Quantum Coding Theory **(UC Berkeley CS294, Spring 2024)**

Lecture 20: Quantum Tanner Codes II

April 5, 2024

Lecturer: John Wright *Scribe: Jack Spilecki*

Today, we continue to study quantum Tanner codes, following the construction given by Leverrier and Zémor [LZ22a, LZ22b].

1 Recap

It will be useful to review the construction from last time. Begin with a group G, and let $A = A^{-1}$ and $B = B^{-1}$ be sets of generators for G, selected so that $|A| = |B| = \Delta$. From these generators, we constructed the *left-right Cayley complex*, the following quadripartite graph:

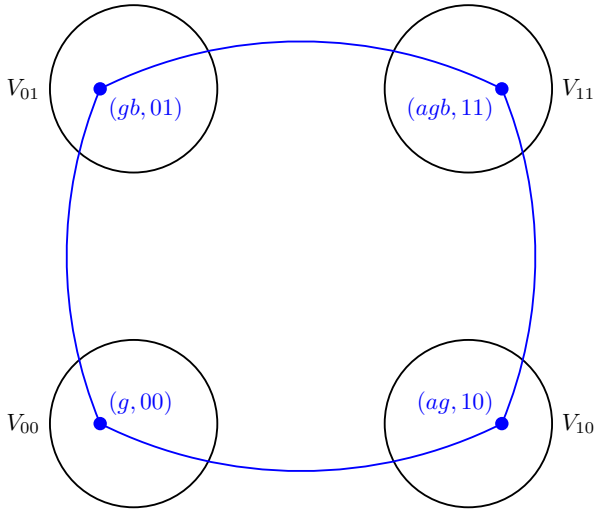

Figure 1: The left-right Cayley complex is a quadripartite graph, with vertices $V = V_{00} \sqcup V_{01} \sqcup V_{10} \sqcup V_{11}$, where $V_{ij} = G \times (i, j)$. We have edges as indicated in the figure. The horizontal edges, corresponding to elements $a \in A$, are *A-edges*, and the vertical edges, corresponding to elements $b \in B$, are *B-edges*.

The most important objects in this graph are the *squares*, the sets of vertices of the form $\{(g, 00), (ag, 10), (gb, 01), (agb, 11)\}$, connected by edges as in the figure. The set of all squares is Q. For every vertex $v \in V$, we define the *Q-neighborhood of v*, denoted $Q(v)$, as the set of squares v is adjacent to. Since v is adjacent to one element for each $a \in A$, and each $b \in B$, we have $Q(v) \cong A \times B$. We can represent a Q-neighborhood in the following way:

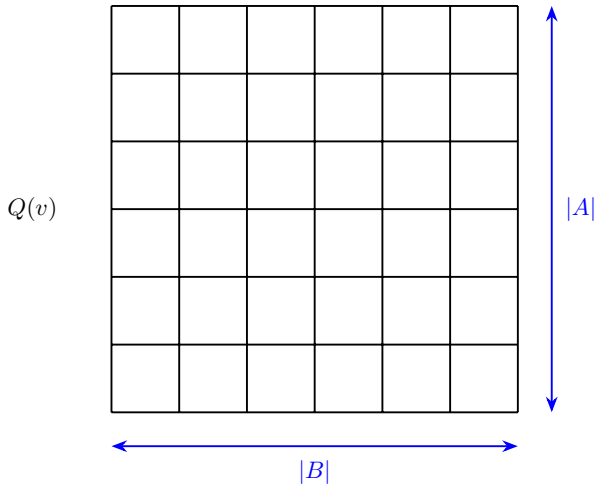

Figure 2: For any vertex $v \in V$, $Q(v)$ can be arranged into an $|A| \times |B|$ grid of squares.

If we define the Q-neighborhoods appropriately (that is, the mapping from $A \times B \to Q(v)$) then we saw that the Q-neighborhoods for neighboring vertices are very nice: in particular, the Q-neighborhoods for vertices connected by an A-edge share their ath rows (and no other squares), and the Q-neighborhoods for vertices connected by an B-edge share exactly their bth columns (and no other squares).

