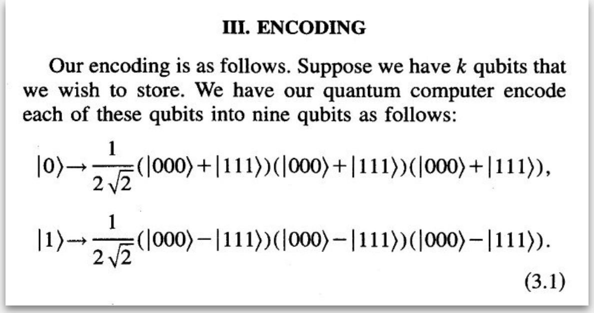
Twisted Product Constructions for LDPC Quantum Codes

Nikolas Breuckmann

Quantum Wave in Computing Reunion 2021

UCL

Quantum Codes



III. ENCODING

Our encoding is as follows. Suppose we have k qubits that we wish to store. We have our quantum computer encode each of these qubits into nine qubits as follows:

$$|0\rangle \rightarrow \frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle),$$

$$|1\rangle \rightarrow \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle). \tag{3.1}$$

Shor (1995)

- quantum errors can be corrected
- hide a state in non-local degrees of freedom
- perform (commuting) *check measurements* to infer error

Parameters of Quantum Codes:

- number of physical qubits *n*
- number of encoded qubits *k*
- smallest weight of an undetectable error *d* (distance)

[[n,k,d]]

Classical Codes & CSS Quantum Codes

Classical (linear, binary) codes

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$$C = \ker H \leq \mathbb{F}_2^{\,n}$$

CSS quantum codes

Defined by two check matrices

$$H_X \qquad H_Z$$

with condition:

$$H_X H_Z^{tr} = 0$$

Bravyi-Leemhuis-Terhal (2010):
Any [[n,k,d]] stabilizer code can be mapped onto a [[4n,2k,2d]] CSS code.

LDPC Quantum Codes

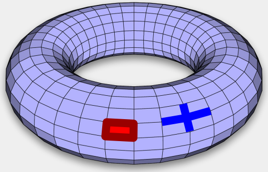
LDPC (Quantum) Codes

Two conditions:

1. Each check operates on O(1) (qu)bits
2. Each (qu)bit participates in O(1) checks

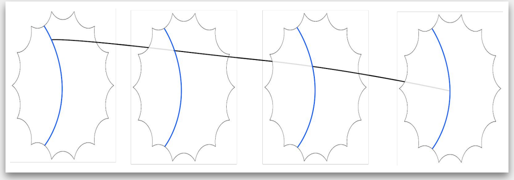
- Classical: H needs to be sparse / CSS: H_X and H_Z must be sparse
- Classical LDPC codes: *good codes* ($k = \Theta(n)$ and $d = \Theta(n)$)
- **Big open question:** *Can LDPC quantum codes be good?*
  - hard: $H_X H_Z^{tr} = 0$
  - take good code H_X then H_Z can not be sparse!
- Until recently: not even distance scaling beyond polylog(n) $\sqrt{n}$

History of LDPC Quantum codes with large distances



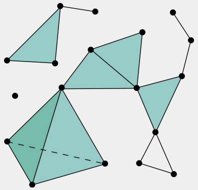
Kitaev: Toric code

$k=2, \quad d=\Theta(\sqrt{n})$



Freedman-Meyer-Luo

$k=1, \quad d\geq\Omega(\sqrt[4]{\log n}\,\sqrt{n})$

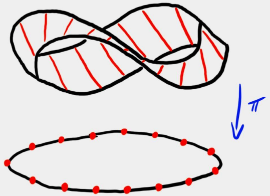


Evra-Kaufman-Zémor

$d=\Omega(\sqrt{\log n}\,\sqrt{n})$

Kaufman-Tessler

$d=\Omega((\log n)^s\sqrt{n})$



Fiber bundle codes

Hastings-Haah-O'Donnell

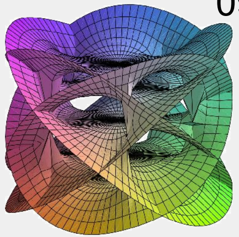
$d\geq\Omega(n^{3/5}/\mathrm{polylog}\,n), \quad k=\Theta(n^{3/5}/\mathrm{polylog}\,n)$

$$\mathcal{A}\otimes_R\mathcal{B}$$

Lifted product codes

Panteleev-Kalachev

$d\geq\Omega(n^{1-\alpha/2}/\log n), \quad k=\Theta(n^{\alpha}\log n)$

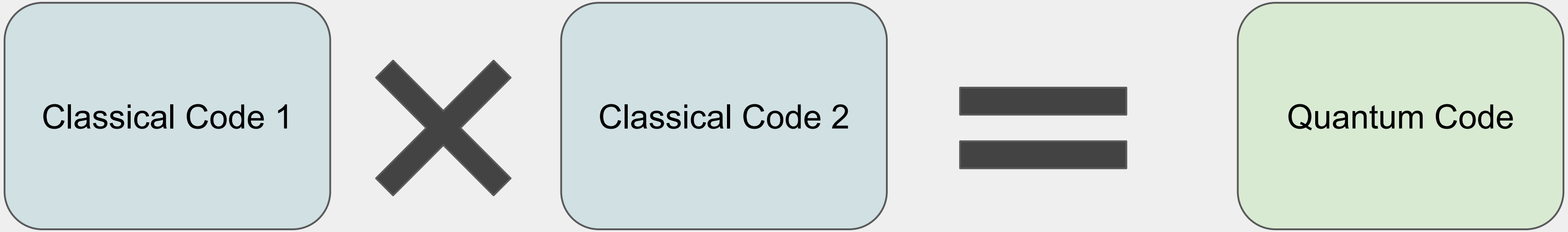


Balanced product codes

Breuckmann-Eberhardt

$d\geq\Omega(n^{3/5}), \quad k=\Theta(n^{4/5})$

Product Constructions

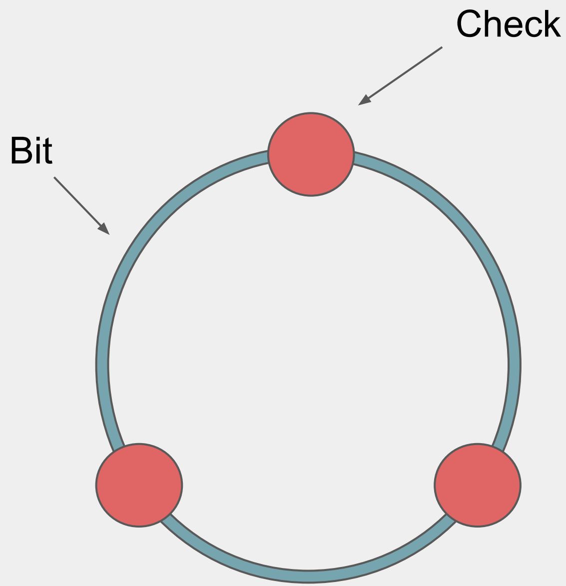


- Can take product of two classical codes
- Quantum code inherits properties of input codes

