

Constructions and performance of classes of quantum LDPC codes

Thomas Camara* Harold Ollivier† Jean-Pierre Tillich‡

Abstract

Two methods for constructing quantum LDPC codes are presented. We explain how to overcome the difficulty of finding a set of low weight generators for the stabilizer group of the code. Both approaches are based on some graph representation of the generators of the stabilizer group and on simple local rules that ensure commutativity. A message passing algorithm for generic quantum LDPC codes is also introduced. Finally, we provide two specific examples of quantum LDPC codes of rate $1/2$ obtained by our methods, together with a numerical simulation of their performance over the depolarizing channel.

1 Introduction

The idea of using quantum systems for processing information has been suggested by R. P. Feynman as a potential way of bypassing the difficulty of simulating quantum physics with classical computers. Since then, it has developed into an exciting research area with implications ranging from cryptography (see e.g. [BB84a, Eke91a, SP00a]) to complexity theory (e.g. [Amb04a, Reg04a]). For instance, quantum computers could solve efficiently some hard problems such as integer factorization [Sho94a], or give quadratic speed-up over optimal classical algorithms, as it is the case for unsorted database search [Gro96a].

However, for taking advantage of the quantum nature of physical systems to process information, it is necessary to protect them from unwanted evolutions. Indeed, if quantum registers are not protected from noise, the very fragile superpositions required for efficiently manipulating quantum information tend to disappear exponentially fast with the number of qubits involved. This effect — called decoherence — can nonetheless be reduced by using quantum error correcting codes. The first scheme of this kind has been proposed in 1995 by P. Shor [Sho95a], and triggered numerous work on quantum error correction. Most notable, was the introduction of the stabilizer formalism [Got97a, CRSS98a] — for defining quantum codes and finding the gate

*INRIA, Projet Codes, BP 105, Domaine de Voluceau, F-78153 Le Chesnay, France.

†Perimeter Institute, 31 Caroline St. N, Waterloo, ON N2L 2Y5, Canada.

‡INRIA, Projet Codes, BP 105, Domaine de Voluceau, F-78153 Le Chesnay, France.

implementation of encoding and decoding circuits — together with the class of CSS codes [CS96a, Ste96a]. With these tools at hand, it has been shown that quantum information processing can be done fault-tolerantly (see for instance [AB97a]), i.e. would be feasible even in the presence of qubit errors and gate faults — provided these events are rare enough.

In spite of these important results, properties of quantum codes are less understood than those of for their classical counterparts. It is thus of interest to tackle the problem from a pragmatic point of view: devise versatile constructions of quantum codes inspired by the best classical codes, and analyze their performances.

When sending classical information over memoryless classical channels, it has been demonstrated that a very efficient way for approaching the channel capacity is obtained by using LDPC codes with Gallager's iterative decoding algorithm. Generalizing these notions to quantum codes seems a promising way, and has indeed been proposed recently [MMM04a]. In this work, D. J. C. MacKay *et al.* have shown how to construct sparse weakly self-dual binary codes that can be used to construct quantum LDPC codes using Calderbank-Shor-Steane's method.

Our work is aimed at finding other constructions of quantum LDPC codes within the stabilizer formalism. While a brief introduction to stabilizer codes is provided below, we would like to pinpoint here the main difficulty for finding LDPC stabilizer codes. As explained in [CRSS98a], each stabilizer code can be viewed as a code over \mathbb{F}_4 , the field with four elements. However, the converse is not true: to correspond to valid stabilizer codes, these codes must be self-orthogonal for some Hermitian trace inner product. Fulfilment of this peculiar constraint makes usual constructions of LDPC codes useless in the quantum setting. Our work provides a partial solution to this problem by defining such codes through simple group theoretical constructions (see Sections 3.2 and 3.3). The resulting codes can be viewed as quantum analogs of regular Gallager codes: each qubit is involved in the same number of parity-check equations, and each parity-check equation involves the same number of qubits. Error estimation has been performed by an iterative algorithm applied to a Tanner graph associated to our constructions. The results of these numerical simulations are presented in Section 4.

2 Stabilizer codes

The code subspace. An (n, k) quantum code is a subspace of dimension 2^k of the Hilbert space $\mathcal{H}_n \cong (\mathbb{C}^2)^{\otimes n}$ of n qubits. Such subspace allows to encode k qubits, so that the rate of an (n, k) quantum code is $\frac{k}{n}$. While defining the code subspace can be done in many different ways, a particularly useful method is known as the stabilizer formalism.