Remark 1.1. Relative to our last lecture, we have swapped the positions of V_{01} and V_{10} in the left-right Cayley complex diagram. This is just because it's pictorially nicer: horizontal edges in the Cayley complex (resp. vertical edges) correspond to Q-neighborhoods that share rows (resp. columns), as in Figure 3.

We also noted that the left-right Cayley complex is a a combination of four Cayley graphs: the induced subgraphs on $V_{00} \sqcup V_{10}$, $V_{10} \sqcup V_{11}$, $V_{01} \sqcup V_{11}$, and $V_{00} \sqcup V_{01}$ are each individually Cayley graphs. The first and third (those with A-edges) are isomorphic to (the double cover of) $\mathrm{Cay}_L(G, A)$, and the second and fourth (those with B-edges) are isomorphic to (the double cover of) $\mathrm{Cay}_R(G, B)$.

On these graphs, we defined two quantum codes, starting from two classical codes. Take C_A and C_B to be classical linear error correcting codes on $|A|$ and $|B|$ bits respectively. We ultimately want to define a CSS code, and for this we need to define an X-code and a Z-code.

Our X-code, C_0, is defined as follows:

- Bits: on squares of Q.
- Local code: for each $v \in V_{00} \sqcup V_{11}$, the bits on $Q(v)$ are elements of $C_A^\perp \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B^\perp$. An element of $C_A^\perp \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B^\perp$ can be written as $c + r$, where $c \in C_A^\perp \otimes \mathbb{F}_2^B$ and $r \in \mathbb{F}_2^A \otimes C_B^\perp$. Therefore, c is a matrix whose columns come from $C_A^\perp$, and r is a matrix whose rows come from $C_B^\perp$. The matrices c and r come from natural codes, but the code $C_A^\perp \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B^\perp$ asks that a neighborhood $Q(v)$ is an overlapping of c and r,

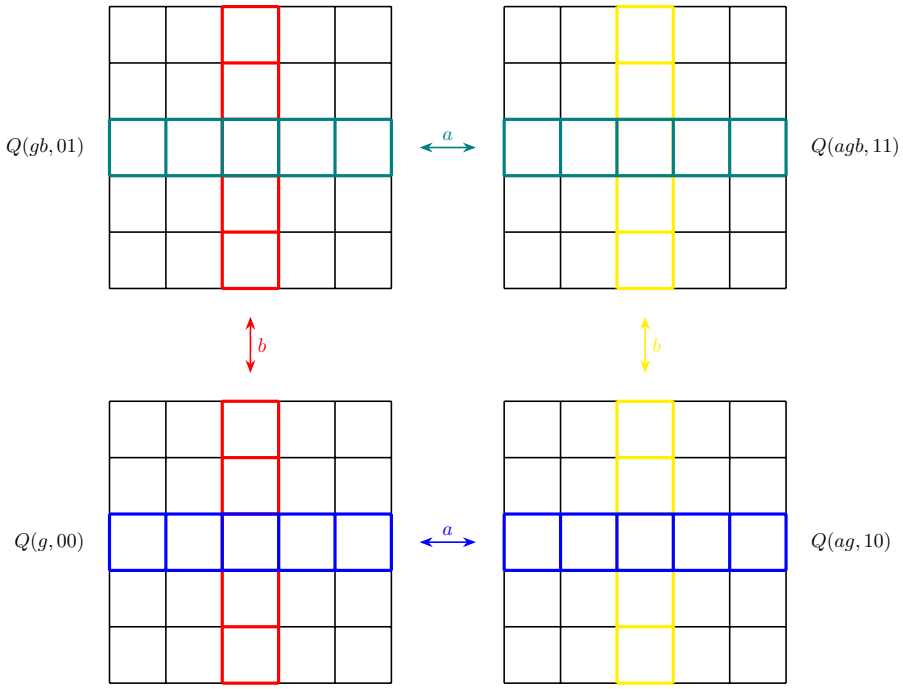

Figure 3: We can arrange the squares in Q-neighborhoods so that if v is adjacent to u by an A-edge, labelled a, then $Q(v)$ and $Q(u)$ share their ath rows, and so that if v is adjacent to u by a B-edge, labelled b, then $Q(v)$ and $Q(u)$ share their bth columns.

and in some sense "sees the two codes simultaneneously" and those codes may conflict with each other in ways we will have to deal with later.

