger E_k \in N(S)$ then $E_j^\dagger E_k g_l = g_l E_j^\dagger E_k$. So if $E_k g_l E_k^\dagger = (-1)^{\beta_l} g_l$ then

$$\begin{aligned} E_j^\dagger g_l E_j &= E_j^\dagger \left((-1)^{\beta_l} E_k g_l E_k^\dagger\right) E_j \\ &= (-1)^{\beta_l} E_j^\dagger E_k g_l E_k^\dagger E_j \\ &= (-1)^{\beta_l} g_l E_j^\dagger E_k E_k^\dagger E_j \\ &= (-1)^{\beta_l} g_l. \end{aligned}$$

So the theorem to be proved can be restated as: a set of errors is correctable if errors with the same syndrome differ by an element of the stabilizer.

If the syndrome corresponding to E_j is unique then we can correct the error by applying the operation $E_j^\dagger$, so the state becomes $E_j^\dagger E_j|\psi\rangle = |\psi\rangle$. If we have two errors E_j and E_k with the same syndrome then, by assumption in the theorem, $E_j^\dagger E_k \in S$. Thus even if we use the 'wrong' operator, $E_j^\dagger$ instead of $E_k^\dagger$, to correct we still have $E_j^\dagger E_k|\psi\rangle = |\psi\rangle$. □

Define the *distance* of a stabilizer code to be the minimum weight (number of non-identity components) of a Pauli operator in $N(S) \setminus S$. Then by the theorem above a distance d stabilizer code can fix errors on any $\lfloor (d-1)/2 \rfloor$ qubits. We will say V_S is an $[n, k, d]$ quantum code if V_S is a 2^k dimensional subspace of $\mathcal{H}_n$ with distance d.

We will say a code *nondegenerately corrects* a set of errors $\{E_j\}$ if each of the errors has a unique syndrome. If the code can correct the set of errors but they do not have unique syndromes then we say that the code *degenerately corrects* the errors.

We will say an $[n, k, d]$ quantum code is *nondegenerate* if it nondegenerately corrects the set of errors of weight less than $\lfloor (d-1)/2 \rfloor$. Otherwise it is *degenerate*.

6.2 Quantum circuit for decoding

We follow the extended version of [MMM04] for this section. To create a quantum circuit for decoding we need an extra *ancilla* qubit, initialized to $|0\rangle$, for each stabilizer generator. Let g_l^c be the controlled g_l operator given by

$$g_l^c(|0\rangle|\psi\rangle) = |0\rangle|\psi\rangle, \quad g_l^c(|1\rangle|\psi\rangle) = |1\rangle g_l|\psi\rangle.$$

If the state of the system after an error occurs is $|\psi\rangle$ then we apply the Hadamard gate H to the ancilla, apply g_l^c to the composite state, then apply H to the ancilla before measuring it in the computational basis. This process has the result

$$\begin{aligned}
(H \otimes I) g_l^c (H \otimes I)(|0\rangle|\psi\rangle) &= (H \otimes I) g_l^c \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} |\psi\rangle \right) \\
&= (H \otimes I) \frac{|0\rangle|\psi\rangle + |1\rangle g_l|\psi\rangle}{\sqrt{2}} \\
&= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} |\psi\rangle + \frac{|0\rangle - |1\rangle}{\sqrt{2}} g_l|\psi\rangle \right) / \sqrt{2} \\
&= |0\rangle \left(\frac{|\psi\rangle + g_l|\psi\rangle}{2} \right) + |1\rangle \left(\frac{|\psi\rangle - g_l|\psi\rangle}{2} \right) \\
&= |\beta_l\rangle|\psi\rangle
\end{aligned}$$

where the last equality can be seen by considering the two cases $\beta_l = 0, 1$. Thus upon measuring the ancilla qubit we recover the syndrome and have not

affected the state at all. For creating a quantum circuit for decoding note that the operator g_l^c can be implemented using controlled Pauli gates.

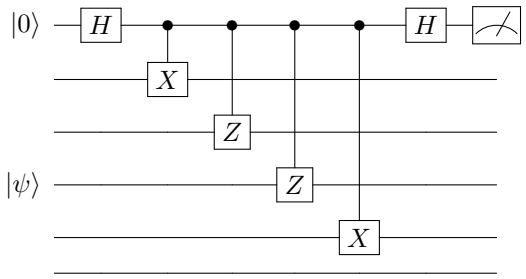


Figure 2: The quantum circuit for measuring a stabilizer generator $g_1 = XZZXI$ where $|\psi\rangle$ is the state of 5 qubits.

6.3 Check matrix representation

Define a function $r\colon G_n \to \mathbb{F}_2^{2n}$ by writing g as a string of X's times a string of Z's, forgetting the constant out the front, and then writing those strings in binary. For example $g = XIY = i(XIX)(IIZ)$ and $r(g) = 101001$.

Notice $r(gg') = r(g) + r(g')$ and that $\ker r = \{\pm I, \pm iI\}$. Also for any string in $\mathbb{F}_2^{2n}$ we can find an operator that maps to it. Thus $G_n/\{\pm I, \pm iI\} \cong \mathbb{F}_2^{2n}$ (an isomorphism of a multiplicative group with the additive group of $\mathbb{F}_2^{2n}$). Define the *effective Pauli group* to be $\overline{G}_n = G_n/\{\pm I, \pm iI\}$. We will choose coset representatives of $\overline{G}_n$ of the form

$$\bigotimes_{i=1}^{n} A_i$$

where each $A_i \in \{I, X, Y, Z\}$,

Now if S is a stabilizer group then since $-I \notin S$ we have no pairs of elements differing by $-I$ or $\pm iI$ so $|S| = |r(S)|$.

Define

$$\Lambda = \begin{bmatrix} 0 & I_n \\ I_n & 0 \end{bmatrix}.$$

Now write $A = \begin{bmatrix} x(g) & z(g) \end{bmatrix}$ and define the *twisted product*

$$r(g) \odot r(g') = r(g)\Lambda r(g')^T = x(g) \cdot z(g') + z(g) \cdot x(g').$$

Then we claim that g and g' commute iff $r(g) \odot r(g') = 0$. To see this, note that wherever g and g' have different non-identity matrices in a position, we get exactly one one in the sum for the twisted product.

The generators for $S = \langle g_1, \ldots, g_m \rangle$ are said to be *irredundant* if removing any generator makes the group smaller.

Proposition 6.4. *The generators g_l of a stabilizer group are irredundant iff the collection of $r(g_l)$ are linearly independent.*

Proof. The rows are linearly independent over $\mathbb{F}_2$ iff $\sum a_i r(g_i) = 0$ with $a_j = 1$ for some j. But $\sum a_i r(g_i) = 0$ iff $\prod g_i^{a_i} \in \{\pm I, \pm iI\}$. But this can only be I because it is in S and $-I \notin S$. So some $a_j = 1$ iff $g_j = \prod_{i \neq j} g_i^{a_i}$ i.e. g_j is dependent on the other generators. □

So, if we create a matrix of rows which have twisted product zero and which are linearly independent then that defines a stabilizer group (choose the g_i corresponding to the r_i to be the one with $c = 1$). If we write the matrix as

$$A = \begin{bmatrix} A_1 & A_2 \end{bmatrix}$$

then the condition that all pairs of rows have twisted product zero becomes $A_1 A_2^T + A_2 A_1^T = 0$.