The definition of stabilizer codes relies heavily on properties of Pauli matrices

for one qubit, i.e. acting on \mathcal{H}_1 :

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

In particular, these matrices square to the identity; they generate a finite multiplicative group \mathcal{G}_1 ; and any pair (P, Q) of elements in \mathcal{G}_1 either commutes ($PQ - QP = 0$) or anti-commutes ($PQ + QP = 0$).

By extension, the Pauli group \mathcal{G}_n over n -qubits is the multiplicative group generated by all possible tensor products of Pauli matrices over n qubits, i.e. elements of the form $P_1 \otimes P_2 \otimes \dots \otimes P_n$ where the P_i 's belong to \mathcal{G}_1 . As in the case of the Pauli group of a single qubit, any pair of elements in \mathcal{G}_n either commutes or anti-commutes.

The code subspace \mathcal{C} of a (n, k) stabilizer code is defined as the largest subspace of \mathcal{H}_n stabilized by the action of a subgroup \mathcal{S} of \mathcal{G}_n . For \mathcal{C} to be of dimension 2^k , it requires \mathcal{S} to be an Abelian subgroup generated by $n - k$ independent elements M_i , and such that $-I^{\otimes n} \notin \mathcal{S}$. That is, the code subspace is defined as,

$$|\psi\rangle \in \mathcal{C} \Leftrightarrow \forall i, M_i|\psi\rangle = |\psi\rangle.$$

Error estimation. Before explaining how errors can be estimated, we must first define what the possible errors are. It is well known that the most general transformation on n qubits allowed by quantum mechanics is a completely positive trace preserving map. However, as shown in [BDSW96a, EM96a, KL97a], it is sufficient to consider errors as elements of \mathcal{G}_n since any admissible quantum operation can be written in Kraus form using only elements of the Pauli group over n qubits.

Thus, let $E \in \mathcal{G}_n$ be an error on the qubits of a (n, k) stabilizer code with stabilizer group $\mathcal{S} = \langle M_i \rangle_{i=1}^{n-k}$. For each i , E and M_i either commute or anti-commute. Hence, for $|\psi\rangle \in \mathcal{C}$, either $M_i E|\psi\rangle = E|\psi\rangle$ or $M_i E|\psi\rangle = -E|\psi\rangle$. The action of E on \mathcal{C} is either to map \mathcal{C} onto itself, or to map \mathcal{C} onto an orthogonal subspace of \mathcal{H}_n . Thus, the measurement associated to the M_i 's reveals onto which orthogonal subspace \mathcal{C} has been mapped: the $n - k$ dimensional binary vector $s(E) = (s_1(E), \dots, s_{n-k}(E))$, such that $(-1)^{s_i(E)} = \langle \psi | E^\dagger M_i E | \psi \rangle$, characterizes EC , and is called the syndrome of E .

Also note that not all errors E with $s(E) = (0, \dots, 0)$ are harmful. Indeed if $E \in \mathcal{S}$, then $\forall |\psi\rangle \in \mathcal{C}$, we have $E|\psi\rangle = |\psi\rangle$, and quantum information is preserved. Only errors with non-trivial action on \mathcal{C} harm quantum information. These errors E are by definition elements of $N(\mathcal{S}) - \mathcal{S}$, where $N(\mathcal{S})$ is the normalizer of \mathcal{S} .

A generic error estimation procedure then consists in finding $\hat{E} \in \mathcal{G}_n$ given the error model for the n qubits and the syndrome $s(E)$ such that $\hat{E}^\dagger E \in \mathcal{S}$. Such \hat{E} is called the estimated error. The most common error model considered in the literature is the depolarizing channel where X , Y and Z errors occur each with probability $p/3 < 1/4$, and independently on each qubit. Usually, the estimated error is chosen accordingly to the maximum likelihood criterion

which is equivalent to choosing for \hat{E} the lowest weight element of \mathcal{G}_n such that $s(\hat{E}) = s(E)$. The error recovery will be successful when $\hat{E}^\dagger E \in \mathcal{S}$, otherwise a block error is produced.