– Parity checks: for each $v \in V_{00} \sqcup V_{11}$, the constraints (on the bits of $Q(v)$) are elements of $C_A \otimes C_B$.

We saw that this is a Tanner code on the graph $\mathcal{G}_0^{\square}$, which has vertices $V_{00} \sqcup V_{11}$, and edges for each square (that is, an edge between $(g, 00)$ and $(agb, 11)$ for each $a \in A$ and $b \in B$, i.e. between opposite vertices of squares):

$$C_0 = \mathrm{Tan}(\mathcal{G}_0^{\square}, C_A^{\perp} \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B^{\perp}).$$

Remark 1.2. Nothing stops us from having two vertices being part of multiple squares. So, the graph $\mathcal{G}_0^{\square}$ could have multiple edges between vertices. We came up with examples of this last time (e.g. take G abelian, and $A = B$, so that $agb = bga$, and therefore $(g, 00)$ and $(agb, 11)$ share at least two edges). We can pick G, A and B so that two vertices are part of at most one square if we want, but it's not necessary.

The Z-code, C_1, is similar:

– Bits: on squares of Q.

– Local code: for each $v \in V_{01} \sqcup V_{10}$, the bits on $Q(v)$ are elements of $C_A \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B$. As in the X-code, the matrix of bits on a square $Q(v)$ can be written as $c + r$, where $c \in C_A \otimes \mathbb{F}_2^B$ is a matrix whose columns come from C_A, and $r \in \mathbb{F}_2^A \otimes C^B$ is a matrix whose rows come from C_B.

– Parity checks: for each $v \in V_{01} \sqcup V_{10}$, the constraints (on the bits of $Q(v)$) are elements of $C_A^\perp \otimes C_B^\perp$.

Then C_1 is also a Tanner code on the graph $\mathcal{G}_1^\square$, similar to $\mathcal{G}_0^\square$ except on $V_{01} \sqcup V_{10}$:

$$C_1 = \mathrm{Tan}(\mathcal{G}_1^\square, C_A \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B).$$

2 Quantum Tanner Codes are CSS Codes

Consider an X-check on a vertex $u \in V_{00}$. Every X-check h_x on $Q(u)$ is an element of $C_A \otimes C_B$. So, each column of h_x is in C_A, and every row of h_x is in C_B. Also, consider a Z-check on a vertex $v \in V_{01}$. Every Z-check h_Z on $Q(v)$ is an element of $C_A^\perp \otimes C_B^\perp$: each column of h_z is in $C_A^\perp$, and every row of h_z is in $C_B^\perp$.

In order for C_0 and C_1 to define a CSS code, we need the parity checks of C_0 and C_1 to be orthogonal. So, we want to show $h_x \cdot h_z = 0$. There are two cases:

1. h_x and h_z overlap on no squares: $Q(u) \cap Q(v) = \varnothing$. Since h_x is only nonzero on $Q(u)$, and h_z is only nonzero on $Q(v)$, $h_x \cdot h_z = 0$.

2. $Q(u) \cap Q(v) \neq \varnothing$. Then u and v are joined by a B-edge, and $Q(u) \cap Q(v)$ overlap on one column, and the remaining squares are not shared. On this column, h_x is an element of C_A, and h_z is an element of $C_A^\perp$. So $h_x \cdot h_z$, which only has a possibly nonzero contribution from this column, is zero. See Figure 4.

Other pairs of X-checks and Z-checks have zero dot product for similar reasons. For example, an X-check coming from V_{00} and a Z-check coming from V_{10} will act on Q-neighborhoods that share a row of squares, and will dot to zero, since one of these rows of bits is an element of C_B and the other an element of $C_B^\perp$.

We have therefore shown:

Fact 2.1. *C_0, C_1 form a CSS code.*

3 Parameters for Quantum Tanner Codes

Now that we have a CSS code, we want to know: how good is this code? What is its rate and distance?