To decode in the check matrix representation we need to find the syndrome of an error E. As discussed earlier this depends on whether E commutes with the generators g_l. Thus we need to take the twisted product of rows of the check matrix A with $r(E)$. In practice if we write $\begin{bmatrix} z(E) & x(E) \end{bmatrix}$ instead of $r(E) = \begin{bmatrix} x(E) & z(E) \end{bmatrix}$ then we can instead use the normal $\mathbb{F}_2$ matrix product. Once again, if the error has unique syndrome then it can be decoded and also if errors with the same syndrome differ by elements of the stabilizer they can be corrected.

If we consider A as a classical binary parity check matrix then elements of the stabilizer (the row space of A) are precisely elements of the dual code of the code described by A.

6.4 $\mathbb{F}_4$ representation

Instead of using binary check matrices we can use matrices over $\mathbb{F}_4$ which are half as wide. A vector $[x, z] \in \mathbb{F}_2^{2n}$ corresponds to a vector $(x + \omega z) \in \mathbb{F}_4^n$. Alternatively we can go straight from $\overline{G}_n$ to $\mathbb{F}_4^n$ by the map

$$I \mapsto 0, \quad X \mapsto 1, \quad Y \mapsto \omega^2, \quad Z \mapsto \omega.$$

For this $\mathbb{F}_4$ representation two generators of the stabilizer group g_m and g'_m commute iff

$$\mathrm{tr}(\overline{r_m} \cdot r_{m'}) = 0$$

where $r_m, r_{m'}$ are the corresponding rows of the $\mathbb{F}_4$ check matrix. To see this notice that different non-identity Pauli matrices correspond to different non-zero elements of $\mathbb{F}_4$ and for $x, y \in \mathbb{F}_4$ we have $\mathrm{tr}(\overline{x}y) = 0$ iff $x = y$ or $x = 0$ or $y = 0$. So

$$\mathrm{tr}(\overline{r_m} \cdot r_{m'}) = \sum_i \mathrm{tr}(\overline{r_{m,i}} r_{m',i})$$

counts the parity of the number of times the two stabilizer generators overlap in different non-identity Pauli matrices.

Thus we can do decoding with matrices and vectors of $\mathbb{F}_4$ but our matrix-vector product is actually carried out via the trace Hermitian form.

6.5 CSS codes

A Calderbank-Shor-Steane (CSS) code is a stabilizer code built out of two classical codes. The following proposition, allowing not necessarily full rank parity check matrices, is stated in [TZ09].

Proposition 6.5. *Assume that (not necessarily full rank) parity check matrices H_X and H_Z define classical $[n, k_1, d_1]$ and $[n, k_2, d_2]$ binary linear codes C_X and C_Z and that all rows of H_X are orthogonal to all rows of H_Z (this is equivalent to $C_X^\perp \subseteq C_Z$ which is equivalent to $C_Z^\perp \subseteq C_X$). Then $CSS(C_X, C_Z^\perp)$, the stabilizer code with binary check matrix*

$$A = \begin{bmatrix} H_X & 0 \\ 0 & H_Z \end{bmatrix},$$

is a quantum $[n, k, d]$ code where

$$k = n - \dim(C_X^\perp) - \dim(C_Z^\perp)$$

and

$$d = \min\{\mathrm{wt}(c) \colon c \in (C_Z \setminus C_X^\perp) \cup (C_X \setminus C_Z^\perp)\}.$$

In particular $d \geq \min(d_1, d_2)$.

Proof. Each independent row of H_X and H_Z halves the dimension of the code subspace so recalling that $C_X^\perp$ is the rowspace of H_X the formula for k follows. Next, recall that the distance of a quantum stabilizer code is

$$d = \min\{\mathrm{wt}(c) \colon c \in N(S) \setminus S\}$$

where

$$N(S) = C_{G_n}(S) = \{g \in G_n \colon gs = sg \text{ for all } s \in S\}.$$

Notice that

$$S = \langle X^a, Z^b \colon a \text{ is a row of } H_X, b \text{ is a row of } H_Z \rangle = \{X^a Z^b \colon a \in C_X^\perp, b \in C_Z^\perp\}$$

where X^a is notation for $X_1^{a_1} \cdots X_n^{a_n}$. Then (ignoring global phase factors)

$$N(S) = \{X^a Z^b \colon a \in C_Z, b \in C_X\}.$$

There exists a minimum weight vector in $N(S) \setminus S$ of the form X^a or Z^b so the formula in the proposition follows. For the final result note that if c is the codeword of minimum weight then $c \in C_X$ or $c \in C_Z$ so $d = \mathrm{wt}(c) \geq \min(d_1, d_2)$. □

6.6 Dual-containing codes

Following [MMM04], call a classical code C *dual-containing* if $C^\perp \subseteq C$ (this is equivalent to $C^\perp$ being weakly self-dual). Using the CSS construction with $C_X = C_Z = C$, where C is a length n dual-containing classical binary code, we have a check matrix

$$\begin{bmatrix} H(C) & 0 \\ 0 & H(C) \end{bmatrix}.$$

defining a $[n, k, d]$ stabilizer code with

$$k = n - 2\,\dim(C^\perp)$$

and

$$d = \min\{\mathrm{wt}(c) \colon c \in C \setminus C^\perp\}.$$

6.7 Examples of stabilizer codes

(i) The 5 qubit code:

The code is $S = \langle XZZXI, IXZZX, XIXZZ, ZXIXZ\rangle$ with binary check matrix

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and $\mathbb{F}_4$ check matrix

$$\begin{bmatrix} 1 & \omega & \omega & 1 & 0 \\ 0 & 1 & \omega & \omega & 1 \\ 1 & 0 & 1 & \omega & \omega \\ \omega & 1 & 0 & 1 & \omega \end{bmatrix}.$$

This code can correct the error set $\{I, X_i, Y_i, Z_i \colon i = 1, \ldots, 5\}$ because each of these has a unique syndrome.

(ii) Steane's 7 qubit code:

We begin with the $[7, 4]_2$ Hamming code C specified by the parity check matrix

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

This code can correct one error because the syndromes of the 8 weight-zero and weight-one errors are distinct. The code is also dual-containing: the codewords in $C^\perp$ are linear combinations of rows of H but the rows of H are all orthogonal to all rows of H. So we can use it to create a $[7, 1]$ quantum stabilizer code that can correct errors in one qubit.

7 Quantum LDPC codes

One of the first papers to suggest using Low Density Parity Check (LDPC) codes to construct quantum error correcting codes was [MMM04]. This paper focuses on finding regular dual-containing low-density parity-check codes i.e. dual-containing codes specified by $M \times N$ binary matrices H where every row has weight k and every column has weight j.