Tillich-Zémor (2013), Bravyi-Hastings (2014), Audoux-Couvreur (2015), Zeng-Pryadko (2019), Evra-Kaufman-Zémor (2020)

Product Constructions

- Algebraic approach is technical
- Want to talk about codes in terms of *topology*



Take repetition code on 3 bits as an example:

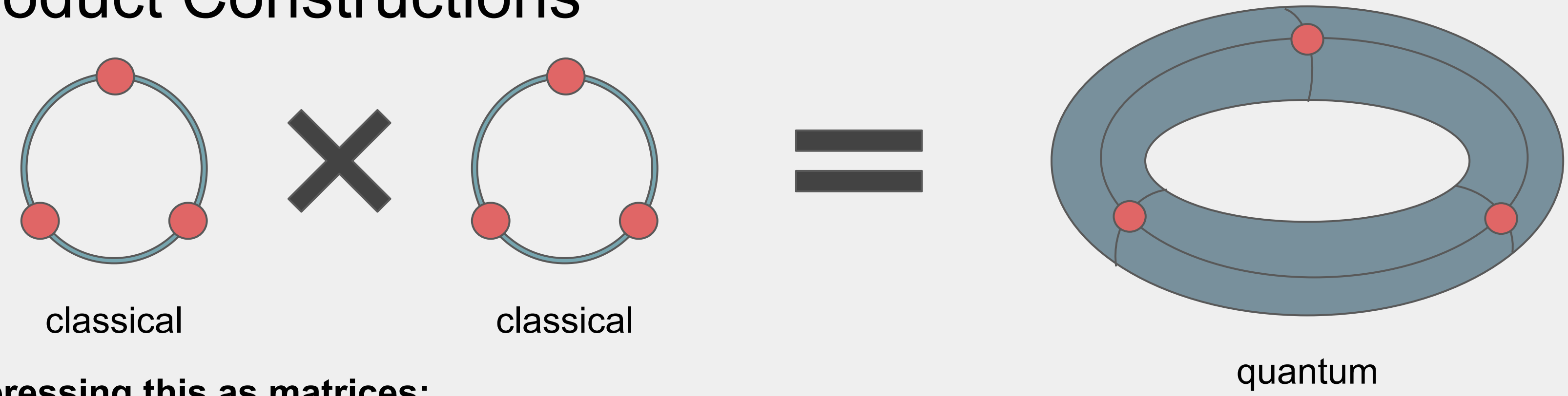
- 3 bits
- codewords are 000 and 111
- 3 parity checks

$$H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

Can be identified with a circle graph:

- edges are bits
- vertices are checks

Product Constructions



Expressing this as matrices:

$$H_X = \left(H_1 \otimes I \;\middle|\; I \otimes H_2^{tr} \right)$$

$$H_Z = \left(I \otimes H_2 \;\middle|\; H_1^{tr} \otimes I \right)$$

$$H_X \cdot H_Z^T = \left(H_1 \otimes I \;\middle|\; I \otimes H_2^T \right) \begin{pmatrix} I \otimes H_2^T \\ H_1 \otimes I \end{pmatrix}$$

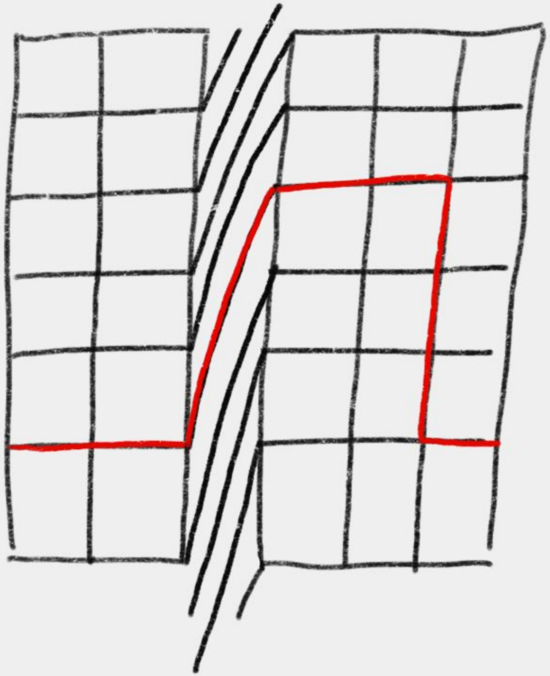
$$= H_1 \otimes H_2^T + H_1 \otimes H_2^T = 0$$

Observation:

- Kronecker product: size of quantum code is product of the input code lengths
- distance is the same as for input codes
- can achieve at most d = $\Theta(\sqrt{n})$

Fiber Bundle Codes

Hastings-Haah-O'Donnell (also Freedman-Meyer-Luo):
introduce *twists* into the product to increase distance

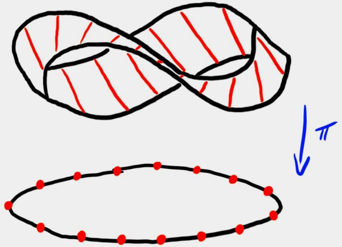


- toric code: shortest non-contractible loop = distance
- take LxL torus: distance is L
- can twist torus to *increase length* of one of the loops

Fiber Bundle Codes

Twists can be described in the language of *fiber bundles*:

- Base manifold
- Fiber manifold
- local data describing twists

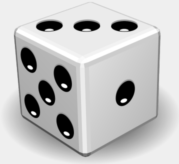


Hastings-Haah-O'Donnell:

Random code as base

+ circle as fiber (repetition codes)

+ random twists



$$d \geq \Omega\left(n^{3/5}/\mathrm{polylog}\, n\right),$$

$$k = \Theta\left(n^{3/5}/\mathrm{polylog}\, n\right)$$

with high probability

Fiber Bundle Codes

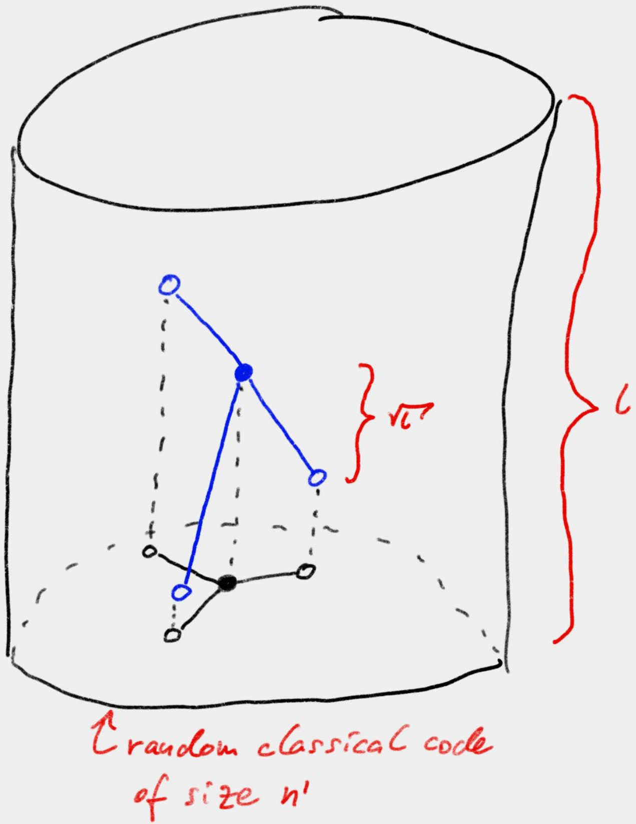
Step 1: Construction of fiber bundle

Input:

- base code has n' bits
- fiber is repetition code of size $\ell = \Theta(n')$
- twists are random and of length $\Theta(\sqrt{\ell})$

Result:

- code has $n = \Theta(\ell\, n') = \Theta(n'^2)$ qubits
- encodes $k = \Theta(n')$ qubits w.h.p.
- X-distance is $\Theta(\sqrt{\ell}\, n') = \Theta(n^{3/4})$ w.h.p.
- Z-distance is $\Theta(\ell) = \Theta(n^{1/2})$ w.h.p.
- still limited by the Z-distance!