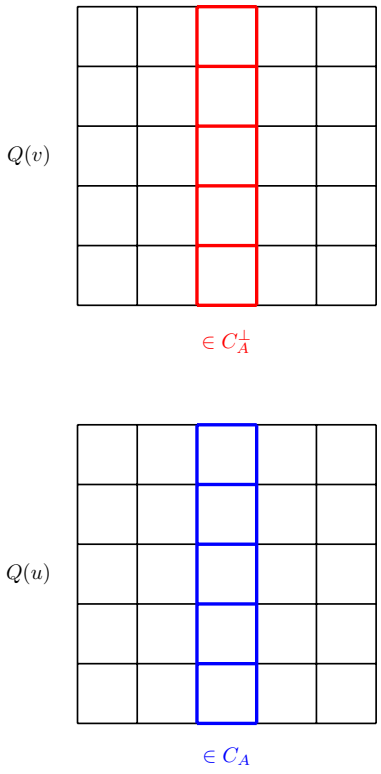

Figure 4: If u and v are joined by a B-edge labelled b, then $Q(u)$ and $Q(v)$ share the squares in their bth row. Since the columns of $Q(u)$ are elements of C_A, and the columns of $Q(v)$ are elements of $C_A^\perp$, these two columns are orthogonal.

3.1 Rate

Recall that $|A| = |B| = \Delta$. In the left-right Cayley complex, we have $|V| = 4|G|$ vertices, and $|Q| = |G||A||B| = |G|\Delta^2$ squares. For our code, which is defined on the squares, $n = |Q| = |G|\Delta^2$.

We have to pick C_A and C_B appropriately if we want a good quantum Tanner code. We are going to pick $C_A = [\Delta, \rho\Delta]$, where $\rho \in (0, 1)$ is a constant, and $C_B = [\Delta, (1-\rho)\Delta]$. This second choice looks a little funny, but will make expressions we see shortly symmetric.

How many independent parity checks do we have in our CSS code? This will allow us to compute the dimension of the CSS code. For the X-code, every vertex imposes parity checks from $C_A \otimes C_B$. Since

$$\dim(C_A \otimes C_B) = \dim(C_A)\dim(C_B) = \rho\Delta \cdot (1-\rho)\Delta = \rho(1-\rho)\Delta^2,$$

this is the number of independent parity checks in the X-code, for a given vertex. For the Z-code, parity checks come from $C_A^\perp \otimes C_B^\perp$, and

$$\dim(C_A^\perp \otimes C_B^\perp) = \dim(C_A^\perp)\dim(C_B^\perp) = (\Delta - \dim(C_A))(\Delta - \dim(C_B)) = (1-\rho)\Delta \cdot \rho\Delta = \rho(1-\rho)\Delta^2.$$

We've used here that C_A and C_B are each codes on Δ bits. Note that our choice for the dimension of C_B makes the number of parity checks of each type equal. The overall

number of X-checks in C_0 is then at most $2|G| \cdot \rho(1-\rho)\Delta^2$, since there are $2|G|$ vertices in $\mathcal{G}_0^{\square}$, and each vertex provides $\rho(1-\rho)\Delta^2$ independent checks, though these may not all be independent. Similarly, there are at most $2|G|\rho(1-\rho)\Delta^2$ independent Z-checks in C_1. Finally, the dimension of the CSS code is

$$\begin{aligned} k &= n - \ell_X - \ell_Z \\ &\geq |G|\Delta^2 - 4|G|\rho(1-\rho)\Delta^2 \\ &= (1 - 4\rho(1-\rho))|G|\Delta^2 \\ &= (1-2\rho)^2 n. \end{aligned}$$

So long as $\rho \neq 1/2$, the CSS code has constant rate.

3.2 Locality

Each parity check only involves a single Q-neighborhood, and each Q-neighborhood has size Δ^2. Moreover, each qubit, which corresponds to a square, is part of at most $4\rho(1-\rho)\Delta^2$ parity checks. This is because a square has four vertices, and hence is in four Q-neighborhoods. For each of these neighborhoods, there are at most $\rho(1-\rho)\Delta^2$ checks that may act on that square. Since $4\rho(1-\rho) \leq 1$, there are therefore Δ^2 checks a qubit may be involved in. So long as Δ is constant, the CSS code has constant locality.

