Quantum error correction: Lecture 4 — hypergraph product codes

Note. During the class, I covered quantum Tanner codes instead of hypergraph product codes.

The toric code that we discussed in the previous lecture is the simplest topological code. In general, one can define a topological code for any tessellation of a closed manifold. Let us restrict ourselves to 2D surfaces. In this case, a tessellation is a partition of the surface into a set of elementary cells called plaquettes (of dimension 2) together with edges (dimension 1) and vertices (dimension 0). A topological code is a CSS code defined by two classical codes $\mathcal{C}_1$ and $\mathcal{C}_2$, with the property that $\mathcal{C}_2^\perp \subseteq \mathcal{C}_1$, where

- the code $\mathcal{C}_1$ is the *cycle code* of the tessellation: the support of codewords corresponds to a cycle, *i.e.* its boundary is zero;
- the code $\mathcal{C}_2^\perp$ is generated by words whose support is the boundary of a set of plaquettes.

The inclusion $\mathcal{C}_2^\perp \subset \mathcal{C}_1$ follows from the fact that the boundary of a boundary is always zero:

$$\partial_1 \circ \partial_2 = 0$$

for the complex

$$C_2 = \mathbb{F}_2^P \quad \xrightarrow{\partial_2} \quad C_1 = \mathbb{F}_2^E \quad \xrightarrow{\partial_1} \quad C_0 = \mathbb{F}_2^V$$

where C_0 is the $\mathbb{F}_2$-space with the basis given by the vertices V (0-dimensional objects), C_1 is the $\mathbb{F}_2$-space with basis elements corresponding to edges E (1-dimensional objects) and C_2 is the $\mathbb{F}_2$-space with basis elements corresponding to plaquettes P (2-dimensional objects).
Physical qubits are associated with edges, Z-type generators are associated with plaquettes and X-type generators are associated with vertices. The space of logical operators corresponds to homologically non trivial cycles (i.e. cycles that are not a boundary):

$$\mathcal{C}_1/\mathcal{C}_2^\perp = \ker \partial_1 / \mathrm{im}\, \partial_2.$$

This group is called the first homology group H_1 of the tessellation and its dimension gives the number of logical qubits encoded in the quantum code. For a closed 2-dimensional surface, the dimension of H_1 is twice the number of holes in the surface. For the toric code, it is equal to 2.

It is also useful to consider the *co-complex* of the tessellation:

$$C_2 = \mathbb{F}_2^P \quad \xleftarrow{\partial_2^T} \quad C_1 = \mathbb{F}_2^E \quad \xleftarrow{\partial_1^T} \quad C_0 = \mathbb{F}_2^V$$

where arrows are reversed and boundary operators are replaced by coboundary operators. The matrix of a coboundary operator is simply the transposed matrix of the corresponding boundary operator. Formally, the three spaces of binary vectors C_2, C_1, C_0 should be replaced by their dual space, i.e. the spaces of linear forms on C_2, C_1, C_0. However, for a finite dimensional space $\mathbb{F}_2^n$, both the space and its dual are isomorphic and we can ignore this issue.
The minimum distance of a CSS code is the minimum of two distances: $d = \min(d_X, d_Z)$ where

$$d_X = \min_{w \in \mathcal{C}_1 \setminus \mathcal{C}_2^\perp} |w|, \qquad d_Z = \min_{w \in \mathcal{C}_2 \setminus \mathcal{C}_1^\perp} |w|.$$

In particular, d_X is the minimal length of a cycle that is not a boundary. Similarly, d_z is the minimal length of a cocycle that is not a coboundary. For the toric code, it is possible to show that the distance scales like the square-root of the length of the code.

Codes obtained in this way from tessellations of 2D surfaces are one of the main approaches towards quantum fault-tolerant computing, yet they don't display very good parameters. In fact, Bravyi, Poulin and Terhal have showed that their parameters always satisfy:

$$kd^2 = O(n).$$

This is quite far from the ideal upper bound that would scale like n^3.

Finding quantum LDPC codes for which d grows significantly faster than $\sqrt{n}$ has remained completely opened for more than 20 years. A simpler task was to improve the rate k/n of the code without decreasing the distance below $\sqrt{n}$. The main new idea in this direction was the hypergraph product code construction due to Jean-Pierre Tillich and Gilles Zémor.