Fiber Bundle Codes

Step 2: Distance balancing (Hastings '17, Evra et al. '20)

Imbalanced Quantum Code $\times$ Classical Code $=$ Balanced Quantum Code

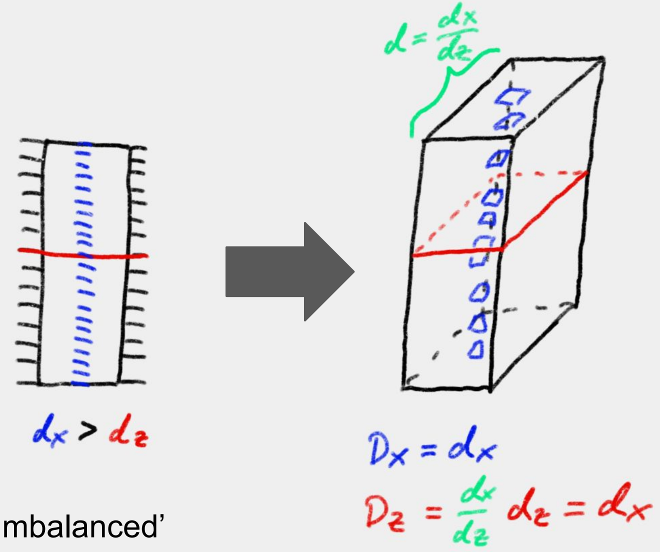
Input:

- quantum code A with parameters $[[n_A,k_A,d_X,d_Z]]$
- classical code B with parameters $[n_B,k_B,d]$

Output:

Quantum code with

- $n' = \Theta(n_A\, n_B)$
- $k' = \Theta(k_A\, k_B)$
- distances
  - $d'_X = d_X$ and
  - $d'_Z = d\, d_Z$



'imbalanced' surface code

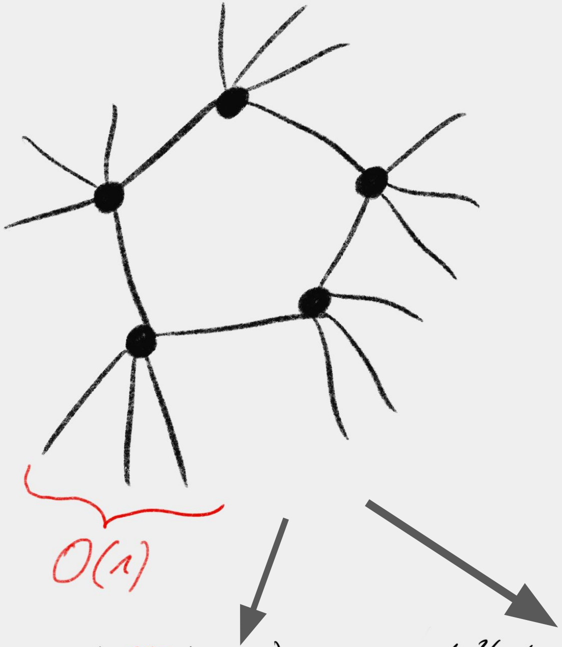
'balanced' 3D code

- when $d_X d_z$ scales faster than $\sqrt{n}$ we break polylog(n)$\sqrt{n}$-distance barrier
- their example: $d_X \geq \Omega(n^{3/4})$ and $d_Z \geq \Omega(n^{1/2})$
- obtain code with $d \geq \Omega(n^{3/5})$

Fiber Bundle Codes

Step 3: Reducing check weights

Checks of random code have polylog(n) weight



$d \geq \Omega(n^{3/5}/\mathrm{polylog}\, n)$, $k = \Theta(n^{3/5}/\mathrm{polylog}\, n)$

Lifted Product Codes

Main idea:

Lift the tensor product of classical codes from $\mathbb{F}_2$

to an ℓ-dimensional, commutative $\mathbb{F}_2$-algebra R

Lifted product:

Take 'check matrices' H_1 & H_2 with entries in R

$$H_X = \left(H_1 \otimes_R I^R \,\middle|\, I^R \otimes_R H_2^* \right) \qquad H_Z = \left(I^R \otimes_R H_2 \,\middle|\, H_1^* \otimes_R I^R \right)$$

The tensor product over R gives quantum codes which are smaller by a factor of ℓ

Lifted Product Codes

Special case: consider the $\mathbb{F}_2$-algebra $R = \mathbb{F}_2[x]/\langle x^\ell - 1\rangle$

and let H_2 be the repetition code of length ℓ (checks generated by 1+x)

Connection to Hastings-Haah-O'Donnell:
Think of x as shift-operator along fiber of length ℓ

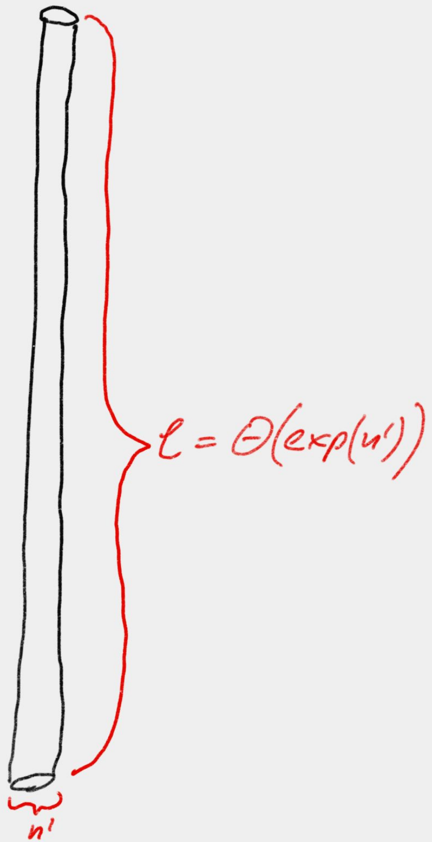
- H_2 is constructed from 'random cyclic lifts' of a special graph called *expander graph* (more on that later)
- Similar to fiber bundle construction: random lifts ≈ random twists
- Slightly more structured than HHO: *No weight reduction needed*

Agarwal et al., SIAM Journal on Discrete Mathematics, vol. 33, no. 3, 2019

Lifted Product Codes

Almost linear distance:

- classical code of length n’
- choose $\ell = \Theta(\exp(n'))$
- obtain quantum code of size $n = \Theta(\ell\, n')$
- encodes $k = \Theta(n') = \Theta(\log(n))$ qubits
- using a very elegant ‘averaging trick’ they show that <u>$d = \Theta(\ell) = \Theta(n/\log(n))$</u>



Lifted Product Codes

Quantum Code × Classical Code × Classical Code = Quantum Code

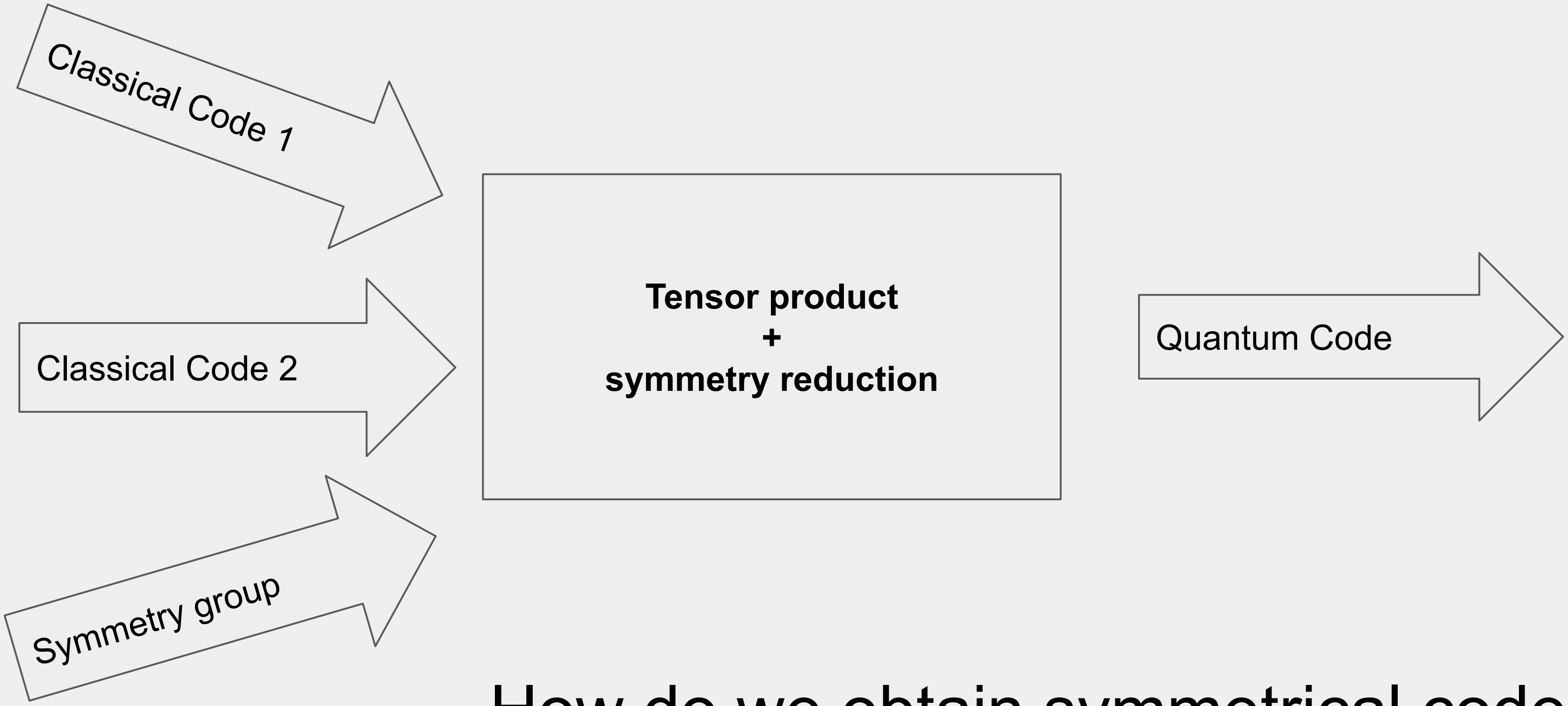
Dimension Balancing

Can take further (regular) tensor products to trade distance for encoded qubits

- [[N,K,D]] quantum code Q
- [n',k',d'] classical code C
- $Q \otimes C \otimes C^*$ gives code with $[[\Theta(n'^2 N), \Theta(k'^2 K), \Omega(d'D)]]$

$$d \geq \Omega\left(n^{1-\alpha/2}/\log n\right), \quad k = \Theta\left(n^{\alpha}\log n\right)$$

Balanced Product Quantum Codes



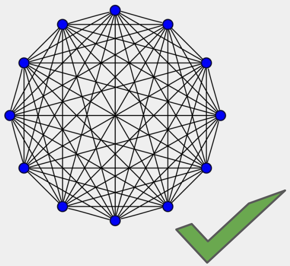
How do we obtain symmetrical codes?

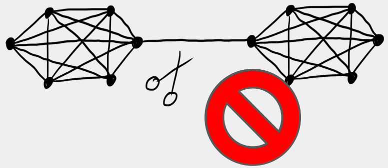
Expander Graphs

Graphs which have strong connectivity property

Measured by *Cheeger constant* h

$$h(X) = \min_{\substack{S \subset V \\ 0 < |S| < \frac{|V|}{2}}} \frac{|\partial S|}{|S|}$$

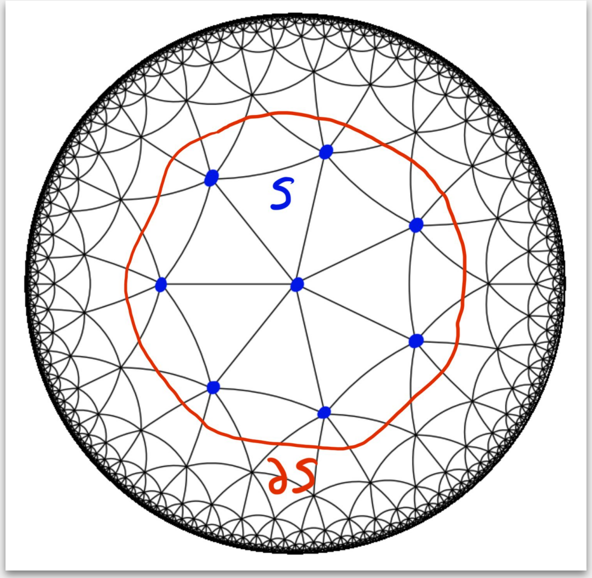




Interested in expander graphs of constant degree

Expander Graphs

- HHO & PK use random graphs which are good expanders
- We use an explicit (non-random) construction