Remark 3.1. If we really care about the locality of the code, the first construction of QLDPC codes, by Panteleev and Kalachev [PK22], has locality Δ instead of Δ^2.

3.3 Distance

Just like with classical Tanner codes, if all we care about is the rate of Tanner code, then all that matters is the rate of the inner code. The distance of the inner code only comes into play now that we are computing the distance of the Tanner code.

We will assume that we have picked C_A, C_B, $C_A^{\perp}$ and $C_B^{\perp}$ so that they all have good distance. In particular, they all have distance at least $\delta\Delta$, for some nonzero constant δ. We have seen previously that for random codes, both the code and its dual have good distance.

However, as in the case of classical Tanner codes, it is not sufficient that the inner code has good minimum distance – we also need that the graph has good expansion properties. For this, we will need to pick G, A and B so that $\mathrm{Cay}_L(G, A)$ and $\mathrm{Cay}_R(G, B)$ are good expanders – in particular, so that they are Ramanujan. Here this means, for either graph, $\lambda \leq 2\sqrt{\Delta}$, since Δ is the degree of each graph.

However, what we really care about (at least if we were constructing a classical Tanner code) is the expansion properties of the graphs on which we define the Tanner code. Neither of these Cayley graphs are the Tanner code graphs, which are instead $\mathcal{G}_0^{\square}$ and $\mathcal{G}_1^{\square}$. What is the expansion of these graphs?

Fact 3.2. *If* $\lambda(\mathrm{Cay}_L(G, A)), \lambda(\mathrm{Cay}_R(G, B)) \leq 2\sqrt{\Delta}$, *then* $\lambda(\mathcal{G}_0^{\square}), \lambda(\mathcal{G}_1^{\square}) \leq 4\Delta = 4\sqrt{\Delta^2}$. *Since* Δ^2 *is the degree of these graphs,* $\mathcal{G}_0^{\square}$ *and* $\mathcal{G}_1^{\square}$ *are* almost Ramanujan.

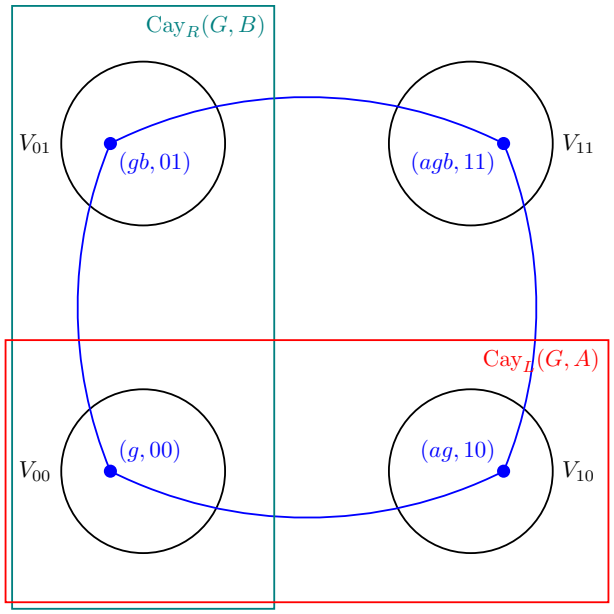

Figure 5: The subgraph on $V_{00} \sqcup V_{10}$ (red) is a copy of (the double cover of) $\mathrm{Cay}_L(G,A)$ (and so is the subgraph on $V_{01} \sqcup V_{11}$). Similarly, the subgraph on $V_{00} \sqcup V_{01}$ (teal) is a copy of (the double cover of) $\mathrm{Cay}_R(G,B)$ (and so is the subgraph on $V_{10} \sqcup V_{11}$).