The dual-containing condition is equivalent to $HH^T = 0$ which is equivalent to every pair of rows of H having even overlap (including a row with itself i.e. k is even).

7.1 Bicycle codes

One method used to find such an H is Construction B from [MMM04], where B is a mnemonic for bicycle. The algorithm (with some choices left undetermined) follows. This construction does not give regular codes: the column weight is only approximately equal to $j = Mk/N$.

Algorithm 1 ConstructBicycleCode(M,N,k)

Require: M, N, k positive integers with N, k even
Ensure: $M \times N$ matrix H with k ones per row

1: Construct a random $N/2 \times N/2$ circulant matrix C with row weight $k/2$
2: Define $H_0 = \begin{bmatrix} C & C^T \end{bmatrix}$
3: Delete $N/2 - M$ rows of H_0, attempting to keep the column weights uniform
4: **return** H, the matrix left after deleting rows of H_0

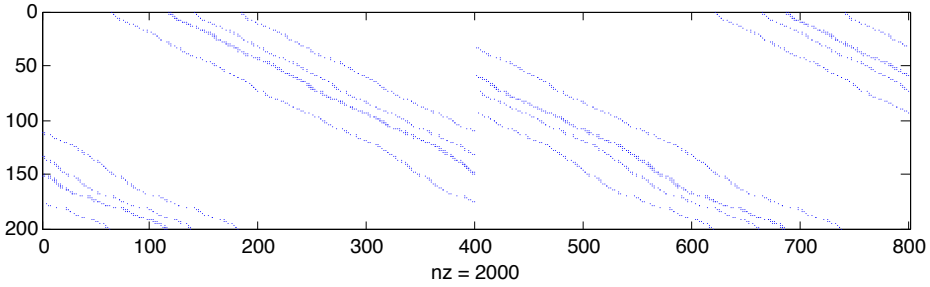


Figure 3: A bicycle code matrix H with $M = 200$, $N = 800$, $k = 10$. Each dot represents a 1.

To see that H satisfies the required overlap condition notice that

$$H_0 H_0^T = \begin{bmatrix} C & C^T \end{bmatrix} \begin{bmatrix} C^T \\ C \end{bmatrix} = CC^T + C^T C = CC^T + CC^T = 0$$

where C and C^T commute because they are both circulant matrices. Then since the rows of H are a subset of the rows of H_0 they must be orthogonal also.

For the CSS code constructed from H with

$$A = \begin{bmatrix} H & 0 \\ 0 & H \end{bmatrix}$$

we have d given by the minimum weight of codewords in $C \setminus C^{\perp}$. So if the row deletion process deletes any row of H_0 independent of the remaining rows then since that row has weight k and is in $C \setminus C^{\perp}$ we have $d \leq k$.

7.2 Tanner graphs for stabilizer codes

We represent a stabilizer code by an edge-labeled bipartite graph, called a Tanner graph. This graph has vertices $V = Q \cup C$ where Q represents the qubits and C represents the stabilizer generators, or checks. The graph has an edge between q and c if the stabilizer represented by c acts non-trivially on the qubit represented by q (i.e. the qth component of the cth stabilizer differs from the identity). In the Tanner graph qubits (or variable nodes) are represented by circles and stabilizer generators (or check nodes) are represented by squares. The edges are labeled by how the check acts on the variable.

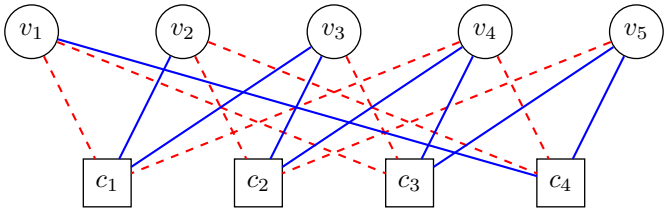


Figure 4: The Tanner graph for the 5 qubit code. Dashed red lines represent 1 and solid blue lines represent ω.

Notice that if checks c and c' both act non-trivially on at least two qubits in common, say v and v', then there exists a 4-cycle (c, v, c', v') in the Tanner graph. We now show that a stabilizer code with distance greater than 1 must contain a 4-cycle, as pointed out in [MMM04].

First note that we may assume that the stabilizer does not include any weight one operators. If it did then that operator $A_q \in \{X, Y, Z\}$ would ensure that all states in the codespace had qth component equal to the $+1$ eigenvector of A_q. Also each of the stabilizer generators would have I or A in the qth component. Then the stabilizer group generated by the old generators with the qth component removed generates the same codespace, as long as we tensor everything with the $+1$ eigenvector.

Assume that our stabilizer code does not contain a 4-cycle. Thus all pairs of checks act on at most one common qubit. Consider a qubit v. If it has two checks c, c' acting on it then v is the only qubit that c and c' have in common

and thus for the stabilizer generators to commute they must act in the same way. Thus all the edges connecting to v are labelled the same. So some $A_q \in N(S)$ and as argued above $A_q \notin S$ so the distance of the code is one.

7.3 Decoding QLDPC codes

The Belief Propagation or Sum Product Algorithm is a general algorithm to carry out various marginalization processes. For a derivation given in more of the sum product philosophy see [PC08]. There the algorithm is described as a way to maximize a certain marginal function and justified by the fact that when the Tanner graph is a tree the messages converge to the correct marginals. Of course, as we have seen the Tanner graph for a reasonable code has cycles, but the example of classical LDPC codes tells us this is not necessarily a fatal problem. We give a slightly different derivation more in the spirit of the name Belief Propagation.

Algorithm 2 F4BeliefPropagation(A,s,maxIter)

Require: $\mathbb{F}_4$-check matrix A, syndrome vector s, positive integer maxIter
Ensure: Output estimated error E and flag for whether decoding was successful

Set $p_q^{\text{ch}} = (1-p, p/3, p/3, p/3)$
Set $p_{q\to c}^{(0)}(b) = p_q^{\text{ch}}(b)$
for $i = 1$ to $i =$maxIter **do**
  for all q and $c \in n(q)$ **do**
    Set $p_{c\to q}^{(i)}(b) = \sum_{E \text{ with } E_q = b} \mathbb{1}\left[\sum_{q' \in n(c)} \text{tr}(\overline{a_{cq'}} E_{q'}) = s_c\right] \prod_{q' \in n(c)\setminus q} p_{q'\to c}^{(i-1)}(E_{q'})$
  end for
  for all c and $q \in n(c)$ **do**
    Set $p_{q\to c}^{(i)}(b) \propto p_q^{\text{ch}}(b) \prod_{c' \in n(q)\setminus c} p_{c'\to q}^{(i)}(b)$
  end for
  for all q **do**
    Let $p_q^{(i)}(b) \propto p_q^{\text{ch}}(b) \prod_{c' \in n(q)} p_{c'\to q}^{(i)}(b)$
  end for
  Let $E = (E_q^*)$ where $E_q^* = \text{argmax}_{b\in\mathbb{F}_4} p_q^{(i)}(b)$
  if $AE = 0$ **then**
    return E and flag successful decoding
  end if
end for
return E and flag unsuccessful decoding

We use Belief Propagation in its syndrome decoding form. This is an iter-

ative message-passing algorithm which sends probabilities representing beliefs that a variable should be a certain value or a check should be satisfied between checks and variables. The messages are probability distributions i.e. functions $\mathbb{F}_4 \to [0,1]$ sent along edges of the Tanner graph. We now give a derivation of what these messages should be.

