Hypergraph Product Codes

(Tillich + Zemor 0903.0566)

From any two classical codes, construct a CSS code. Why hypergraph? A graph $G = (V, E)$ is a set of vertices, and a set of edge = pairs of vertices. In a hypergraph, elements of E are not pairs of edges, but arbitrary subsets of V.

Codes + hypergraphs. Parity check matrix H defines a hypergraph. If $r \times n$, each of the r rows defines a subset of the n bits (vertices), those where the matrix has 1 rather than 0 (r subsets)

Another hypergraph: Each of the n columns defines a subset of the r checks (vertices), those where matrix has 1.

One hypergraph: sets of bits participating in a check

Other hypergraph: set of checks involving a bit

The hypergraph product puts together the hypergraphs of two classical codes to get a new code.

The construction of the HGP uses the concept of the transpose of a code. If C has parity check H, then C^T has parity check H^T. The roles of checks + bits are interchanged. C^T depends not just on C, but also on how C is described by H.

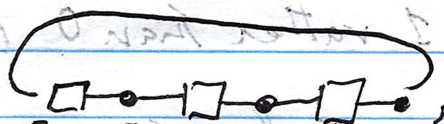
20 Apr 2026

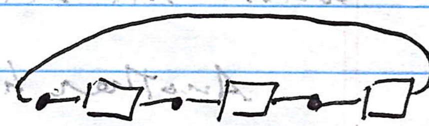
A nice way to visualize the transpose: The ~~Tan~~ Tanner graph. This is a bipartite graph, with two types of vertices — variable nodes & check nodes. Each edge connects a variable node to a check node.

Each check node connects to all bits in the check: row of H
Each variable node connects to all checks it participates in: column of H

$H \to H^T$: the variable node become checks, the check nodes become variables

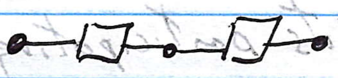
Example: Repetition code on a cycle.

$H =$ 

, $H^T =$ 

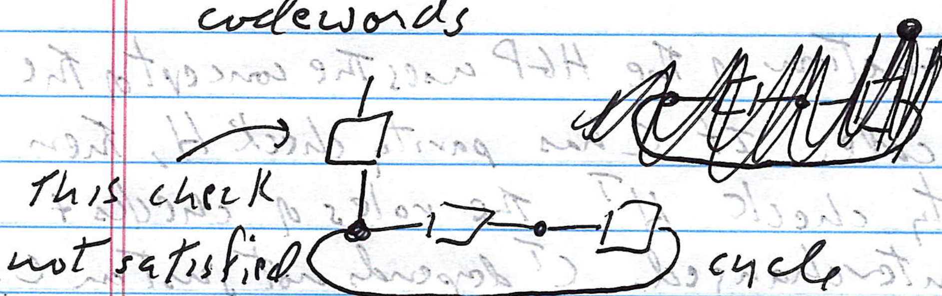
Same code as H!

What if not "periodic boundary condition"?

$H =$  $H^T =$ 

3-bit rep code — Trivial 2 bit code

Remark: Tanner graph might have ~~lots~~ short cycles, but these need not correspond to low-weight codewords



All checks and variables have constant degree → LDPC

As for BB codes, there are two qubit registers, and the code is defined by two parity check matrices:

A is $r_1 \times n_1$ and B is $r_2 \times n_2$

These act on the two registers in X and Z check according to

$$H_X = \left(A \otimes I_{n_2},\ I_{r_1} \otimes B^T\right)$$

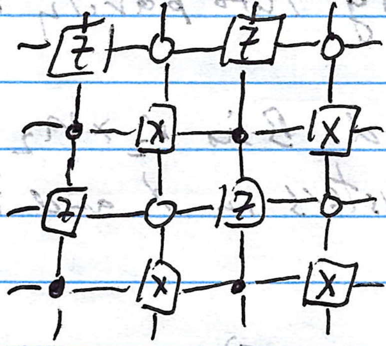
$$H_Z = \left(I_{n_1} \otimes B,\ A^T \otimes I_{r_2}\right)$$

register 1: $n_1 n_2$ qubits   register 2: $r_1 r_2$ qubits

Both registers are arranged in a grid, with "horizontal" and "vertical" directions. In H_X, the A check acts on qubits in the same "row" and B^T on qubits in the same "column" (This row and column characterize the grid, not the parity check matrix.) Furthermore A acts on first register and B^T on second register. The H_Z matrix is chosen so that for each collision between H_X and H_Z in register 1, there is a corresponding collision in register 2. Therefore the number of collisions is even, and the checks commute:

$$H_X H_Z^T = \underbrace{A \otimes B^T}_{\text{1st register}} + \underbrace{A \otimes B^T}_{\text{2nd register}} = 0$$