Quantum codes as codes over \mathbb{F}_4 . Stabilizer codes might also be viewed as codes over $\mathbb{F}_4 = \{0, 1, \omega, \bar{\omega}\}$, with $1 + \omega + \omega^2 = 1 + \omega + \bar{\omega} = 0$. This duality is due to the additive structure of \mathbb{F}_4 which echoes the multiplicative structure of the Pauli operators I, X, Y, Z . The mapping between one and the other is the following: $I \leftrightarrow 0$, $X \leftrightarrow \omega$, $Z \leftrightarrow \bar{\omega}$ and $Y \leftrightarrow 1$. In addition, with the trace operation $\text{tr}(x) \triangleq x + \bar{x} = x + x^2$, it can be checked that $\text{tr}(a\bar{b})$ is 0 iff the Pauli operators associated to a and b commute and 1 iff they anti-commute. This motivates the definition of the following inner product over \mathbb{F}_4^n :

$$\langle u, v \rangle \triangleq \text{tr} \sum_{i=1}^n u_i \bar{v}_i.$$

The generators of a stabilizer code, when expressed as row-vectors of \mathbb{F}_4^n , form a matrix M which will be called a parity-check matrix of the stabilizer code. In what follows, both the i -th generator defining the stabilizer code and its expression as the i -th row of M are denoted by M_i . These rows are orthogonal with respect to the inner product defined above. The converse is also true: any $(n - k) \times n$ matrix over \mathbb{F}_4 with orthogonal rows defines a stabilizer code over n qubits.

While stabilizer codes were originally introduced using the language of the Pauli group, we will mainly employ the \mathbb{F}_4 formalism. This choice simplifies the notation, and better points out analogies and differences in constructing classical and quantum LDPC codes. Interested readers are redirected to [Got97a, CRSS98a] for a more complete treatment of stabilizer codes such as gate implementation and additional connections between stabilizer codes and codes over \mathbb{F}_4 .

3 Constructing regular quantum LDPC codes

According to the definition of a parity-check matrix for stabilizer codes, it is natural to introduce quantum LDPC codes as stabilizer codes with sparse parity-check matrix:

Definition 1. *An (a, b) -regular quantum LDPC code is a stabilizer code whose parity-check matrix M has “ a ” non-zero entries per column and “ b ” non-zero entries per row.*

In the rest of this article we are going to construct and analyze some (a, b) -regular LDPC codes.

3.1 Iterative error estimation for quantum LDPC codes

To define the iterative error estimation algorithm we use the following Tanner graph associated to a parity check matrix M . It is a bipartite graph with vertex set $V \cup W$. The vertices of V are associated to qubits of the code (i.e. columns of M) and vertices of W to the generators of the stabilizer code (i.e. rows of M). There is an edge between $v \in V$ and $w \in W$ iff the corresponding entry M_{wv} in M is non zero. We label this edge with the entry M_{wv} . The purpose of the iterative error estimation algorithm is to find the most likely value for each \hat{E}_i such that $\hat{E} = (\hat{E}_1, \dots, \hat{E}_n)$ is compatible with the observed syndrome $s(E) = (s_i(E))_{i=1}^{n-k}$. The local constraint on \hat{E} associated to the j -th check node (i.e. vertices in W) is given by the equation $\langle \hat{E}, M_j \rangle = s_j(E)$.

This graph is nothing but the syndrome version of a Tanner graph. The algorithms we can use to estimate errors are the SUM-PRODUCT or the MIN-SUM algorithms in their syndrome decoding form applied to this graph. That is, the version of these algorithms which estimates the most likely value of the error for each qubit given the observed syndrome. Note that iterative decoding of classical codes estimates instead the most likely value of the codeword symbols for each bit of the code.

3.2 Generic construction of (a, b) -regular quantum LDPC codes

In order to construct quantum LDPC codes we restrict our attention to codes having a parity check matrix M with two kinds of non zero entries, ω and $\bar{\omega}$.

