Flowering graphs

Proximity test to codes on graphs by flowering

Hugo Delavenne, Tanguy Medevielle, Élina Roussel

Future Surycat team
LIX, École Polytechnique, Institut Polytechnique de Paris
Inria

Friday 4^{th} April 2025
@ C2 Days 2025, Pornichet

SNARK means **S**uccinct **N**on-interactive **AR**gument of **K**nowledge.

It turns $A : x \mapsto y$ (that runs in $\tau(|x|)$) into

$$A' : \underbrace{x}_{\text{input}}, \underbrace{\alpha}_{\text{randomness}} \mapsto \underbrace{y}_{\text{output}}, \underbrace{\pi}_{\text{SNARK proof that } ``A(x)=y"}$$

satisfying:

- $|\pi| \ll \tau$
- A' runs in $\tilde{O}(\tau)$
- there is a verifier $\mathcal{V}$ in $\mathrm{poly}(|\pi|)$ such that
  - **Completeness:** if $A(x) = y$ then $\mathbb{P}_{\alpha}(\mathcal{V}(\pi) \text{ accepts}) = 1$
  - **Soundness:** if $A(x) \neq y$ then $\mathbb{P}_{\alpha}(\mathcal{V}(\pi) \text{ accepts}) \leqslant s$.

Contents

1 Interactive proximity tests

2 Flowering protocol

3 Flowering graphs

- Interactive Oracle Proof of Proximity
  - Interactive Proofs & arithmetization
  - Oracle access
  - Interactive Oracle Proofs of Proximity
- Fast Reed-Solomon IOPP
  - Folding Reed-Solomon codes
  - FRI protocol
  - Complexities
  - Completeness and soundness

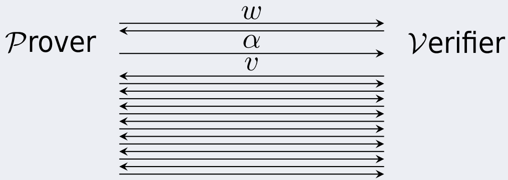
Definition **Non-interactive proof**

$$\mathcal{P}\text{rover} \xrightarrow{\quad w \quad} \mathcal{V}\text{erifier}$$

Definition **Sigma protocol**

$$\mathcal{P}\text{rover} \quad \begin{array}{c} \xrightarrow{\quad\quad w \quad\quad} \\ \xleftarrow[\quad\quad \alpha \quad\quad]{} \\ \xrightarrow[\quad\quad v \quad\quad]{} \end{array} \quad \mathcal{V}\text{erifier}$$

Definition **Interactive Proof**



Arithmetization reduces checking a computation to testing proximity to a code C:

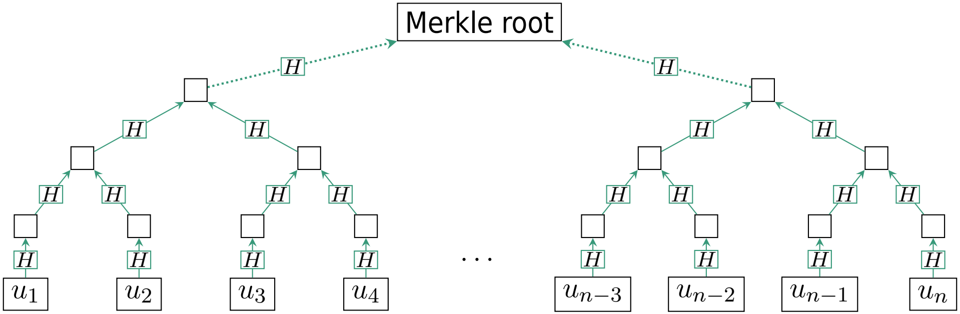
$$\begin{aligned} A(x) = y \quad &\Longrightarrow \quad \text{Arithmetization}(A, x, y) \in C \\ A(x) \neq y \quad &\Longrightarrow \quad \Delta(\text{Arithmetization}(A, x, y), C) > \delta \end{aligned}$$

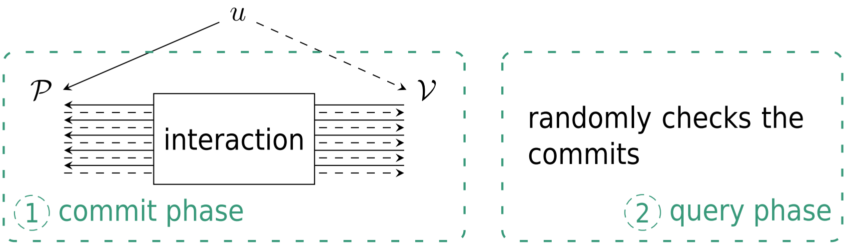
With arithmetization + proximity test + Fiat-Shamir we get a **NARK**!

Definition **Oracle access**

$\mathcal{P}$ can provide to $\mathcal{V}$ an **oracle access** to $u \in \mathbb{F}^n$ by giving **black-box** access to u.

In practice, $\mathcal{P}$ provides the root of a **Merkle tree**.