Hyperbolic graphs are expanders

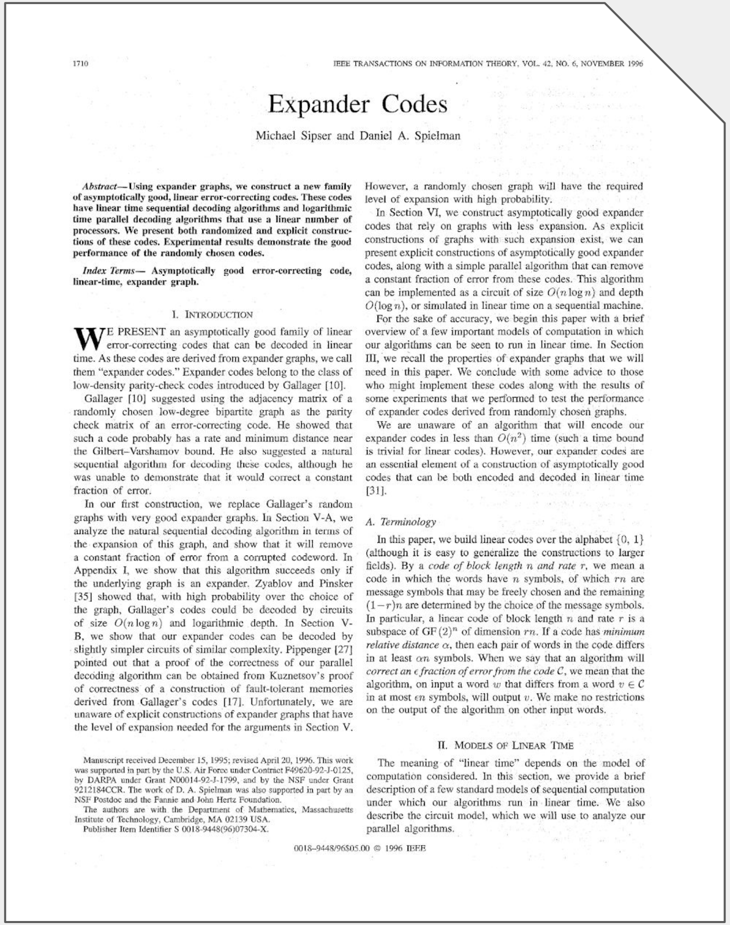
Lubotzky-Phillips-Sarnak (1988)

- *Optimal* expanders
- Cayley graphs of PGL(2,$\mathbb{F}_q$) or PSL(2,$\mathbb{F}_q$)
- Can be thought of as hyperbolic tessellations

Expander Codes

Sipser-Spielman (1996)

- Combine:
  - Family of expander graphs
  - Local block codes
- Obtain family of good classical codes



Expander Codes

Michael Sipser and Daniel A. Spielman

***Abstract*—Using expander graphs, we construct a new family of asymptotically good, linear error-correcting codes. These codes have linear time sequential decoding algorithms and logarithmic time parallel decoding algorithms that use a linear number of processors. We present both randomized and explicit constructions of these codes. Experimental results demonstrate the good performance of the randomly chosen codes.**

***Index Terms*— Asymptotically good error-correcting code, linear-time, expander graph.**

I. Introduction

WE PRESENT an asymptotically good family of linear error-correcting codes that can be decoded in linear time. As these codes are derived from expander graphs, we call them "expander codes." Expander codes belong to the class of low-density parity-check codes introduced by Gallager [10].

Gallager [10] suggested using the adjacency matrix of a randomly chosen low-degree bipartite graph as the parity check matrix of an error-correcting code. He showed that such a code probably has a rate and minimum distance near the Gilbert–Varshamov bound. He also suggested a natural sequential algorithm for decoding these codes, although he was unable to demonstrate that it would correct a constant fraction of error.

In our first construction, we replace Gallager's random graphs with very good expander graphs. In Section V-A, we analyze the natural sequential decoding algorithm in terms of the expansion of this graph, and show that it will remove a constant fraction of error from a corrupted codeword. In Appendix I, we show that this algorithm succeeds only if the underlying graph is an expander. Zyablov and Pinsker [35] showed that, with high probability over the choice of the graph, Gallager's codes could be decoded by circuits of size $O(n \log n)$ and logarithmic depth. In Section V-B, we show that our expander codes can be decoded by slightly simpler circuits of similar complexity. Pippenger [27] pointed out that a proof of the correctness of our parallel decoding algorithm can be obtained from Kuznetsov's proof of correctness of a construction of fault-tolerant memories derived from Gallager's codes [17]. Unfortunately, we are unaware of explicit constructions of expander graphs that have the level of expansion needed for the arguments in Section V. However, a randomly chosen graph will have the required level of expansion with high probability.

In Section VI, we construct asymptotically good expander codes that rely on graphs with less expansion. As explicit constructions of graphs with such expansion exist, we can present explicit constructions of asymptotically good expander codes, along with a simple parallel algorithm that can remove a constant fraction of error from these codes. This algorithm can be implemented as a circuit of size $O(n \log n)$ and depth $O(\log n)$, or simulated in linear time on a sequential machine.

For the sake of accuracy, we begin this paper with a brief overview of a few important models of computation in which our algorithms can be seen to run in linear time. In Section III, we recall the properties of expander graphs that we will need in this paper. We conclude with some advice to those who might implement these codes along with the results of some experiments that we performed to test the performance of expander codes derived from randomly chosen graphs.

We are unaware of an algorithm that will encode our expander codes in less than $O(n^2)$ time (such a time bound is trivial for linear codes). However, our expander codes are an essential element of a construction of asymptotically good codes that can be both encoded and decoded in linear time [31].

A. Terminology

In this paper, we build linear codes over the alphabet $\{0, 1\}$ (although it is easy to generalize the constructions to larger fields). By a *code of block length n and rate r*, we mean a code in which the words have n symbols, of which rn are message symbols that may be freely chosen and the remaining $(1-r)n$ are determined by the choice of the message symbols. In particular, a linear code of block length n and rate r is a subspace of $GF(2)^n$ of dimension rn. If a code has *minimum relative distance α*, then each pair of words in the code differs in at least αn symbols. When we say that an algorithm will *correct an ϵ fraction of error from the code C*, we mean that the algorithm, on input a word w that differs from a word $v \in C$ in at most ϵn symbols, will output v. We make no restrictions on the output of the algorithm on other input words.

II. Models of Linear Time

The meaning of "linear time" depends on the model of computation considered. In this section, we provide a brief description of a few standard models of sequential computation under which our algorithms run in linear time. We also describe the circuit model, which we will use to analyze our parallel algorithms.

Manuscript received December 15, 1995; revised April 20, 1996. This work was supported in part by the U.S. Air Force under Contract F49620-92-J-0125, by DARPA under Grant N00014-92-J-1799, and by the NSF under Grant 9212184CCR. The work of D. A. Spielman was also supported in part by an NSF Postdoc and the Fannie and John Hertz Foundation.