We start our construction by choosing a group G with cardinality equal to a multiple of the length of the code we are interested in. Then we choose two subgroups H and K of G , with $|K| > |H|$. The cosets xH are associated to qubits whereas the cosets yK are associated to rows of the parity-check matrix. In other words the length n and the number of rows $n - k$ in the parity check matrix are given by

$$n = \frac{|G|}{|H|}, \quad n - k = \frac{|G|}{|K|}.$$

We then pick a set of generators G of G that can be partitioned into two sets $G = G_\omega \cup G_{\bar{\omega}}$ such that the following properties are satisfied:

$$(G_\omega)^{-1} = G_\omega, \quad (G_{\bar{\omega}})^{-1} = G_{\bar{\omega}} \quad (1)$$

$$\forall (g_\omega, g_{\bar{\omega}}) \in G_\omega \times G_{\bar{\omega}}, \quad g_\omega g_{\bar{\omega}} = g_{\bar{\omega}} g_\omega \quad (2)$$

$$g, g' \in G, \quad h, h' \in H, \quad k, k' \in K, \quad ghk = g'h'k' \implies g = g' \quad (3)$$

We put an edge between coset xH and coset yK iff there exists $g \in G$ such that $xgH \cap yK \neq \emptyset$, or equivalently iff there exist $h \in H, k \in K$ such that $y = xghk$. We label this edge with a ω if the corresponding g belongs to G_ω and with a $\bar{\omega}$ otherwise. It can be checked that the degree of any vertex xH is equal to $a \triangleq |G||H|/|H \cap K|$ and the degree of any vertex yK is equal to $b \triangleq |G||K|/|H \cap K|$. This is a simple consequence of the following lemma:

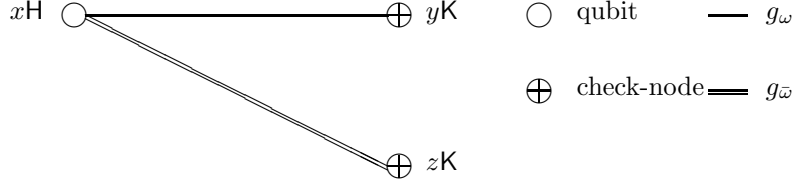


Figure 1: Subgraph induced by the qubit xH and the two check-nodes yK and zK .

Lemma 1. *The intersection of a coset xH and yK is either empty or equal to a coset $z(H \cap K)$.*

This defines the Tanner graph of our code and therefore also its parity-check matrix M . Property (3) imposes that there are no multiple edges in the graph. The point of this construction is that the commutation of the g_ω 's with the $g_{\bar{\omega}}$'s implies the orthogonality of the rows of M . Thus, the matrix M is a valid parity-check matrix for defining a stabilizer code.

Proposition 1. *The parity-check matrix M associated to the Tanner graph given by this construction has orthogonal rows.*

Proof. Given two rows M_i and M_j of M , a parity-check matrix of a stabilizer code, one can partition the set of qubits in two classes: qubits for which the corresponding entries in M_i and M_j have inner product 0, and those for which the entries have inner product 1. For M_i and M_j to be orthogonal, it is necessary and sufficient that the number of qubits of the second type is even. Equivalently, and this is how our construction is tailored, we must show that qubits of the second type can be paired together.

Consider a qubit belonging to the second class; say it corresponds to the vertex xH and the rows M_i and M_j correspond to the cosets yK and zK . Since for this qubit the entries of M_i and M_j have inner product 1, one of these two rows has a ω at the position of xH and the other one has a $\bar{\omega}$. Without loss of generality we may assume that the subgraph of the Tanner graph induced by xH , yK and zK is as in Fig. 1. In other words there exist $g_\omega \in G_\omega$, $g_{\bar{\omega}} \in G_{\bar{\omega}}$ such that $xg_\omega H \cap yK \neq \emptyset$ and $xg_{\bar{\omega}} H \cap zK \neq \emptyset$. Let $x' \triangleq xg_\omega g_{\bar{\omega}}$. Note that there exist h_1, h_2 in H and k_1, k_2 in K such that

$$y = xg_\omega h_1 k_1 \quad (4)$$

$$z = xg_{\bar{\omega}} h_2 k_2 \quad (5)$$

Since $xg_\omega = x'g_{\bar{\omega}}^{-1}$, using (4) implies that $y = x'g_{\bar{\omega}}^{-1}h_1k_1$. Therefore, there is an edge labeled by $\bar{\omega}$ between $x'H$ and yK . Because $g_\omega g_{\bar{\omega}} = g_{\bar{\omega}} g_\omega$ we also have $xg_{\bar{\omega}} = x'g_\omega^{-1}$. A similar reasoning shows that $z = x'g_\omega^{-1}h_2k_2$, which implies that there is an edge labeled by ω between $x'H$ and zK . Each qubit of the second class is necessarily involved in a 4-cycle with another such qubit (See Fig. 2).