This fact may appear as a homework problem. For some intuition: edges in $\mathcal{G}_0^{\square}$ correspond to a choice of an edge in $\mathrm{Cay}_L(G,A)$ and an edge in $\mathrm{Cay}_R(G,B)$. The corresponding adjacency matrices commute, and so the eigenvalues multiply. The second largest (absolute value of an) eigenvalue is then at most $2\sqrt{\Delta} \cdot 2\sqrt{\Delta} = 4\Delta$.

For classical Tanner codes, this is all we need. For quantum Tanner codes, we will need another property. This deals with the fact that we have one new feature in these codes, which is that we seemingly have two codes overlapping and intersecting with each other, and we want to know how these codes relate to each other when they overlap.

Let's look at this property in terms of the Z-code, $C_1 = \mathrm{Tan}(\mathcal{G}_1^{\square}, C_A \otimes \mathbb{F}_2^B + F_2^A \otimes C_B)$. Let x be a codeword. For each $v \in V_{01} \sqcup V_{01}$, we can restrict x to $Q(v)$ to get a matrix of bits $x_v = c_v + r_v$, where c_v is a matrix whose columns are in C_A, and r_v is a matrix whose rows are in C_B.

Individually, we can understand c_v and r_v, but what happens when we add them together? In our analysis, we would love it if we really didn't have to care about this much – that it would still look like columns came from C_A, and rows from C_B.

Let's write $||c_v||$ for the number of nonzero columns in c_v, and $||r_v||$ for the number of nonzero rows in r_v. Then $|c_v| \geq \delta\Delta||c_v||$, where $|c_v|$ is the number of nonzero entries in c_v, since for each nonzero column, there must be at least $\delta\Delta$ nonzero bits, as $\delta\Delta$ is the distance. Similarly, $|r_v| \geq \delta\Delta||r_v||$.

A heuristic, which would be great to be able to use, is that

$$|x_v| \approx |c_v| + |r_v| \geq \delta\Delta(||c_v|| + ||r_v||).$$

If we could do this, then we could analyze our code as if the c_v's and r_v's were separate. This heuristic isn't quite true, though. For example, if c_v and r_v are nonzero in a lot of the same locations, $|x_v|$ can be much smaller than $|c_v| + |r_v|$.

So what we want is a code where we can actually instantiate this heuristic. A code of this form is called a *κ-product expanding code*.

Definition 3.3. A code $C = C_A \otimes \mathbb{F}_2^B + F_2^A \otimes C_B$ is *κ-product expanding* if any codeword x has a decomposition $x = c + r$ such that

$$|x| \geq \kappa\Delta(||c|| + ||r||).$$

From our heuristic, we would like κ close to δ. That's not always going to be the case, but we will be able to get codes where κ is a reasonably large constant.

How do we get product expanding codes? Intuitively, this property *doesn't* hold when there's amazing structure in C_A and C_B so that we get a lot of cancellation. A natural thing to try then is to pick random codes C_A and C_B, and hope they are product expanding. This was indeed shown by Pantaleev and Kalachev:

Theorem 3.4. *[PK22] Pick $C_A = [\Delta, \rho\Delta]$ and $C_B = [\Delta, (1-\rho)\Delta]$ uniformly at random. The code $C_A \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B$ is κ-product expanding with high probability as $\Delta \to \infty$ for*

$$\kappa = \frac{1}{2} \min\left(\frac{1}{4} H_2^{-1}\left(\frac{\rho}{8}\right)^2, H_2^{-1}\left(\frac{\rho^2}{8}\right)\right).$$

Here, $H_2(p) = -p\log_2 p - (1-p)\log_2(1-p)$ is the binary entropy (on $[0, 1/2]$, so that it is invertible).

Remark 3.5. In our dream scenario, $\kappa \approx \delta$. Recall from Homework 2 that the distance of the uniformly random code C_B is about $H_2^{-1}(\rho)$ with high probability. So κ is similar-looking to the distance for these random codes.

Remark 3.6. In a recent paper by Dinur, Lin and Vidick [DLV24], the authors generalized these good QLDPC codes to give the best known locally testable codes. When they originally wrote the paper, their theorem was true assuming a conjecture similar to our product expansion statement. This statement was later proven, also by Panteleev and Kalachev [PK24], for random codes. Whenever we are dealing with QLDPC codes, locally testable codes, or other codes in this family, these product expansion-type properties allow us to make our analysis, and so there is currently a lot of activity in proving expansion properties of random codes.

