Quantum error correction: Lecture 3 — the toric code

At the end of the 90s, Alexei Kitaev showed that cellullations of surfaces (and of higher-dimensional manifolds) gave a very general method to derive CSS codes, with parameters depending on the properties of the surface. The most famous example is the *toric code*, which can be realized by taking a square cellullation of a torus.

Consider an $N \times N$ square grid on a torus, and put a qubit on each of the $2N^2$ edges. We define a CSS code by choosing the following generators of weight 4:

- *plaquette* operators: for each plaquette p on the grid, define $g_p^X := \bigotimes_{e \in \partial p} X_e$, where $e \in \partial p$ means that edge e belongs to the boundary of plaquette p,
- *star* operators: for each vertex v in the grid, define $g_v^Z := \bigotimes_{e \sim v} Z_e$, where $i \sim v$ means that edge e is incident to vertex v.

Let us immediately verify that these generators commute: for this, it is enough to notice that a vertex and a plaquette operator either do not overlap, or else overlap in exactly 2 positions.

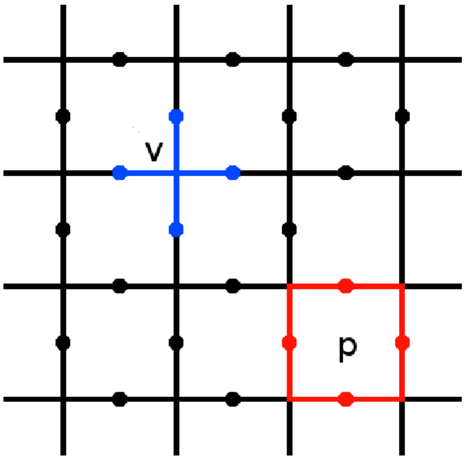

Figure 1: Local structure of the toric code: qubits are placed on edges, vertex operators are the product of X operators applied to the 4 neighboring qubits of a vertex, plaquette operators are the product of Z operators applied to the 4 qubits on the boundary of a plaquette (By James Wootton, `https://commons.wikimedia.org/w/index.php?curid=11823316`)

There are N^2 vertices on the grid and N^2 plaquettes, so we have defined $2N^2$ generators. Note, however, that these generators are not independent since the product of all vertex operators is the identity, and the product of all plaquette operators is also the identity (this is because every qubit, *i.e.* every edge, belongs to two plaquettes and to two stars):

$$\bigotimes_p g_p^X = \mathbb{1}, \quad \bigotimes_v g_v^Z = \mathbb{1}.$$

There are the only nontrivial relations, meaning that there are $2N^2 - 2$ independent generators for $2N^2$ qubits, which yields 2 logical qubits.
From our earlier definition of CSS codes, we need two classical codes $\mathcal{C}_1$ and $\mathcal{C}_2$, with $\mathcal{C}_2 \subseteq \mathcal{C}_1$:

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

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

$$\partial\partial = 0.$$

This relation is at the heart of all topological/homological quantum error correcting code constructions.

CSS codes from algebraic topology. More formally, one can define the following chain complex:

$$C_2 = \mathbb{F}_2^{N^2} \quad \xrightarrow{\partial_2} \quad C_1 = \mathbb{F}_2^{2N^2} \quad \xrightarrow{\partial_1} \quad C_0 = \mathbb{F}_2^{N^2}$$

where C_2, C_1, C_0 are vector spaces corresponding respectively to the spaces of plaquettes (Z-generators), edges (qubits) and vertices (or star, X-generators), and such that

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

The classical codes of the CSS construction are given 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 $(\ker \partial_1)/(\operatorname{Im} \partial_2)$, which is, by definition, the first homology group of the complex. In order to study the X-type logical operators, one can consider the co-complex, where the boundary operators are the transposed operators of the boundary operators of the complex:

$$C_0 = \mathbb{F}_2^{N^2} \quad \xrightarrow{\partial_1^T} \quad C_1 = \mathbb{F}_2^{2N^2} \quad \xrightarrow{\partial_2^T} \quad C_2 = \mathbb{F}_2^{N^2}.$$

This is again a valid complex since $\partial_1^T \circ \partial_2^T = 0$.
Every such chain complex of length 3 gives rise to a CSS code.

Logical operators of the toric code. In order to describe the logical qubits of the toric code, we need to understand the equivalence classes of $\mathcal{C}_1/\mathcal{C}_2$, that is, the cycles that are not a boundary. There are indeed two inequivalent families of such cycles, corresponding to the two types of loop around the torus. These cycles are *homologically nontrivial* meaning that they cannot be deformed (by addition of boundary) to yield the zero cycle. For this reason, the toric code is an example of *topological code*: properties of the quantum code result from the topology of the underlying manifold. In fact, the toric code is given by a specific cellullation of the torus, that is, a decomposition of the torus in plaquettes. The standard toric code uses square plaquettes but one could choose other types of plaquettes, for instance triangles.