The authors are with the Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

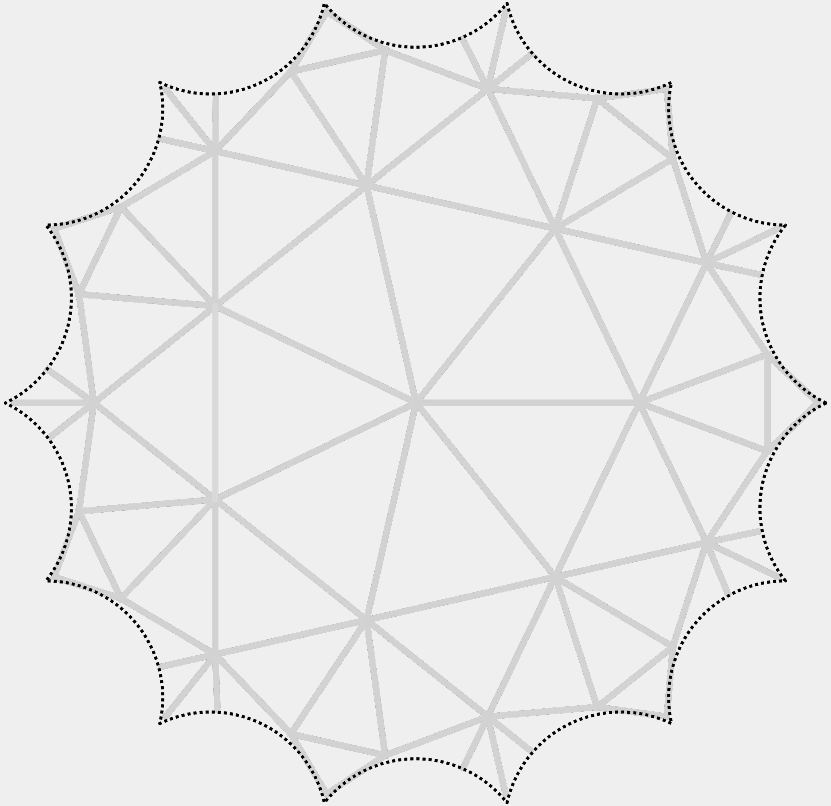
Publisher Item Identifier S 0018-9448(96)07304-X.

animation by Greg Egan

Expander Codes

Example

- Hyperbolic graph
- Edges are bits
  & vertices are parity checks
- **Not** checking for parity only
  (as in toric code)
- Check for code words of a
  local code L

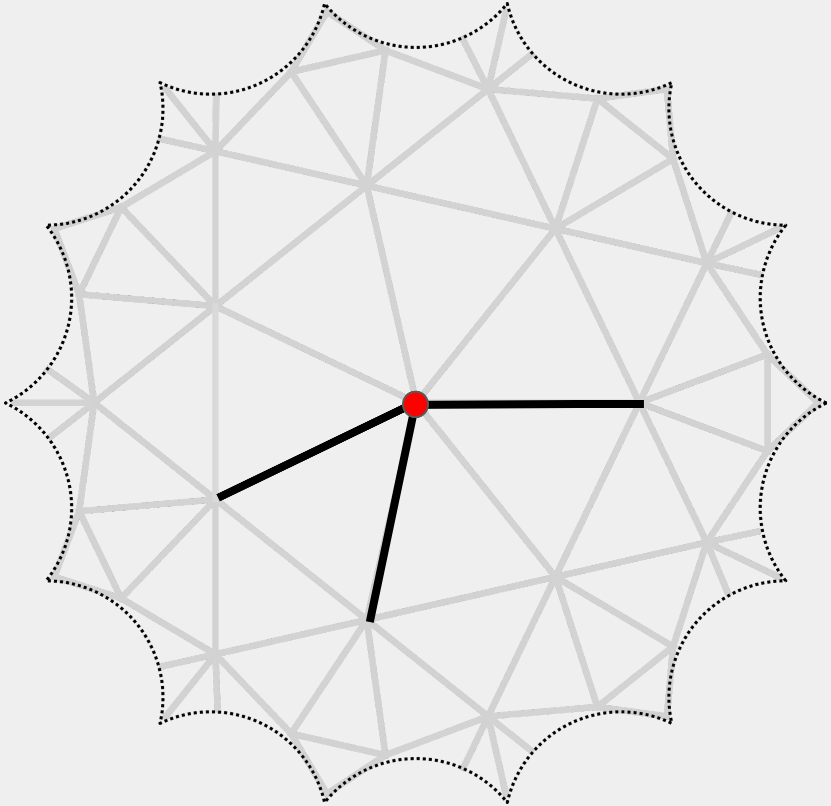


All code words of the [7,4,3] Hamming code (up to cyclic shifts)

```
[0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 1, 1, 1]
[0, 0, 0, 1, 0, 1, 1]
[0, 0, 0, 1, 1, 1, 1]
[0, 0, 1, 0, 0, 1, 1]
[0, 0, 1, 0, 1, 0, 1]
[0, 0, 1, 1, 0, 1, 1]
[0, 0, 1, 1, 1, 0, 1]
[0, 1, 0, 1, 0, 1, 1]
[1, 1, 1, 1, 1, 1, 1]
```