A The Toric Code as a Quantum Tanner Code

Previously, we saw how the toric code is an example of a hypergraph product code. Here, we show it is also an example of a quantum Tanner code.

Take $G = \mathbb{Z}_N \times \mathbb{Z}_N$. The elements of G will roughly correspond to position on the torus. We will take $A = \{(+1, 0), (-1, 0)\}$ and $B = \{(0, +1), (0, -1)\}$.

A square then consists of four vertices: $(a, b, 00)$, $(a \pm_1 1, b, 10)$, $(a, b \pm_2 1, 01)$ and $(a \pm_1 1, b \pm_2 1, 11)$, where $\pm_i$ represents a consistent choice of $+$ or $-$. The Q-neighborhood of a vertex v consists of four squares.

Recall that the CSS code is defined so that qubits lie on the squares. Geometrically, we can view this as qubits placed on the "00-11" diagonals of squares in the left-right-Cayley complex, or alternatively, the edges in the graph $\mathcal{G}_0^{\square}$. These diagonals then lie themselves on a square lattice, rotated relative to the original lattice. This will end up being the primal lattice for the toric code.

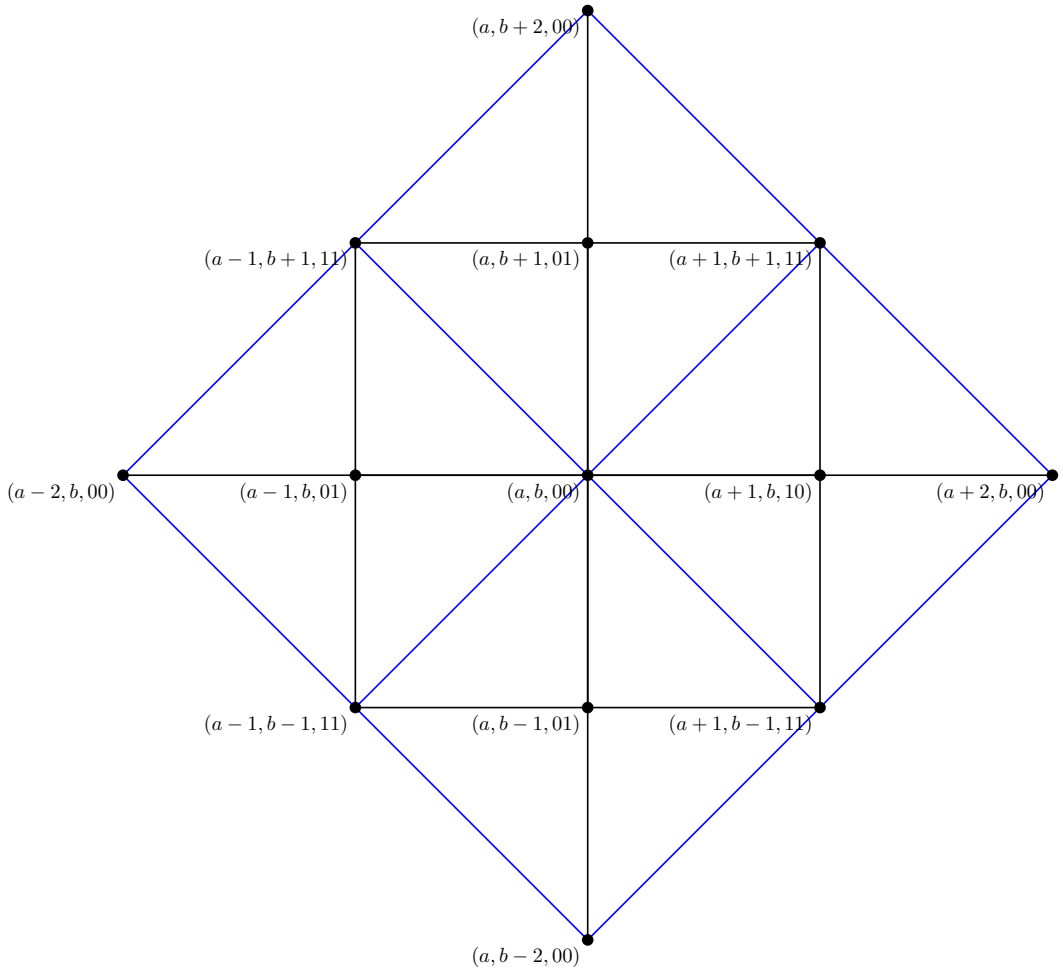

Figure 6: The black points (resp. edges) represent the vertices (resp. edges) in the left-right Cayley complex. The blue edges are the "00-11" diagonals of squares in the complex, or alternatively edges in the graph $\mathcal{G}_0^{\square}$, on which we place our bits/qubits, and which we will see forms the primal lattice for the toric code. The squares in the black lattice are the elements of Q. For example, the four squares in the figure form the Q-neighborhood of $(a, b, 00)$.

The inner code is a code on 2×2 matrices, so a natural choice is to pick $C_A = C_B =: C$, the repetition code on two bits. This codes is self dual: $C = C^\perp$. The elements of $C_A \otimes C_B = C_A^\perp \otimes C_B^\perp$ are

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \text{ and } \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

This is because if there is a nonzero entry, then every entry in its row and column must be a 1, since each row and column must belong to C. The elements of $(C_A \otimes C_B)^\perp = C_A^\perp \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B^\perp = C_A \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B$ then consist of all 2×2 matrices with an even number of 1's.