As discussed earlier, after an error E from the depolarizing channel we can measure the syndromes of the stabilizer generators. We consider the error E to be in $\mathbb{F}_4^n$ and then the syndrome is given by

$$s_c = \sum_q \operatorname{tr}\left(\overline{a_{cq}} E_q\right).$$

Let

$$P(E) = \prod_q P(E_q)$$

where $P(E_q)$ is defined by the memoryless Pauli channel. Define $P(E \mid s)$ by only allowing the syndromes $s = (s_c)$ which satisfy the commutation conditions

$$P(E \mid s) = K_s \, P(E) \prod_c \mathbb{1}\left[\sum_{q' \in n(c)} \operatorname{tr}(\overline{a_{cq'}} E_{q'}) = s_c\right]$$

where the normalizing constant K_s is chosen such that

$$\sum_{E \in \mathbb{F}_4^n} P(E \mid s) = 1.$$

First the variable nodes send their prior probability of error, given by the channel model. So for the depolarizing channel we have

$$p^{(0)}_{q \to c}(b) = p_q^{\mathrm{ch}}(b) = \begin{cases} 1-p & \text{if } b = 0 \\ p/3 & \text{if } b = 1 \\ p/3 & \text{if } b = \omega \\ p/3 & \text{if } b = \omega^2. \end{cases}$$

Let $n(c)$ be the set of variable nodes connected to check node c, let $n(c) \setminus q$ be shorthand for $n(c) \setminus \{q\}$ and make similar definitions for $n(q)$ and $n(q) \setminus c$. For integer $i \geq 1$, a check node c, receiving beliefs from connected variable nodes $n(c)$ about which error $E_{q'} \in \mathbb{F}_4$ they are, calculates its updated message to a variable node q about whether check c is satisfied using only the messages from other variable nodes $n(c) \setminus q$ (assumed to be conditionally independent) and the fixed syndrome $s = (s_c)$:

$$
\begin{aligned}
p^{(i)}_{c\to q}(b) &= P(\text{check } c \text{ is satisfied} \mid E_q = b, s) \\
&= P\left(\sum_{q'\in n(c)} \operatorname{tr}(\overline{a_{cq'}} E_{q'}) = s_c \;\middle|\; E_q = b, s \right) \\
&= \sum_{E \text{ with } E_q = b} P\left(\sum_{q'\in n(c)} \operatorname{tr}(\overline{a_{cq'}} E_{q'}) = s_c \right) P(E \mid s) \\
&= \sum_{E \text{ with } E_q = b} \mathbb{1}\left[\sum_{q'\in n(c)} \operatorname{tr}(\overline{a_{cq'}} E_{q'}) = s_c \right] P(E \mid s) \\
&= \sum_{E \text{ with } E_q = b} \mathbb{1}\left[\sum_{q'\in n(c)} \operatorname{tr}(\overline{a_{cq'}} E_{q'}) = s_c \right] \prod_{q'\in n(c)\setminus q} P(E_{q'} \mid s) \\
&= \sum_{E \text{ with } E_q = b} \mathbb{1}\left[\sum_{q'\in n(c)} \operatorname{tr}(\overline{a_{cq'}} E_{q'}) = s_c \right] \prod_{q'\in n(c)\setminus q} p^{(i-1)}_{q'\to c}(E_{q'}).
\end{aligned}
$$

Then each variable node sends to each check node its updated belief about its value relying on the independent beliefs of the other check nodes and the channel information.

$$
\begin{aligned}
p^{(i)}_{q\to c}(b) &= P(E_q = b \mid \text{checks } c' \in n(q) \setminus c \text{ satisfied}) \\
&= \frac{P(\text{checks } c' \in n(q) \setminus c \text{ satisfied} \mid E_q = b) P(E_q = b)}{P(\text{checks } c' \in n(q) \setminus c \text{ satisfied})} \\
&= \frac{P(E_q = b)}{P(\text{checks } c' \in n(q) \setminus c \text{ satisfied})} \prod_{c'\in n(q)\setminus c} P(c' \text{ satisfied} \mid E_q = b) \\
&= K_{qc}\, p^{\mathrm{ch}}_q(b) \prod_{c'\in n(q)\setminus c} p^{(i)}_{c'\to q}(b),
\end{aligned}
$$

where the constant

$$
K_{qc} = \frac{1}{P(\text{checks } c' \in n(q) \setminus c \text{ satisfied})}
$$

can be found by insisting that

$$
\sum_{b\in\mathbb{F}_4} p^{(i)}_{q\to c}(b) = 1.
$$

For our posterior probability distribution at step i we take

$$
\begin{aligned}
p^{(i)}_q(b) &= P(E_q = b \mid \text{checks } c' \in n(q) \text{ satisfied}) \\
&= K_q\, p^{\mathrm{ch}}_q(b) \prod_{c'\in n(q)} p^{(i)}_{c'\to q}(b),
\end{aligned}
$$

where again K_q is chosen to normalize the probability distribution. Then the qubit-wise estimate for E_q is $E_q^* = \text{argmax}_{b \in \mathbb{F}_4} p_q^{(i)}(b)$.

We summarize this process in Algorithm 2. This algorithm can be made more efficient but we give a simple version. Appendix B details a more numerically stable version of the algorithm and we give more optimizations there.

7.4 Binary decoding

A slightly less complicated decoding algorithm, more easily compared to classical binary codes, works at the cost of ignoring the correlations between X and Z errors from the depolarizing channel. This is the approach taken in most simulations in [MMM04].

More precisely, we use a binary check matrix A for our quantum LDPC code. If $E_{\mathbb{F}_4} \in \mathbb{F}_4^N$ is an error on N qubits from the depolarizing channel then in binary we can write $r(E_{\mathbb{F}_4}) = [E_X, E_Z]$. As discussed in the section on binary check matrices if we instead write $E = [E_Z, E_X] \in \mathbb{F}_2^{2N}$ then the syndrome can be calculated using standard $\mathbb{F}_2$ matrix-vector multiplication with the formula $s = AE$.

So if $E_{\mathbb{F}_4}$ is coming from a depolarizing channel of strength p then the probability of each bit of E being 1 is given by $2p/3$. However the X and Z errors are correlated: whether E_i is 1 is not independent of whether $E_{(i+N) \bmod 2N}$ is 1. For binary decoding we ignore this correlation and thus can think of our task as classical syndrome decoding over a BSC.

An almost identical analysis as for the $\mathbb{F}_4$ decoding earlier gives a similar algorithm so we describe only the alterations necessary:

(i) Now b varies over $\mathbb{F}_2$ instead of $\mathbb{F}_4$.