Example: 2×2 toric code, the HGP of two $n=2$ repetition codes, with $H = \begin{pmatrix}1&1\\1&1\end{pmatrix} = H^T$

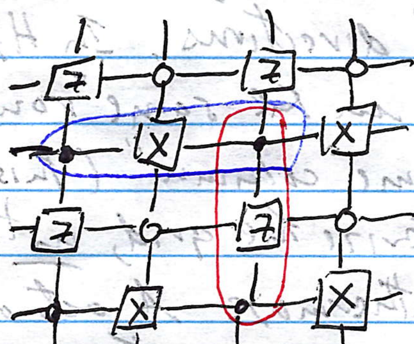


(Tanner graph, periodically identified)

2×2 grid of black nodes
2×2 grid of white nodes

A: X checks couple black nodes on same row of grid
B^T: X checks couple white nodes in same column

B: Z checks couple black nodes in same column
A^T: Z checks couple white nodes in same row



By construction, if X and Z checks collide at a black node, there is also a corresponding collision on a white node

In the HGP code the length n is the number of columns in H_X (or H_Z). This is

$$n = n_1 n_2 + r_1 r_2$$

variable-variable register w/ $n_1 n_2$
check-check register w/ $r_1 r_2$

How many X checks? Number of rows in H_X
$= r_1 n_2$

How many Z checks? Number of rows in H_Z
$= n_1 r_2$

20 Apr 2026

Therefore, the number of logical qubits is

$$K \geq \underbrace{n_1 n_2 + r_1 r_2}_{n} - \underbrace{n_2 r_1 - n_1 r_2}_{\#\text{ of checks}} = (n_1 - r_1)(n_2 - r_2)$$

K can be larger if checks are redundant.
E.g. for the Toric code = HGP of rep codes with $n = r$ but 1 redundant check; we actually have $K = 2$.

If A and B both have full rank (linearly independent rows), they are parity checks for codes with $K_1 = (n_1 - r_1)$ and $K_2 = (n_2 - r_2)$ and we have $K = K_1 K_2 \Rightarrow$ code rate is

$$R = \frac{K_1 K_2}{n_1 n_2 + r_1 r_2} \geq \frac{K_1 K_2}{2 n_1 n_2} = \frac{1}{2} R_1 R_2$$

The HGP of two constant rate (="high-rate") codes is a ~~constant rate~~ high-rate quantum code. (For check weights of quantum code, see below)

~~In this case we can choose logical operators to act only on the first register. A Z-logical operator commutes with X stabilizer if corresponding bit string is annihilated by A. It crosses the grid "horizontally." Therefore it can't be in the Z stabilizer acting on register 1 because the operators in the Z stabilizer cross the grid "vertically." We need the second register, though, for the Z and X checks to commute!~~

~~However, we're interested in LDPC codes for which checks are redundant. That means A^T and B^T are parity checks for nontrivial codes, and there are logical operators acting only on the second register.~~

We can choose $K_1 K_2$ logical operators of X type, each acting on just the first register such that each is $u \otimes v$ where $u \in \mathrm{Ker}\, A$ and $v \in \mathrm{Ker}\, B$. Here $v \in \mathrm{Ker}\, B$ is condition to commute with H_Z, and $\mathrm{Ker}\, A = 0$ ensures that no two differ by an element of X stabilizer. Likewise, there are $K_1^T K_2^T$ X logical operators supported on second register where each has the form $u \otimes v$ $u \in \mathrm{Ker}\, A^T$ and $v \in \mathrm{Ker}\, B^T$. These are all the logical ops

$$K = K_1 K_2 + K_1^T K_2^T$$

But what about distance?

$$d = \min(d_1, d_2, d_1^T, d_2^T)$$

Codewords of this weight exist of the form

$$\nabla = \begin{pmatrix} 0 & \begin{matrix} u_1 \\ \vdots \\ u_{n_1} \end{matrix} & 0 \end{pmatrix} \qquad \nabla = \begin{pmatrix} 0 \\ v_1 \cdots v_{n_2} \\ 0 \end{pmatrix}$$

Support on a <u>single strip</u>, code word of classical code (or its transpose)

7

20 Apr 2026

We're interested in LDPC codes which typically have highly redundant checks, so if we want distance $d = \Omega(\sqrt{n})$ we'll need our codes ~~to have codes~~ + their transposes to have distance $\Omega(n_1, n_2)$. Hence A, B, A^T, B^T all are parity check matrices of good codes.