The X-code looks at the bits on $Q(v)$, for each $v \in V_{00} \sqcup V_{11}$, and enforces that these bits are elements of $C_A \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B$. In the figure above, this translates to taking a point that coincides with a vertex of the blue lattice, and checking that there is an even number of flipped bits on the blue edges adjacent to this vertex.

The Z-code looks at bits of $Q(v)$, for $v \in V_{01} \sqcup V_{10}$, and checks that these bits are also elements of $C_A \otimes \mathbb{F}_2^B + \mathbb{F}_2^A \otimes C_B$. In the figure above, v is a point at the center of a blue plaquette, and we can see that this means checking that an even number of edges bordering a single plaquette in the primal lattice are flipped.

The X-code and Z-code therefore act exactly as in a surface code! That is, the quantum Tanner code for this choice of group, generators, and inner code, is a surface code on a surface tiled by squares.

The surface is either a torus or a pair of tori, depending on the parity of N. We can see this by considering a path starting at $(a, b, 00)$ and continuing $(a, b, 00) \to (a+1, b+1, 11) \to (a+2, b+2, 00) \to \ldots$, alternating between V_{00} and V_{11}.

If N is odd, then after N steps we're at the vertex $(a, b, 11)$ rather than $(a, b, 00)$, so it takes $2N$ steps in a single direction to return to our starting point. The same holds for the transverse direction, and moreover, we can reach any point in $V_{00} \cup V_{11}$ from any other. So for odd N, we end up with a single torus of size $2N \times 2N$.

For N even, after taking only N steps in any fixed direction we get back to where we started. But we're unable to get to half of the points! For example, we cannot get to $(a, b, 11)$ from $(a, b, 00)$ taking blue edges (since we have to take an even number of steps to get from $(a, b, ii) \to (a, b, jj)$, but after an even number of steps, $j = i$). We end up with two tori as a result.

References

[DLV24] Irit Dinur, Ting-Chun Lin, and Thomas Vidick. Expansion of higher-dimensional cubical complexes with application to quantum locally testable codes, 2024. 3.6

[LZ22a] Anthony Leverrier and Gilles Zémor. Decoding quantum tanner codes, 2022. (document)

[LZ22b] Anthony Leverrier and Gilles Zémor. Quantum tanner codes, 2022. (document)

[PK22] Pavel Panteleev and Gleb Kalachev. Asymptotically good quantum and locally testable classical ldpc codes, 2022. 3.1, 3.4

[PK24] Pavel Panteleev and Gleb Kalachev. Maximally extendable sheaf codes, 2024. 3.6