(ii) The channel probabilities are $p_q^{\text{ch}} = (1 - 2p/3, 2p/3)$.

(iii) The message sent from check to qubit is

$$p_{c \to q}^{(i)}(b) = \sum_{E \text{ with } E_q = b} \mathbb{1}\left[A_c \cdot E = s_c\right] \prod_{q' \in n(c) \setminus q} p_{q' \to c}^{(i-1)}(E_{q'}).$$

7.5 Performance of QLDPC codes

To discuss the performance of our codes on the depolarizing channel, the natural approach is to try to define and find the quantum capacity of the channel. This has been attempted and, although finding a simple formula for the capacity is still an open problem, could be used to bound our results. See page 74 onwards of Chapter 7 of [Pre98] for a slightly non-rigorous exposition of known results on the quantum channel capacity of the depolarizing channel. We will take a different approach, referring only to capacities of classical channels, following [MMM04].

We can simulate the performance of bicycle codes by, for different depolarizing strengths, creating a large number of random errors and then finding

the proportion of those that the belief propagation decoder fails to determine correctly. Note that in these simulations we are ignoring the possibility that a wrongly identified error may still correct the actual error. Practically, at least, this is a reasonable assumption: the belief propagation decoder tries to identify the most likely error and in the simulations I have run all the failures of the decoder to do this are detected failures. That is, the decoder does not converge to any error with the correct syndrome. Thus the possibility of an error with the correct syndrome differing from the correct error by an element of the stabilizer does not occur.

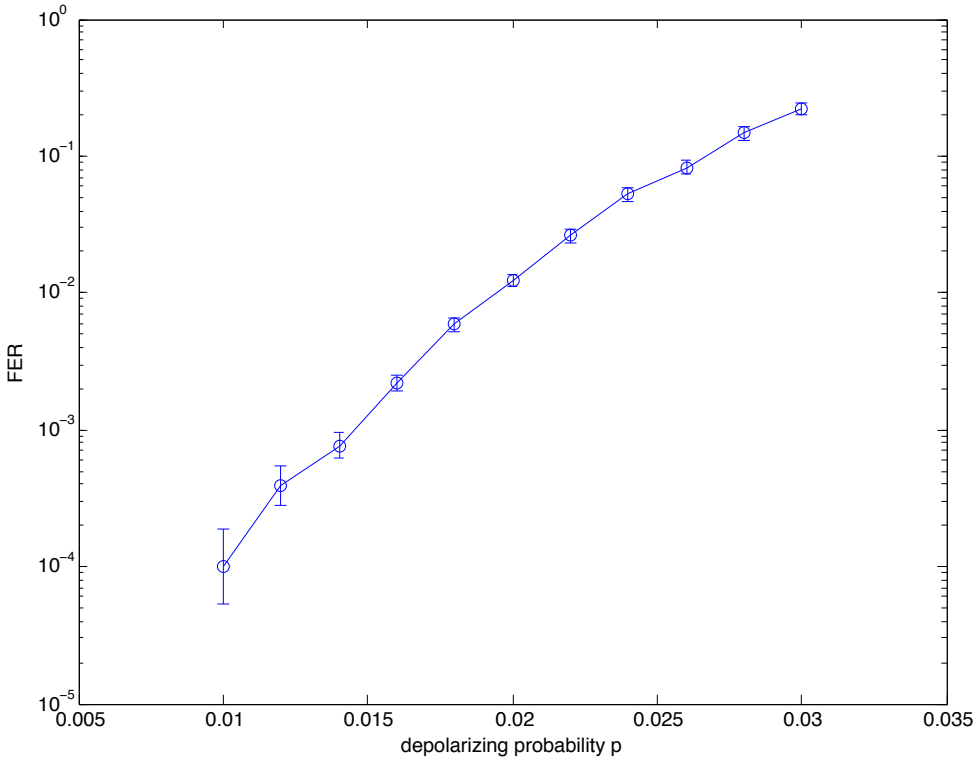


Figure 5: Probability of belief propagation decoder failure for bicycle code with $M = 200$, $N = 800$, $k = 10$ and the decoder using up to 90 iterations. The error bars are two-sigma confidence intervals. The results of this experiment are very similar to the identical experiment seen in Figure 3 of [PC08]

With the assumption that we are not taking advantage of any nondegeneracy we can bound the rates of our quantum codes by considering the capacities of related classical channels.

Proposition 7.1. *Using stabilizer codes the rate at which we can reliably de-*

termine errors from their syndromes is upper bounded by

$$R_4^Q = 1 - h_2(p) - p \log 3.$$

For stabilizer codes using binary decoding (i.e. ignoring the correlations between X and Z errors) the rate at which we can reliably determine errors from their syndromes is upper bounded by

$$R_{BSC}^Q = 1 - 2h_2(2p/3).$$

If we restrict ourselves to using bounded distance decoders separately on the X and Z errors then we can achieve rates up to

$$R_{GV}^Q = 1 - 2h_2(4p/3).$$

Proof. We use a full-rank $M \times 2N$ binary check matrix A. This gives quantum rate $(N - M)/N = 1 - M/N$. For errors E' coming from the depolarizing channel acting on N qubits recall that if $r(E') = [E_X, E_Z]$ we will write this as $E = [E_Z, E_X]$ and then we can find syndrome $s \in \mathbb{F}_2^M$ by the $\mathbb{F}_2$ matrix-vector product

$$s = AE.$$

Errors coming from the depolarizing channel then become errors from the 4-ary symmetric channel acting on elements of $\mathbb{F}_2^2$. So if we could find E from s reliably (i.e. with probability of failure less than ϵ) then we could communicate reliably over the 4-ary symmetric channel using syndrome decoding. More precisely, let

$$C = \{x \in \mathbb{F}_2^{2N} \mid Ax = 0\}$$

i.e. C is the code with parity check matrix A. Then C is a $[2N, 2N - M]_2$ classical code. To use it to communicate we send x, receive $x + E$, calculate syndrome $s = A(x + E) = AE$. Then by assumption we can recover E reliably from s and thus recover $x = (x + E) - E$. So, by the calculation of the capacity of the 4-ary symmetric channel we have

$$\frac{2N - M}{2N} \frac{\text{inf. bits}}{\text{codebits}} = \frac{2N - M}{N} \frac{\text{inf. bits}}{\text{ch. use}} < 2 - h_2(p) - p \log 3$$

which gives

$$R = 1 - \frac{M}{N} = \frac{2N - M}{N} - 1 < 1 - h_2(p) - p \log 3.$$

If we ignore the correlations between X and Z errors then being able to reliably determine errors from their syndromes implies that we can communicate over the BSC with crossover probability $2p/3$. So we must have

$$\frac{2N - M}{2N} < 1 - h_2\left(\frac{2p}{3}\right)$$

and thus the quantum rate is

$$R = 1 - \frac{M}{N} = 2\left(\frac{2N - M}{2N}\right) - 1 < 1 - 2h_2\left(\frac{2p}{3}\right).$$