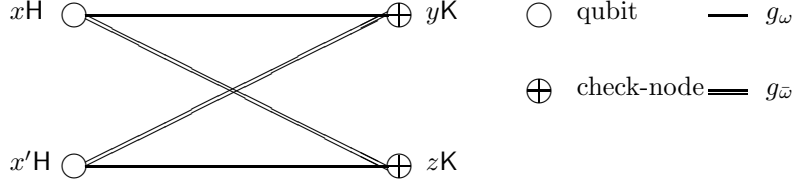


Figure 2: Subgraph showing that each qubit xH of the second type (involved in M_i with a ω and in M_j with a $\bar{\omega}$) is necessarily part of a 4-cycle with another qubit $x'H = xg_\omega g_{\bar{\omega}}H$ of the second type.

From the absence of multiple edges in our Tanner graph, qubits of the second class can always be arranged in pairs forming the 4-cycle described above. Hence, $\langle M_i, M_j \rangle = 0$ for any i and j , and M is the parity-check matrix of a stabilizer code. \square

Remark 1. For satisfying the commutation constraints, it is necessary to introduce many 4-cycles in the Tanner graph associated to stabilizer codes. More precisely, let us define the *4-cycle graph* associated to a Tanner graph as the graph with vertex set the qubits and with edges connecting two qubits iff they are involved in the same 4-cycle in the Tanner graph. The 4-cycle graph associated to our construction is a regular graph with degree at least $|G_\omega||G_{\bar{\omega}}||H|^2/|H \cap K|^2$. More generally, (a, b) -regular codes, where each qubit is involved in a_ω generators with a ω and $a_{\bar{\omega}}$ generators with a $\bar{\omega}$, have a 4-cycle graph where each vertex has degree at least $a_\omega a_{\bar{\omega}}$. The 4-cycle graph is of higher degree in our construction if the group or the generators are badly chosen, for instance if G is Abelian, or if there are commuting generators in G_ω or in $G_{\bar{\omega}}$ that are not inverse of each other.

Remark 2. A closer look at the proof shows that 4-cycles come from qubits which are involved in two different ways in two generators. It could be thought that in order to avoid these 4-cycles we just have to ensure that each qubit is involved in the same way in all the generators it belongs to. This is a bad solution. Assume for instance that a qubit is involved in all its generators with label ω . Then the single qubit error with a ω at this position and 0 elsewhere generally belongs to the set of undetectable errors. 4-cycles are therefore unavoidable if we want stabilizer codes with good error-correcting capabilities.

Example 1. We obtain a $(6, 12)$ -regular quantum LDPC code with the following choices:

- $G = PSL_2(\mathbb{F}_5) \times PSL_2(\mathbb{F}_5)$ ($PSL_2(\mathbb{F})$ denotes the quotient of group of 2×2 matrices over the field \mathbb{F} of determinant 1 by its center, that is $\{\mathbb{I}, -\mathbb{I}\}$;

- $H = \{\mathbb{I}\}$ and $K = \{\mathbb{I}, u\}$ where $u^2 = \mathbb{I}$, with u given by

$$u = \left(\begin{pmatrix} 1 & 2 \\ -1 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 2 \\ -1 & -1 \end{pmatrix} \right);$$

- $G_\omega = \left\{ \left(\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \mathbb{I} \right), \left(\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \mathbb{I} \right), \left(\begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}, \mathbb{I} \right) \right\};$
- $G_{\bar{\omega}} = \left\{ \left(\mathbb{I}, \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \right), \left(\mathbb{I}, \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \right), \left(\mathbb{I}, \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} \right) \right\}.$

This code has block length $n = 3600$ and $n - k = 1800$ generators. By reducing the parity-check matrix in its standard form, see [Got97a], we know that these generators are independent. The dimension of the code (that is the number of encoded qubits) is therefore 1800. The code is of rate $1/2$.