Idea: Tanner graph is <u>double-sided expander</u>

what are check weights

X checks: $wt(A\ row) + wt(B\ column)$

Z checks: $wt(B\ row) + wt(A\ column)$

This means sets of variables (if not too large) trigger many checks for both H and H^T

Tanner graph $G = (V \cup C, E)$ — variables, checks

An (s, α) expander if any set of S variables with $|S| \leq s$ has at least $\alpha |S|$ check neighbors.

Suppose variable nodes have $\leq \omega_{max}$ neighbors (degree) and suppose $\alpha > \omega_{max}/2$. The ~~...~~ errors on bits in S with $|S| \leq s$ has to trigger at least one check

20 Apr 2026

(hence not a codeword). Why? If all checks are satisfied, then each check has even # of variable error neighbors. If all checks with error neighbors have at least 2 neighbors (true for a codeword), then total # of neighbors $\leq \frac{w_{max}}{2}|S|$.
Therefore, since expansion $\Rightarrow > \frac{w_{max}}{2}|S|$ neighbors, at least one check has a single neighbor.

If our (s, α) expander also has $s = \Omega(n)$, then codeword weight $> s \Rightarrow$ i.e. code distance $= \Omega(n)$. (Not possible in finite Euclidean dim.)

We also want high rate – more variables than checks. n = number of variables, and m = number of checks. Rate

$$R \geq \frac{n-m}{n} = 1 - \frac{m}{n}$$

at least $n-m$ unconstrained.

Suppose D_v = ave. vertex degree and D_c = ave. check degree. Then, because we can count edges two ways:

$$D_v n = D_c m \Rightarrow \frac{m}{n} = \frac{D_v}{D_c}$$

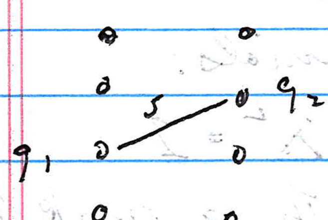
$$\Rightarrow R \geq 1 - \frac{D_v}{D_c}$$

Tension: To get high expansion we want large enough D_v, which reduces rate.

Our bipartite graph is <u>unbalanced</u>, with a different number of checks and variables to get high rate, and we want expansion in both directions: sets of variables have many check neighbors and sets of checks have many variable neighbors (so A and A^T have large distance). Fortunately only A and not A^T needs to have high rate.

There are randomized constructions of unbalanced bipartite graphs that yield double-sided expansion with high probability, but for practical error correction we want explicit constructions with nice structure (for decoding and logical operations).

One way to get a <u>balanced</u> double-sided expander is with Cayley graphs. Nodes are elements of a nonabelian group, and we pick a set of generators. An edge connects g_1 ~~at~~ on left with g_2 on right if $g_2 = s g_1$ and s is in the generating set.



Degree is size of generator set.

How to get double-sided expansion in unbalanced bipartite graph? Now choose two subgroups K and H, and nodes are cosets. So e.g. number of variables is

$$|G|/|H| \text{ and number of checks is } |G|/|K|$$

where $|K| > |H|$. Cosets connected by edge
Connect gK to H cosets such that gKs contains a representative of that coset, ~~[illegible]~~ where s is a generator.

$$\text{Rate} \geq \frac{|G|/|H| - |G|/|K|}{|G|/|H|} = 1 - \frac{|H|}{|K|}$$

Need to be careful how we choose G, H, K, and generating set. A coset graph

Another way to build on H.G.P code construction to build codes of practical interest: Lifted product codes. Replace entries in A and B by polynomials - i.e. circulant matrices
These could be multivariate polynomials, but we can construct useful codes where entries are univariate monomials:
X^p where $p \in \{0, 1, \ldots, L-1\}$ and $X^L = 1$
These are $L \times L$ matrices with a single 1 in each row and each column.