Similarly, the Gilbert-Varshamov rate for the BSC gives quantum rate

$$R < 1 - 2h_2\left(\frac{4p}{3}\right).$$

□

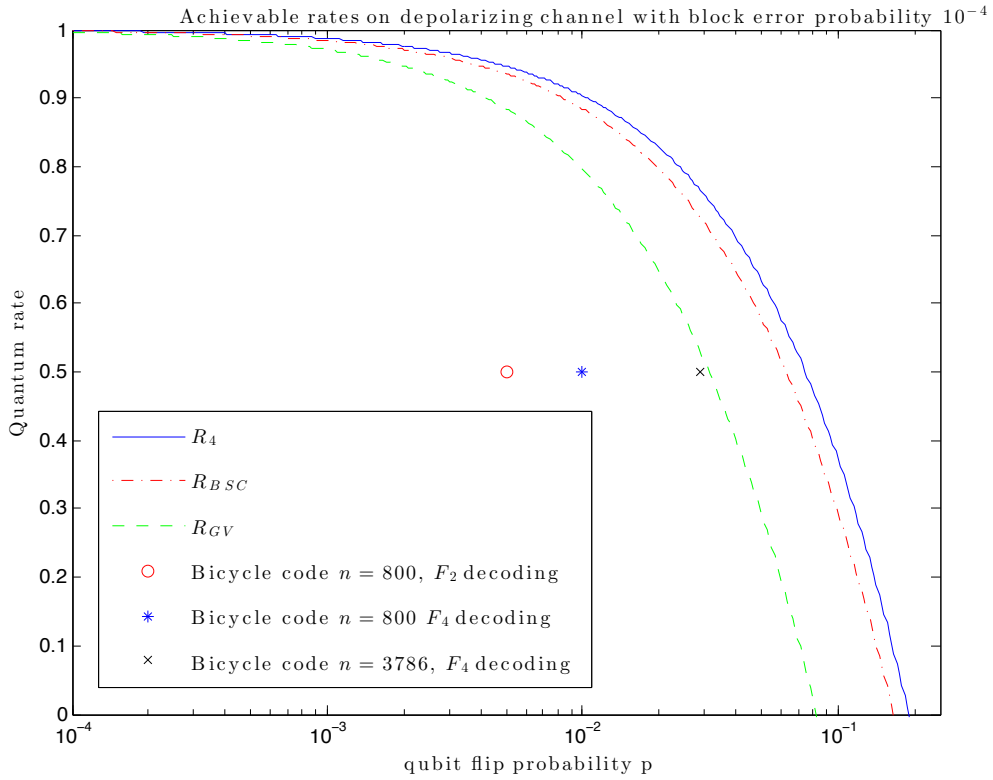


Figure 6: The bounds on performance of QLDPC codes and the simulated performance of some bicycle codes. As discussed in the body of the text $R_4(p) = 1 - h_2(p) - p \log_2 3$, $R_{BSC}(p) = 1 - 2h_2(2p/3)$ and $R_{GV}(p) = 1 - 2h_2(4p/3)$. The points represent code/decoder pairs and are placed with the largest depolarizing strength such that the probability of decoder failure is less than 10^{-4}. A similar diagram is given in [MMM04]. It should be noted that the horizontal axis in that diagram is f instead of p, where $f = 2p/3$. Also, that paper has a length 3786 bicycle code slightly to the right of the Gilbert-Varshamov bound rather than slightly to the left. This small difference may be explainable by differences in code construction or decoder implementation.

A Linear Algebra

A.1 Symmetric bilinear forms

Let V be a finite dimensional vector space over a field K. A map

$$B\colon V \times V \to K$$

is a *symmetric bilinear form* if for all $u, v, w \in V$ and $\lambda \in K$ we have

(i) $B(u+v, w) = B(u, w) + B(v, w)$,

(ii) $B(\lambda u, v) = \lambda B(u, v)$ and

(iii) $B(u, v) = B(v, u)$.

Consider the map

$$B'\colon V \to V^*$$

given by

$$v \mapsto (w \mapsto B(v, w)).$$

We say B is *nondegenerate* if B' is an isomorphism i.e. if $B(v, w) = 0$ for all $w \in V$ implies $v = 0$ (this is really the condition for B' to be injective but the dimensions of V and V^* are equal so this is sufficient for isomorphism).

Let

$$W^\perp = \{v\colon B(v, w) = 0 \text{ for all } w \in W\}.$$

We can think of $W^\perp$ as being the kernel of the map

$$B'_W\colon V \to W^*$$

given by

$$v \mapsto (w \mapsto B(v, w))$$

i.e. $B'_W(v)$ is just $B'(v)$ restricted to W.

Now if B is nondegenerate the image of B' is V^* so the image of B'_W is W^*. So by the rank nullity theorem we get

$$\dim(\mathrm{im}(B'_W)) + \dim(\ker(B'_W)) = \dim(V)$$

which using $\dim(W^*) = \dim(W)$ we can see becomes

$$\dim(W) + \dim(W^\perp) = \dim(V).$$

A.2 Linear algebra over $\mathbb{C}$

We use Dirac bra-ket notation: consider a finite dimensional complex vector space with basis $|v_i\rangle$ and dual basis $\langle v_i|$. The inner product of $|v\rangle$ with $|w\rangle$ (in that order) will be written $\langle v|w\rangle$.

If V and W are finite dimensional $\mathbb{C}$-vector spaces with bases $|v_i\rangle$ and $|w_i\rangle$ and $A\colon V \to W$ is a linear operator then using the completeness relation $I_V = \sum_i |v_i\rangle\langle v_i|$ and similar for I_W we can write

$$A = \sum_{ij} \langle w_j|A|v_i\rangle |w_j\rangle\langle v_i|.$$

If A is a linear operator on V then there exists the adjoint $A^\dagger$ with inner product of $|v\rangle$ with $A|w\rangle$ equal to inner product between $A^\dagger|v\rangle$ and $|w\rangle$. Define $|v\rangle^\dagger = \langle v|$. If A is a matrix representing A then $A^\dagger = (A^*)^T$.

An operator is *normal* if $AA^\dagger = A^\dagger A$. It is *Hermitian* if $A^\dagger = A$. It is *unitary* if $A^\dagger A = I$. Hermitian implies normal. Unitary operators preserve inner products and all their eigenvalues have modulus 1. An operator is diagonalizable iff it is normal. If A is diagonalizable and has eigenvectors $|i\rangle$ and eigenvalues λ_i then $A = \sum_i \lambda_i |i\rangle\langle i|$.

A *positive operator* has $\langle v|A|v\rangle \geq 0$ for all $|v\rangle$ (*positive definite* if strict inequality). Positive operators are Hermitian.

If we take a spectral decomposition $A = \sum_a a|a\rangle\langle a|$ then for a function f define $f(A) = \sum_a f(a)|a\rangle\langle a|$. This can be used to define the square root of a positive operator, the logarithm of a positive definite operator or the exponential of a normal operator.