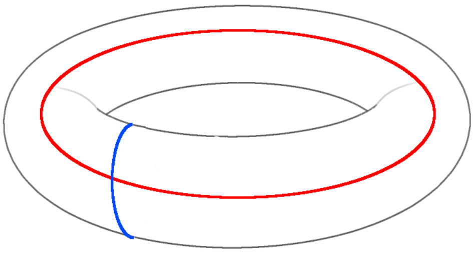

Figure 2: Local structure of the toric code: qubits are placed on edges, vertex operators are the product of X operators applied to the 4 neighboring qubits of a vertex, plaquette operators are the product of Z operators applied to the 4 qubits on the boundary of a plaquette (By James Wootton, `https://commons.wikimedia.org/w/index.php?curid=11823316`)

In particular, since the minimum size of a nontrivial cycle is N, we deduce that the minimum distance of the code is also N, and the parameters of the toric code read:

$$[[2N^2, 2, N]].$$

Another particularly interesting feature of the toric code is that it is an example of *low-density parity-check (LDPC)* code, meaning that each generator only involves a constant number of qubits (4 for the toric code) and that each qubit is only involved in a constant number of generators (4 again for the toric code). This LDPC condition is particularly important when it comes to experimental implementation since measuring the syndrome for the toric code will only require small circuits involving at most 4 physical qubits for each bit of the syndrome. In fact, the leading approaches to build a quantum computer are based on the toric code (or its cousin the surface code).

Despite an intensive study of quantum LDPC codes, it turned out to be extremely difficult to find better LDPC codes than the toric codes. For about 20 years, the best bound for the minimum distance was $n^{1/2}\log^{1/4} n$. In 2020, a series of papers showed that $d_{\min} = \Theta(n/\log n)$ is achievable! This was quickly followed by 2 major breakthroughs: Hastings, Haah and O'Donnell showed how to get $d_{\min} \approx n^{3/5}$, and then Panteleev and Kalachev achieved a distance $n/\log n$. Finally, in November 2021, Panteleev and Kalachev proved the existence of asymptotically good quantum LDPC codes with dimension $k = \Theta(n)$ and distance $d_{\min} = \Theta(n)$. We will discuss some examples of codes better than the toric code in the next lecture.

Decoding the toric code.

Let us first consider X-type errors. Their associated syndrome is obtained via the Z-type generators, corresponding to vertices. Let X^E be an X-type error with support on the set E. Then the syndrome of X_E is given by the set of vertices in the *boundary of* E:

$$s(X^E) = \partial E.$$

In order to decode, one must therefore find an error with the appropriate syndrome, that is to find a pattern of small weight with a given boundary, such that this pattern differs from the true error by a sum of generators: they should differ by a boundary. A particularly popular decoder is the algorithm *minimum weight perfect matching.* This is not optimal, however, since the most probable error isn't necessarily the error of minimum weight, but rather the error whose equivalence class is the most probable. However, the minimum weight perfect matching algorithm is efficient. More precisely, its complexity scales like the cube of the number of qubits, *i.e.* like N^6. While this is efficient in the computer science sense, that is polynomial time, this is far from fast, and alternative decoders have been devised which have a complexity scaling linearly with the number of qubits, but slightly worse efficiency (that is, they cannot correct errors as large as the minimum weight perfect matching decoder).

In order to address Z-type errors, it is convenient to exploit *Poincaré duality*: the dual of the cellulation looks exactly like the initial cellulation, but with the roles of vertices and plaquettes exchanged. In other words, one can correct Z-type errors with the same approach as explained for the X-type errors, but working in the dual cellulation.

The threshold of the toric code is about $10-11\%$ for the depolarizing channel, corresponding to an error model where X, Y, Z errors occur independently with probability $p/3$ and not error occurs with probability $1-p$. This means that for error rates below the threshold, increasing the code size will result in better (lower) logical error rate (after optimal decoding).

Beyond the toric code. This approach isn't limited to tessellations of the torus, but can be generalized in a straightforward way to cellullations of arbitrary closed manifolds in arbitrary dimensions. Even dimensions are convenient to exploit the Poincaré duality, and this is why the 4-dimensional toric code is also quite popular. One can also change the geometry and work with hyperbolic geometry rather than Euclidean space. In particular, hyperbolic geometry is very useful to get codes with a large dimension. It is then possible to define quantum codes with a linear dimension $k = \Theta(n)$ and with a polynomial distance $d_{\text{min}} = \Omega(n^{1/5})$, which is not possible in Euclidean geometry.