3.3 Construction of a (4,8)-regular quantum LDPC code

If we want to obtain a (4,8)-regular quantum LDPC code from the construction given previously, it turns out that we cannot avoid choosing an Abelian group for G . This would not be a wise choice as it would induce many unwanted 4-cycles. Here, we give an alternative construction for (4,8) codes such that the associated 4-cycle graph is 4-regular. In addition, we impose that each column of M has two entries with a ω and two with a $\bar{\omega}$. This defines a quantum LDPC code of type $([2_\omega, 2_{\bar{\omega}}], 8)$.

Since the 4-cycle graph will be constructed as a Cayley graph, we remind that:

Definition 2. For G a finite group and $S \subset G$ satisfying $S = S^{-1}$, the Cayley graph $X(G, S)$ is the graph having G as vertex set and edges formed by pairs $\{x, y\}$ such that $x = gy$. $X(G, S)$ is k -regular with $k = |S|$.

Our alternative construction is obtained by following a 3-step procedure.

Step 1: construction of the 4-cycle graph. The intuition behind our construction is the following remark which explains how the 4-cycle graph can be used in order to build the Tanner graph. We notice that the subgraph of the 4-cycle graph of a $([2_\omega, 2_{\bar{\omega}}], 8)$ code induced by the qubits involved in a generator is a 2-regular graph with 8 vertices. It is therefore either a cycle of length 8 or a union of cycles. We choose to construct $([2_\omega, 2_{\bar{\omega}}], 8)$ codes for which all these subgraphs are single cycles. Therefore, following certain cycles in the 4-cycle graph will reveal which qubits are involved in the associated generator.

We obtain the 4-cycle graph in the following way. For p prime such that $p \equiv 1 \pmod{4}$, let \mathbb{F}_p be the finite field of size p and consider

$$G = \{ M \in GL_2(\mathbb{F}_p) \mid (\det M)^2 = \pm 1 \}.$$

Choose $g_+, g_- \in GL_2(\mathbb{F}_p)$, such that:

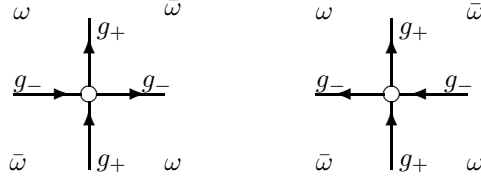


Figure 3: Subgraphs of the 4-cycle graph showing the two types of qubits encountered. Each quadrant corresponds to an 8-cycle associated with a generator in which the qubit has a non-zero entry. The ω 's and $\bar{\omega}$'s of each quadrant represent the value of the entry for the corresponding generator obtained in the labeling step.

- $\det g_+ = \pm 1$ and $\det g_- = \pm i$
- \mathbb{G} is generated by the set

$$S = \{g_+, g_+^{-1}, g_-, g_-^{-1}\},$$

$$\text{satisfying } (g_+ g_-^{-1})^4 = (g_- g_+)^4 = \mathbb{I}.$$

The graph $X(\mathbb{G}, S)$ is 4-regular and has $4p(p^2 - 1)$ vertices. This defines the 4-cycle graph.

Step 2: Construction of a $(4, 8)$ -regular graph. It is easy to check that any qubit is involved in four cycles of length 8 corresponding respectively to the relations $(g_+ g_-)^4 = \mathbb{I}$, $(g_- g_+^{-1})^4 = \mathbb{I}$, $(g_- g_+)^4 = \mathbb{I}$, $(g_+^{-1} g_+)^4 = \mathbb{I}$. This gives the 4 generators in which this qubit is involved. The $(4, 8)$ -regular graph is then obtained by putting an edge between each qubit and the generators it belongs to.

Step 3: Edge labeling. For the $(4, 8)$ -regular graph constructed in the previous step to be the Tanner graph associated to a quantum LDPC code, we must now label the edges in a way that commutation relations are satisfied. This can be done in many different ways. We give one way of performing this task which is the one used for the example considered in the next section.

Let us first notice there is a natural bipartition of \mathbb{G} , and hence of the qubits, in two classes $\{g \in \mathbb{G}, \det g = \pm 1\}$ and $\{g \in \mathbb{G}, \det g = \pm i\}$. The edges leaving a qubit of the first class are labeled as follows. The two edges linking the qubit to the generators corresponding to the cycles $(g_+ g_-)^4 = \mathbb{I}$ and $(g_- g_+)^4 = \mathbb{I}$ get the label ω ; the two other edges receive the label $\bar{\omega}$. The edges leaving the qubits of the second class are labeled in the opposite way: the two edges linking such qubit to the generators of type $(g_+ g_-)^4 = \mathbb{I}$ and $(g_- g_+)^4 = \mathbb{I}$ get the label $\bar{\omega}$, whereas the two other edges receive the label ω (see Fig.3).