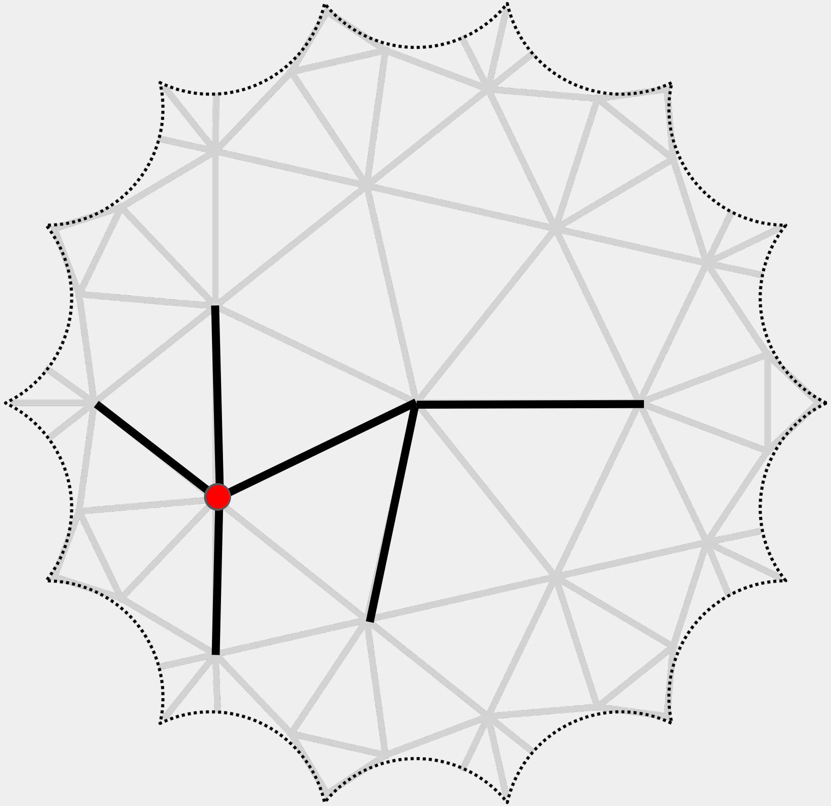
Expander Codes

Put code word of Hamming
code around a vertex



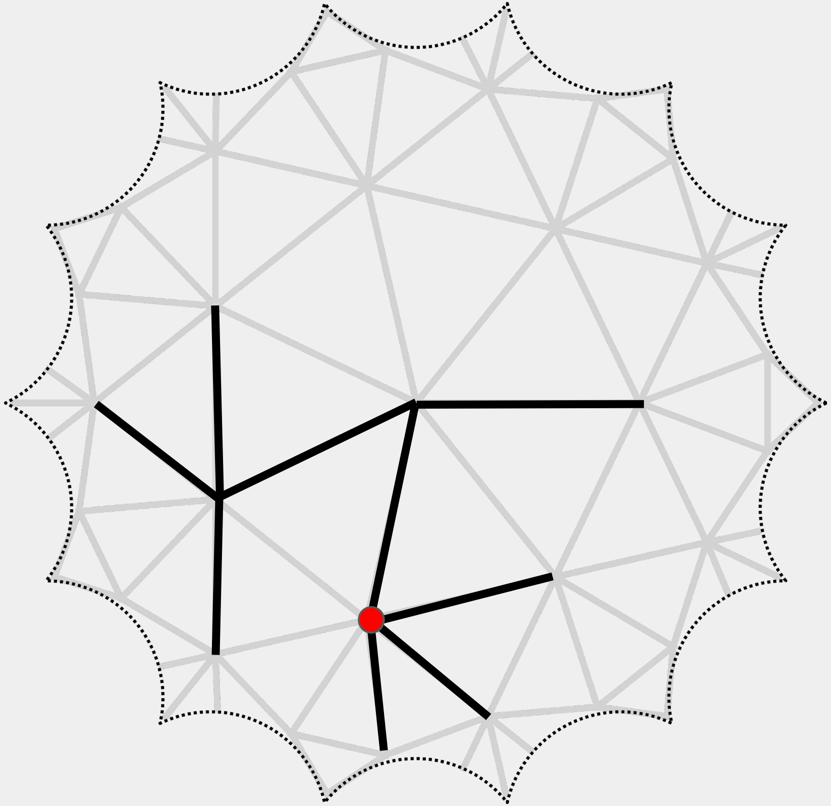
```
[0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 1, 1, 1]
[0, 0, 0, 1, 0, 1, 1]
[0, 0, 0, 1, 1, 1, 1]
[0, 0, 1, 0, 0, 1, 1]
[0, 0, 1, 0, 1, 0, 1]
[0, 0, 1, 1, 0, 1, 1]
[0, 0, 1, 1, 1, 0, 1]
[0, 1, 0, 1, 0, 1, 1]
[1, 1, 1, 1, 1, 1, 1]
```

Expander Codes



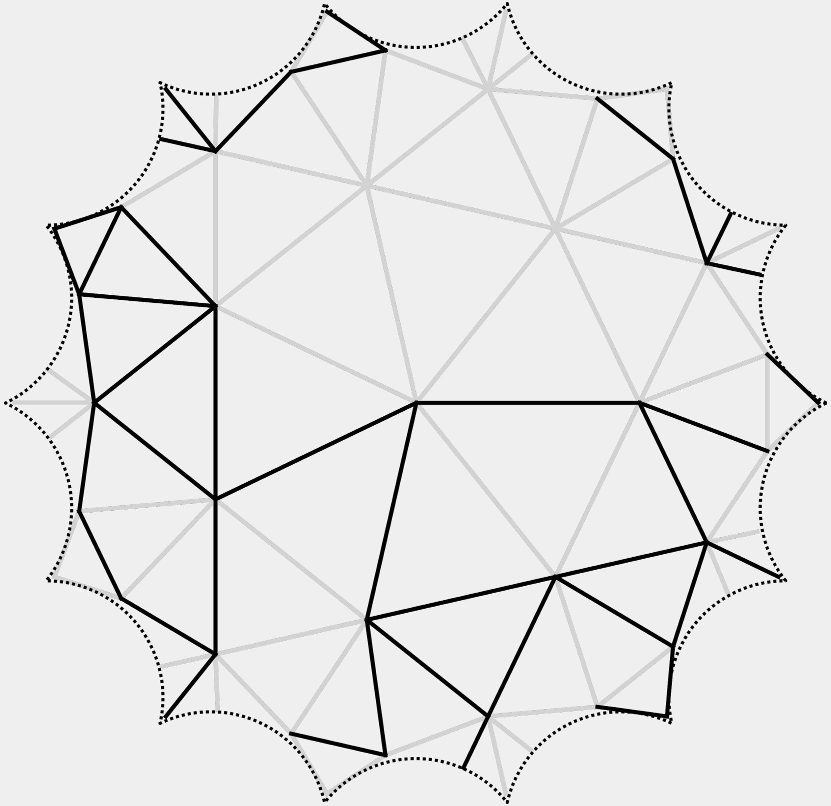
```
[0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 1, 1, 1]
[0, 0, 0, 1, 0, 1, 1]
[0, 0, 0, 1, 1, 1, 1]
[0, 0, 1, 0, 0, 1, 1]
[0, 0, 1, 0, 1, 0, 1]
[0, 0, 1, 1, 0, 1, 1]
[0, 0, 1, 1, 1, 0, 1]
[0, 1, 0, 1, 0, 1, 1]
[1, 1, 1, 1, 1, 1, 1]
```

Expander Codes



```
[0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 1, 1, 1]
[0, 0, 0, 1, 0, 1, 1]
[0, 0, 0, 1, 1, 1, 1]
[0, 0, 1, 0, 0, 1, 1]
[0, 0, 1, 0, 1, 0, 1]
[0, 0, 1, 1, 0, 1, 1]
[0, 0, 1, 1, 1, 0, 1]
[0, 1, 0, 1, 0, 1, 1]
[1, 1, 1, 1, 1, 1, 1]
```

Expander Codes

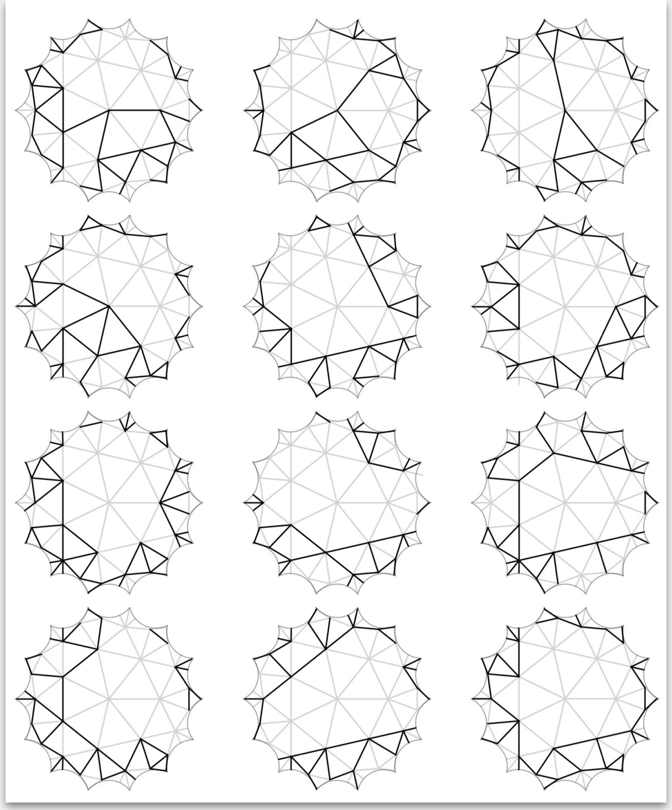


0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	1	0	1	1
0	0	0	1	1	1	1
0	0	1	0	0	1	1
0	0	1	0	1	0	1
0	0	1	1	0	1	1
0	0	1	1	1	0	1
0	1	0	1	0	1	1
1	1	1	1	1	1	1

Expansion of graph X
+
distance of local code L

→enforce code words of weight Θ(n)!