CSS codes from algebraic topology. Following the idea of the topological quantum code construction described above, we can see that a CSS code is in general given by a chain complex of length 3:

$$C_2 = \mathbb{F}_2^{m_Z} \quad \xrightarrow{\partial_2} \quad C_1 = \mathbb{F}_2^n \quad \xrightarrow{\partial_1} \quad C_0 = \mathbb{F}_2^{m_X}$$

where C_2, C_1, C_0 are vector spaces corresponding respectively Z-generators, physical qubits and X-generators, and such that

$$\partial_1 \circ \partial_2 = 0.$$

The classical codes of the CSS construction are given as before by

$$\mathcal{C}_1 = \ker \partial_1, \quad \mathcal{C}_2 = \operatorname{Im} \partial_2.$$

In this language, the space of Z-logical operators is again $H_1 = (\ker \partial_1)/(\operatorname{Im} \partial_2)$.

Tensor product of chain complexes. A natural way to obtain a chain complex of length 3 is to take the tensor product of two chains complexes of length 2:

$$(C_1 \xrightarrow{\partial_C} C_0) \qquad \otimes \qquad (D_1 \xrightarrow{\partial_D} D_0).$$

This gives a new chain complex

$$(C_1 \otimes D_1) \qquad \xrightarrow{\partial_2} \qquad (C_0 \otimes D_1 + C_1 \otimes D_0) \qquad \xrightarrow{\partial_1} \qquad (C_0 \otimes D_0),$$

with

$$\begin{aligned}\partial_2 &:= (\partial_C \otimes \mathrm{id}_D, \mathrm{id}_C \otimes \partial_D)\\ \partial_1 &:= \mathrm{id}_C \otimes \partial_D + \partial_C \otimes \mathrm{id}_D.\end{aligned}$$

It is immediate to check that

$$\begin{aligned}\partial_1 \circ \partial_2 &= (\mathrm{id}_C \otimes \partial_D + \partial_C \otimes \mathrm{id}_D)(\partial_C \otimes \mathrm{id}_D, \mathrm{id}_C \otimes \partial_D)\\ &= \partial_C \otimes \partial_D + \partial_C \otimes \partial_D\\ &= 0\end{aligned}$$

since we work over $\mathbb{F}_2$.

Note that a chain complex of length 2 is nothing but a classical linear code, where the boundary operator plays the role of the parity check matrix. This means that starting with two arbitrary classical codes (specified by their parity-check matrices), one obtains a quantum CSS code: this is the hypergraph product construction. Note that the terminology of hypergraph product code construction comes from the initial intuition behind the construction: the idea was to take the product of the Tanner graphs of two classical codes.

Let us change the notations a little bit and denote by $C_1 = [n_1, k_1, d_1]$ and $C_2 = [n_2, k_2, d_2]$ the two classical codes with respective parity-check matrices H_1 and H_2. We also define the transpose codes $C_1^T = [n_1 - k_1, k_1^T, d_1^T]$ and $C_2^T = [n_2 - k_2, k_2^T, d_2^T]$ that admit H_1^T and H_2^T as their parity-check matrices.

Theorem 1. *The hypergraph product code $C_1 \otimes C_2$ has the following parameters*

$$[\![n_1 n_2 + (n_1 - k_1)(n_2 - k_2), k_1 k_2 + k_1^T k_2^T, \min(d_1, d_2, d_1^T, d_2^T)]\!].$$

The toric code as an hypergraph product code. A first example of hypergraph product code is obtained by taking $C_1 = C_2$ to be a repetition code with an $n \times n$ cyclic parity-check matrix. In this case, we get $n_1 = n_2 = n$, $k_1 = k_2 = k_1^T = k_2^T = 1$ and $d_1 = d_2 = d_1^T = d_2^T = n$. One exactly recovers the parameters of the toric code $[\![2n^2, 2, n]\!]$.

Hypergraph product code of two good LDPC codes. Consider two good classical codes, that is $k_i, d_i = \Theta(n)$. If the parity-check matrices are chosen to be full rank, then

$k_1^T = k_2^T = 0$ and we get the corresponding minimum distances $d_1^T, d_2^T = \infty$. Then the resulting CSS code has parameters:

$$[\![\Theta(n^2), \Theta(n^2), \Theta(n)]\!].$$

In other words, one keeps the same distance than the toric code, but the number of logical qubits is now linear in the length instead of being constant!

Decoding hypergraph product codes. Given the remarkable parameters of the construction and its versatility, it is natural to ask whether it is possible to decode hypergraph product codes efficiently. Several solutions exist and try to mimic the decoding of classical LDPC codes. One solution is particularly efficient and comes with a decoder that can be parallelized to work in logarithmic depth: the small-set-flip decoder of quantum expander codes. Such codes even offer the possibility of quantum fault-tolerance with only a constant space overhead instead of a polylogarithmic overhead as promised by concatenation.