It is then straightforward to check that this labeling defines commuting generators.

Remark 3. It should be noted that the $(4, 8)$ -regular quantum LDPC code obtained with this construction looks in many ways like a (classical) cycle graph. In particular the minimum weight of an undetectable error (in other words the minimum distance of the quantum code) is at most logarithmic in the number of qubits. This can be seen by considering the subgraph of the Tanner graph consisting in taking only edges with label ω . This graph is $(2, 4)$ -regular. It has therefore a cycle of logarithmic size in the number of vertices. It is straightforward to check that the error consisting in putting a ω at qubits belonging to this cycle and 0's elsewhere is an undetectable error.

Example 2. We obtain a $(4, 8)$ quantum LDPC code with the following choices:

- $p = 13$;
- $g_+ = \begin{pmatrix} 9 & 9 \\ 12 & 10 \end{pmatrix}, g_- = \begin{pmatrix} 11 & 7 \\ 5 & 6 \end{pmatrix}$.

It can be checked readily that $n = 8736$ and $k = 4370$.

4 Results

We present in Fig. 4 simulation results of two quantum LDPC codes of rate $1/2$ on the depolarizing channel. We plot the block error probability against the probability that there is an error of type X, Y or Z . The first code is a $(4, 8)$ -regular LDPC code of length 8736 and the second one is a $(6, 12)$ -regular LDPC code of length 3600. Despite the fact that the latter code is smaller, it behaves much better with respect to the iterative error estimation algorithm.

5 Conclusion

One of the drawbacks of the quantum LDPC codes constructed in this article is that they have many cycles of size 4. However, it should be emphasized that this is not due to the particular construction chosen here, but that it is a characteristic of all stabilizer codes. This affects iterative decoding performances when the usual SUM-PRODUCT or MIN-SUM algorithm is used. It would be interesting to study how variants of this algorithm (see [YFW04a] for instance) would overcome this problem. In light of the fact that well chosen (classical) irregular LDPC codes perform much better than their regular counterpart, it would also be interesting to study whether our construction could be generalized to yield irregular LDPC codes.

Finally, we would like to point out that quantum LDPC codes might be good candidates for constructing fault-tolerant architectures. First, we hope that the minimum distance of these codes increases linearly with the block size, as it is the case for most classical LDPC codes. This would warrant that any finite weight error can be corrected for sufficiently large block sizes. Second, the rate of such codes does not decrease to zero, thus possibly improving the overhead

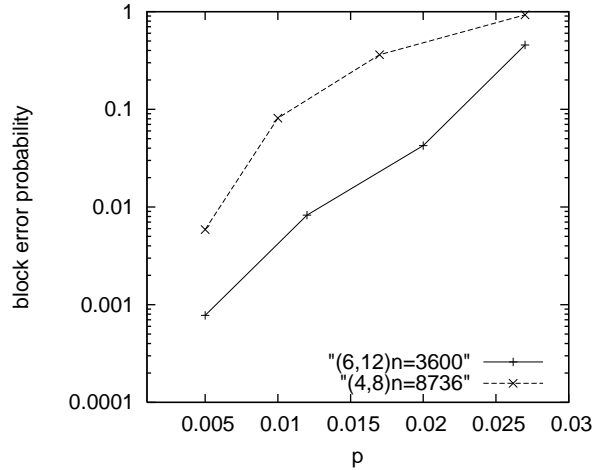


Figure 4: Numerical simulations of the performance of a (4, 8) and a (6, 12)-regular quantum LDPC codes. The error estimation algorithm is the syndrome version of MIN-SUM. The error model is the depolarizing channel which acts on one-qubit density matrices as: $\rho \rightarrow (1 - p)\rho + \frac{p}{3}(X\rho X + Y\rho Y + Z\rho Z)$.

requirements over schemes employing concatenation or toric codes [DKLP02a]. Finally, and by definition, measurements involved in the determination of the syndrome for quantum LDPC codes is of fixed gate complexity, making it less prone to produce erroneous syndromes than concatenated codes. If these are potential advantages of using quantum LDPC codes for fault-tolerance, the issue of error propagation in a circuit manipulating encoded information needs to be addressed for them to become of practical interest. We chose to postpone discussion of these issues to forthcoming publications.