B Box-plus decoder

The belief propagation decoder we described earlier has problems with numerical stability when implemented on a computer. For example multiplying floating point numbers together all of which are close to zero can lead to arithmetic underflow. One way around this difficulty is storing logarithms of probabilities instead of probabilities. The operations required then change: we choose to organize the operations required by using the *box-plus decoder* (see [WSM04] for example for an exposition of the box-plus decoder in the context of nonbinary LDPC codes). We apply this idea from classical LDPC codes to our belief propagation decoder for quantum LDPC codes.

Define $\boxplus\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ by

$$a \boxplus b = \log\left(\frac{1 + e^{a+b}}{e^a + e^b}\right).$$

The following properties are easily checked

(i) $a \boxplus b = b \boxplus a$

(ii) $(a \boxplus b) \boxplus c = a \boxplus (b \boxplus c)$

(iii) $a \boxplus (-b) = (-a) \boxplus b = -(a \boxplus b)$

(iv) $a \boxplus b = 0$ iff $a = 0$ or $b = 0$

(v) $a \boxplus \infty = a$, in the sense that $\lim_{x \to \infty} a \boxplus x = a$.

A related operation is the $\max^*$ function

$$\begin{aligned} \max{}^*(x, y) &= \log(e^x + e^y) \\ &= \max(x, y) + \log(1 + e^{-|x-y|}). \end{aligned}$$

Then

$$\begin{aligned} a \boxplus b &= \log\left(\frac{1 + e^{a+b}}{e^a + e^b}\right) \\ &= \max{}^*(0, a + b) - \max{}^*(a, b). \end{aligned}$$

Our interest in $\boxplus$ can be explained by the following calculation. Consider two independent $\mathbb{F}_2$-valued random variables a_1, a_2 with L_i defined for $i = 1, 2$ by

$$L_i = \log\left(\frac{P(a_i = 0)}{P(a_i = 1)}\right).$$

Then

$$\begin{aligned}
L(a_1+a_2) &= \log\left(\frac{P(a_1+a_2=0)}{P(a_1+a_2=1)}\right)\\
&= \log\left(\frac{P(a_1=0)P(a_2=0)+P(a_1=1)P(a_2=1)}{P(a_1=0)P(a_2=1)+P(a_1=1)P(a_2=0)}\right)\\
&= \log\left(\frac{1+\frac{P(a_1=0)P(a_2=0)}{P(a_1=1)P(a_2=1)}}{\frac{P(a_1=0)}{P(a_1=1)}+\frac{P(a_2=0)}{P(a_1=1)}}\right)\\
&= \log\left(\frac{1+e^{L_1+L_2}}{e^{L_1}+e^{L_2}}\right)\\
&= L_1 \boxplus L_2
\end{aligned}$$

i.e. to find the log-likelihood-ratio for the sum of a_1 and a_2 we just use our box-plus operator to add their separate log-likelihood-ratios.

We now describe a log-domain belief propagation decoder utilizing these $\boxplus$ and $\max^*$ operations. The benefit of working with log-likelihood-ratios (LLRs) instead of probabilities is numerical stability. Working over $\mathbb{F}_4$ the algorithm has some of the features of nonbinary ldpc decoders but recall that our syndromes are actually in $\mathbb{F}_2$ so some of our calculations take place over that field.

Define

$$L(E_q=b) = \log\left(\frac{P(E_q=0)}{P(E_q=b)}\right).$$

Define the depolarizing channel log-likelihood-ratio vector (LLRV) by

$$\boldsymbol{L}_q^{\mathrm{ch}} = \left[\log\left(\frac{1-p}{p/3}\right), \log\left(\frac{1-p}{p/3}\right), \log\left(\frac{1-p}{p/3}\right)\right].$$

The initial message sent from each variable node to check nodes connected to it is

$$\boldsymbol{L}_{q\to c}^{(0)} = \boldsymbol{L}_q^{\mathrm{ch}}.$$

Next we discuss the messages from check c. Say there are m variable nodes connected to check c, labelled $q_1, \ldots q_m$. Now

$$L\left(\sum_{i=1}^{m} \mathrm{tr}(\overline{a_{cq_i}} E_{q_i}) = 1\right) = \boxplus_{i=1}^{m} L(\mathrm{tr}(\overline{a_{cq_i}} E_{q_i}) = 1).$$

To calculate the LLRs on the right hand side notice that if $x, y \in \mathbb{F}_4$ are the two elements which are not 0 or a_{cq} then we have

$$\begin{aligned}
L(\mathrm{tr}(\overline{a_{cq}} E_q) = 1) &= \log\left(\frac{P(E_q=0)+P(E_q=a_{cq})}{P(E_q=x)+P(E_q=y)}\right)\\
&= \log\left(\frac{1+P(E_q=a_{cq})/P(E_q=0)}{P(E_q=x)/P(E_q=0)+P(E_q=y)/P(E_q=0)}\right)\\
&= \log\left(\frac{1+e^{-L(E_q=a_{cq})}}{e^{-L(E_q=x)}+e^{-L(E_q=y)}}\right)\\
&= \max{}^*(0, -L(E_q=a_{cq})) - \max{}^*(-L(E_q=x), -L(E_q=y))
\end{aligned}$$

which can be calculated from $\boldsymbol{L}_{q\to c}$. For our purposes it is more efficient to compute forward and backward sums

$$F_j = \boxplus_{i=1}^{j} L(\mathrm{tr}(\overline{a_{cq_i}} E_{q_i}) = 1)$$

and

$$B_j = \boxplus_{i=j}^{m} L(\mathrm{tr}(\overline{a_{cq_i}} E_{q_i}) = 1).$$

Then

$$\begin{aligned} & L\left(\sum_{\substack{i=1\\ i\neq j}}^{m} \mathrm{tr}(\overline{a_{cq_i}} E_{q_i}) + s_c = 1\right) \\ &= (-1)^{s_c} \boxplus_{\substack{i=1\\ i\neq j}}^{m} L(\mathrm{tr}(\overline{a_{cq_i}} E_{q_i})) \\ &= (-1)^{s_c}(F_{j-1} \boxplus B_{j+1}) \end{aligned}$$

where the first equality holds because $L(s_c = 1) = (-1)^{s_c}\infty$ and $(\pm\infty)\boxplus a = \pm a$ for all a.