Expander Codes



Counting degrees of freedom (bits) and constraints (checks) we get bound:

$$k \geq \text{const.} \times n$$

Obtain classical codes with parameters

$[n, k=\Theta(n), d=\Theta(n)]$

"Good codes"

Balanced Products

Topological notion:
Consider two topological spaces X & Y on which group H acts from left & right, respectively.

For any pair in their cartesian product $(x, y) \in X \times Y$

We define the anti-diagonal action $h \cdot (x, y) = (x h^{-1}, h y)$

The *balanced product* X x_H Y is then given by the quotient space:

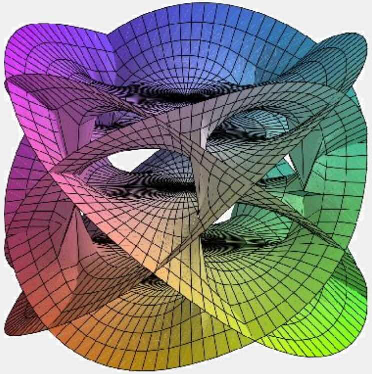
$$X \times_H Y = X \times Y / H$$

Balanced Product Quantum Codes

Use *balanced product* to take product with repetition code

Input:

- expander code with cyclic symmetry
- repetition code
- cyclic symmetry group H



Geometric intuition:

- expander code comes with associated Riemann surface
- wrap surface around itself (giving rise to twists)
- glue-in circles (repetition codes)

For the experts:

- Deal with a double complex
- Checks are defined through boundary operators respecting balanced product
- Current Example:
    - *One type of check is simply the check matrix of the expander code!*

$$\begin{tikzcd}
C_1(X) \otimes_H C_1(\mathbb{Z}_\ell) \arrow[r, "\partial\otimes id"] \arrow[d, swap, "id\otimes\partial"] & (C_0(X)\otimes L_0) \otimes_H C_1(\mathbb{Z}_\ell) \arrow[d, "id\otimes\partial"] \\
C_1(X) \otimes_H C_0(\mathbb{Z}_\ell) \arrow[r, swap, "\boxed{\partial\otimes id}"] & (C_0(X)\otimes L_0) \otimes_H C_0(\mathbb{Z}_\ell)
\end{tikzcd}$$

Balanced Product Codes

Codes constructed using balanced product:

1. Take s-regular expander graph X to be Cayley graph of $PGL(2,\mathbb{F}_q)$:
   a. $|PGL(2,\mathbb{F}_q)| = q(q^2-1)$
   b. contains cyclic subgroup H of order $|H| = q$
2. Local code L guaranteed to exist

Better bounds could be attainable if we knew other Cayley expanders

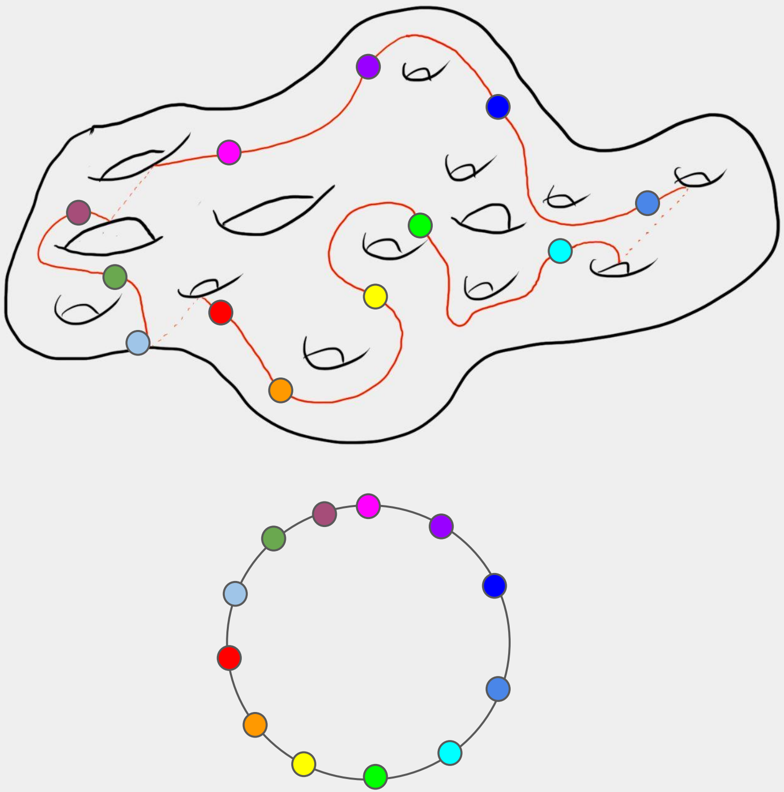
Properties of the code:

- Number of qubits n = 3 x number of edges in X
- Number of logicals $k = \Theta(n^{2/3})$
- Z-distance $d_Z = \Theta(n)$
- X-distance $d_X \geq \Omega(n^{1/3})$

using Panteleev-Kalachev bounds

Taking product with suitable classical code gives $k = \Theta(n^{4/5})$ and $d \geq \Omega(n^{3/5})$

Concrete Example



Genus-14 surface with order-13 cyclic symmetry

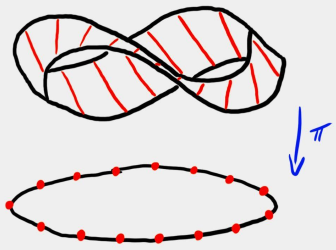
Local Code: Hamming [7,4,3]

Quantum code:

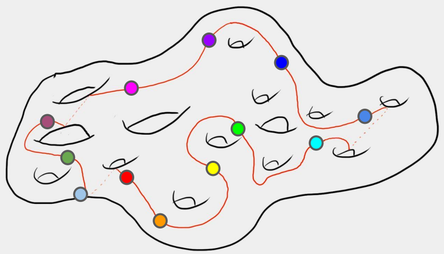
- 1014 physical qubits
- 6 logical qubits
- Monte Carlo:
  - X-distance ≤ 13
  - Z-distance ≤ 18
- check-weights 6 (X-checks) and 4-8 (Z-checks)

Balanced Products are symmetric

- **Fiber bundle & Lifted product codes:**
  - defined in terms of *base* and *fiber*
  - There are restrictions on what the fiber can be



- **Balanced product codes:**
  - different point of view
  - product is more general
  - associated group algebra can be non-commutative



Conjecture on Good LDPC Quantum Codes

Conjecture:

Consider two suitable classical codes C_1 & C_2 (good & LDPC) of length n with common symmetry group G of size $\Theta(n)$.

The balanced product $C_1 \otimes_G C_2$ is a good LDPC quantum code [$k = \Theta(n)$ and $d = \Theta(n)$]

Already checked that $k = \Theta(n)$.

Things I did not talk about:

- All these products can be cast in the language of homology
    - streamlines a lot of concepts
    - makes proofs simpler
- Find more background in perspective article arXiv:2103.06309

Thank you!