This work was supported in part by ACI Sécurité Informatique — Réseaux Quantiques. H.O. would like to thank D. Poulin for stimulating discussions and comments on an earlier version of the manuscript.

References

- [AB97a] Aharonov, D., and Ben-Or, M. Fault-tolerant quantum computation with constant error rate. In *Proc. 29th. Ann. ACM Symp. on Theory of Computing*, 1997. arXiv, quant-ph/9611025. Longer version quant-ph/9906129.
- [Amb04a] Ambainis, A. Quantum walk algorithm for element distinctness. In *Proc. 45th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, 2004. arXiv, quant-ph/0311001.

- [BB84a] Bennett, C. H., and Brassard, G. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE international Conference on Computers, Systems and Signal Processing, Bangalore, India*, page 175, New York, 1984. IEEE Press.
- [BDSW96a] Bennett, C. H., DiVincenzo, D. P., Smolin, J. A., and Wootters, W. K. Mixed state entanglement and quantum error-correcting codes. *Phys. Rev. A*, 54:3824, 1996. arXiv, quant-ph/9604024.
- [CRSS98a] Calderbank, A. R., Rains, E. M., Shor, P. W., and Sloane, N. J. A. Quantum error correction via codes over $GF(4)$. *IEEE Trans. Inf. Theory*, 44:1369, 1998. arXiv, quant-ph/9608006.
- [CS96a] Calderbank, A. R., and Shor, P. W. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54:1098–1105, 1996. arXiv, quant-ph/9512032.
- [DKLP02a] Dennis, E., Kitaev, A., Landahl, A., and Preskill, J. Topological quantum memory. *J. Math. Phys.*, quant-ph/0110143(9):4452–4505, 2002. arXiv, quant-ph/0110143.
- [Eke91a] Ekert, A. K. Quantum cryptography based on Bell’s theorem. *Phys. Rev. Lett.*, 67:661, 1991.
- [EM96a] Ekert, A., and Machiavello, C. Quantum error correction for communication. *Phys. Rev. Lett.*, 77:2585–2588, 1996.
- [Got97a] Gottesman, D. *Stabilizer codes and quantum error correction*. PhD thesis, California Institute of Technology, Pasadena, CA, 1997. arXiv, quant-ph/9705052.
- [Gro96a] Grover, L. A fast quantum mechanical algorithm for database search. In *Proc. 28th Annual ACM Symposium on the Theory of Computation*, pages 212–219, New York, NY, 1996. ACM Press, New York. arXiv, quant-ph/9605043.
- [KL97a] Knill, E., and Laflamme, R. Theory of quantum error-correcting codes. *Phys. Rev. A*, 55:900, 1997. arXiv, quant-ph/9604034.
- [MMM04a] MacKay, D. J. C., Mitchison, G., and McFadden, P. L. Sparse graph codes for quantum error-correction. *IEEE Trans. Info. Theory*, 50(10):2315–2330, 2004. arXiv, quant-ph/0304161.
- [Reg04a] Regev, O. A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space, 2004. arXiv, quant-ph/0406151.
- [Sho94a] Shor, P. W. Algorithms for quantum computation: Discrete logarithms and factoring. In S. Goldwasser, Ed., *Proceedings of the 35th Annual Symposium on the Foundations of Computer Science*, pages 124–134, Los Alamitos, CA, 1994. IEEE Computer Society.

- [Sho95a] Shor, P. W. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:2493, 1995.
- [SP00a] Shor, P. W., and Preskill, J. Simple proof of security of BB84 quantum key distribution protocol. *Phys. Rev. Lett.*, 85:441–444, 2000. arXiv, quant-ph/0003004.
- [Ste96a] Steane, A. M. Error correcting codes in quantum theory. *Phys. Rev. Lett.*, 77:793, 1996.
- [YFW04a] Yedidia, J. S., Freeman, W. T., and Weiss, Y. Constructing free energy approximations and generalized belief propagation algorithms. Technical Report TR2004-40, Mitsubishi Electric Research Laboratories, 2004.