Now the message $\boldsymbol{L}_{q_j\to c}$ has 3 components. However there is no way to distinguish between the outcomes $v_{q_j} = 0$ and $v_{q_j} = a_{cq_j}$ based only on the syndrome equation. More precisely, say the elements of $\mathbb{F}_4$ that are not 0 or a_{cq_j} are x, y. Then

$$\begin{aligned} \boldsymbol{L}_{c\to q_j}(b) &= \log\left(\frac{P(\text{check } c \text{ satisfied} \mid E_{q_j} = 0, s)}{P(\text{check } c \text{ satisfied} \mid E_{q_j} = b, s)}\right) \\ &= \log\left(\frac{P\left(\sum_{\substack{i=1\\ i\neq j}}^{m} \mathrm{tr}(\overline{a_{cq_i}} E_{q_i}) + \mathrm{tr}(\overline{a_{cq_j}} 0) = s_c\right)}{P\left(\sum_{\substack{i=1\\ i\neq j}}^{m} \mathrm{tr}(\overline{a_{cq_i}} E_{q_i}) + \mathrm{tr}(\overline{a_{cq_j}} b) = s_c\right)}\right) \end{aligned}$$

from which we can see

$$\boldsymbol{L}_{c\to q_j}(a_{cq_j}) = 0 \text{ and } (\boldsymbol{L}_{c\to q_j})_x = (\boldsymbol{L}_{q_j\to c})_y = (-1)^{s_c}(F_{j-1} \boxplus B_{j+1}).$$

For messages from a variable node to a check node

$$\begin{aligned}\boldsymbol{L}_{q\to c}(b) &= \log\left(\frac{P(E_q = 0 \mid \text{checks } c' \in n(q)\setminus c \text{ satisfied}}{P(E_q = b \mid \text{checks } c' \in n(q)\setminus c \text{ satisfied}}\right)\\ &= \log\left(\frac{p_q^{\mathrm{ch}}(0)\displaystyle\prod_{c'\in n(q)\setminus c} p^{(i)}_{c'\to q}(0)}{p_q^{\mathrm{ch}}(b)\displaystyle\prod_{c'\in n(q)\setminus c} p^{(i)}_{c'\to q}(b)}\right)\\ &= \boldsymbol{L}_q^{\mathrm{ch}}(b) + \sum_{c'\in n(q)\setminus c} \boldsymbol{L}_{c'\to q}(b)\end{aligned}$$

so

$$\boldsymbol{L}_{q\to c} = \boldsymbol{L}_q^{\mathrm{ch}} + \sum_{c'\in n(q)\setminus c} \boldsymbol{L}_{c'\to q}.$$

Now to find the best estimate for a codeword we calculate

$$\boldsymbol{L}_q = \boldsymbol{L}_q^{\mathrm{ch}} + \sum_{c'\in n(q)} \boldsymbol{L}_{c'\to q}.$$

If all the components are positive then decode to 0. Otherwise decode to the element $E_q \in \mathbb{F}_4^*$ with least (i.e. most negative) $\boldsymbol{L}_q(x)$.

We summarize the algorithm below with some small rearrangements to improve efficiency.

Algorithm 3 BoxPlusF4BeliefPropagation(A,s,maxIter)

Require: $\mathbb{F}_4$-check matrix A, syndrome vector s, positive integer maxIter
Ensure: Output estimated error E and flag for whether decoding was successful

Set $\boldsymbol{L}_q^{\mathrm{ch}} = \left[\log\left(\frac{1-p}{p/3}\right), \log\left(\frac{1-p}{p/3}\right), \log\left(\frac{1-p}{p/3}\right)\right]$
Set $\boldsymbol{L}_{q\to c}^{(0)} = \boldsymbol{L}_q^{\mathrm{ch}}$
for $i = 1$ to $i =$maxIter **do**
  for all check nodes c **do**
    Label variable nodes in $n(c)$ as $q_1, \ldots, q_m$
    for $j = 1$ to $j = m$ **do**
      Let x, y be the elements of $\mathbb{F}_4$ that are not 0 or a_{cq_j}
      Calculate $L(\mathrm{tr}(\overline{a_{cq_j}} v_{q_j}) = 1)$
      $= \max^*(0, -\boldsymbol{L}_{q_j\to c}^{(i-1)}(a_{cq_j})) - \max^*(-\boldsymbol{L}_{q_j\to c}^{(i-1)}(x), -\boldsymbol{L}_{q_j\to c}^{(i-1)}(y))$
    end for
    Set $F_0 = \infty$ and $B_{m+1} = \infty$
    for $j = 1$ to $j = m$ **do**
      Set $F_j = F_{j-1} \boxplus L(\mathrm{tr}(\overline{a_{cq_j}} E_{q_j}) = 1)$
      Set $k = m + 1 - j$
      Set $B_k = B_{k+1} \boxplus L(\mathrm{tr}(\overline{a_{cq_k}} E_{q_k}) = 1)$
    end for
    for $j = 1$ to $j = m$ **do**
      Set $\boldsymbol{L}_{c\to q_j}(a_{cq_j}) = 0$
      Set $\boldsymbol{L}_{c\to q_j}(x) = \boldsymbol{L}_{q_j\to c}(y) = (-1)^{s_c}(F_{j-1} \boxplus B_{j+1})$
    end for
  end for
  for all variable nodes q **do**
    Set $\boldsymbol{L}_q^{(i)} = \boldsymbol{L}_q^{\mathrm{ch}} + \sum_{c'\in n(q)} \boldsymbol{L}_{c'\to q}^{(i)}$
    Decide the most likely error E_q by
    $E_q = \mathrm{argmin}(0, \boldsymbol{L}_q(1), \boldsymbol{L}_q(\omega), \boldsymbol{L}_q(\omega^2))$
    for all check nodes $c \in n(q)$ **do**
      Set $\boldsymbol{L}_{q\to c}^{(i)} = \boldsymbol{L}_q^{(i)} - \boldsymbol{L}_{c\to q}^{(i)}$
    end for
  end for
  Set $E = (E_q)$
  if $AE = 0$ **then**
    return E and flag successful decoding
  end if
end for
return E and flag unsuccessful decoding

References

[CT06] T.M. Cover and JA Thomas. *Elements of Information*. John Wiley & Sons, New Jersey, 2006.

[Got97] D. Gottesman. Stabilizer codes and quantum error correction. *Arxiv preprint quant-ph/9705052*, 1997.

[Gur10] V. Guruswami. Course notes for introduction to coding theory. *Carnegie Mellon University*, 2010.

[Mac03] D.J.C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge Univ Pr, 2003.

[MMM04] D.J.C. MacKay, G. Mitchison, and P.L. McFadden. Sparse-graph codes for quantum error correction. *Information Theory, IEEE Transactions on*, 50(10):2315–2330, 2004.

[NC] M.A. Nielsen and I.L. Chuang. Quantum computation and quantum information.

[PC08] D. Poulin and Y. Chung. On the iterative decoding of sparse quantum codes. *Quantum Information & Computation*, 8(10):987–1000, 2008.

[Pre98] J. Preskill. Lecture notes for physics 229: Quantum information and computation. *California Institute of Technology*, 1998.

[RU08] T.J. Richardson and R.L. Urbanke. *Modern coding theory*. Cambridge Univ Pr, 2008.

[Ryc12] M. Rychlik. On axiomatic characterization of quantum operations. 2012.

[TZ09] J.P. Tillich and G. Zémor. Quantum ldpc codes with positive rate and minimum distance proportional to $n^{1/2}$. In *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, pages 799–803. IEEE, 2009.

[WSM04] H. Wymeersch, H. Steendam, and M. Moeneclaey. Log-domain decoding of ldpc codes over gf(q). In *Communications, 2004 IEEE International Conference on*, volume 2, pages 772–776. IEEE, 2004.