- **Completeness:** if $u \in C$ then $\mathbb{P}(\mathcal{V}^{u,\leftrightarrow \mathcal{P}}$ accepts$) = 1$
- **Soundness:** if $\Delta(u, C) > \delta$ then for any $\mathcal{P}$, $\mathbb{P}(\mathcal{V}^{u,\leftrightarrow \mathcal{P}}$ accepts$) \leqslant s$

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive Oracle Proofs.
In *Theory of Cryptography: 14th International Conference, TCC 2016-B, Beijing, China, October 31-November 3, 2016, Proceedings, Part II 14*, pages 31–60. Springer, 2016

We test proximity to **RS$[\mathcal{L}_N, K]$** $:= \{f : \mathcal{L}_N \to \mathbb{F} \mid f \in \mathbb{F}[X]_{<K}\}$, with $|\mathcal{L}_N| = N$.

Idea. Test even and odd parts $f(X) =: f_{\text{even}}(X^2) + X f_{\text{odd}}(X^2)$.

Definition Fold

$$\text{Fold}[f, \alpha](Y) := f_{\text{even}}(Y) + \alpha f_{\text{odd}}(Y) = \frac{f(X) + f(-X)}{2} + \alpha \frac{f(X) - f(-X)}{2X},$$

with "$Y = X^2$", $\alpha \in \mathbb{F}$

- **Field restriction:** $\mathcal{L}_{N/2} := \{x^2 \mid x, -x \in \mathcal{L}_N\}$ so $\mathbb{F}$ must have 2^N roots of unity
- **Validity preservation:** $f \in \text{RS}[\mathcal{L}_N, K] \iff \underset{\alpha}{\mathbb{P}}(\text{Fold}[f, \alpha] \in \text{RS}[\mathcal{L}_{N/2}, K/2]) > \frac{1}{|\mathbb{F}|}$
- **Local check:** $\mathcal{V}$ computes $\text{Fold}[f, \alpha](x^2)$ with 2 queries to f

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. In ***45th international colloquium on automata, languages, and programming (ICALP 2018)***. **Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

FRI protocol

$$f_0 : \mathcal{L}_N \to \mathbb{F}$$

$\mathcal{P}$ $\mathcal{V}$

$$\alpha_0 \xleftarrow{\$} \mathbb{F}$$

$$f_1 : \mathcal{L}_{N/2} \to \mathbb{F}$$

$$f_1 = \mathsf{Fold}[f_0, \alpha_0]$$

$$\vdots$$

$$\alpha_{r-1} \xleftarrow{\$} \mathbb{F}$$

$$f_r : \mathcal{L}_{N/2^r} \to \mathbb{F}$$

$$f_r = \mathsf{Fold}[f_{r-1}, \alpha_{r-1}]$$

1 commit phase

$$x_0 \xleftarrow{\$} \mathcal{L}_N,\ x_1 := x_0^2$$

$$f_1(x_1) \stackrel{?}{=} \mathsf{Fold}[f_0, \alpha_0](x_1)$$

$$\vdots$$

$$x_r := x_{r-1}^2$$

$$f_r(x_r) \stackrel{?}{=} \mathsf{Fold}[f_{r-1}, \alpha_{r-1}](x_r)$$

$$f_r \stackrel{?}{\in} \mathsf{RS}[\mathcal{L}_{N/2^r}, K/2^r]$$

repeat m times

2 query phase

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

Proposition **FRI complexities [BBHR18]**

FRI protocol for RS$[\mathcal{L}_N, K]$ with m repetitions has following complexity:

- Prover complexity: $< 6N$ (without encoding)
- Verifier complexity: $< 2m \log K$
- Number of queries: $< 2m \log K$

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

Proposition **FRI completeness**

If $f_0 \in \mathsf{RS}[\mathcal{L}_N, K]$ then $\mathcal{V}$ accepts with probability 1.

Theorem **FRI soundness**

If $\Delta(f_0, \mathsf{RS}[\mathcal{L}_N, K]) > \delta$ then for any $\tilde{\mathcal{P}}$ and $\varepsilon > 0$, $\mathcal{V}$ accepts with probability

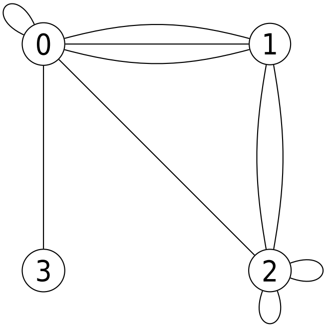
$$\leqslant \frac{K^2 \log K}{(2\varepsilon)^7 q} + \left(1 - \min\left(\delta, 1 - \sqrt{K/N} - \varepsilon\right)\right)^m.$$

[BCI+23] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity Gaps for Reed–Solomon Codes. *J. ACM*, 70(5), October 2023

- Graphs
  - Regular Indexed Multigraphs (RIM)
  - Words and codes on graphs
- Folding graphs
  - Tutorial: Cutting graphs
  - Graph isomorphism & folding
  - « *Vous voulez des exemples ?* »
- Flowering
  - Flowering protocol
  - Complexity comparison
  - Completeness and soundness

$\Gamma = (V, E)$ is a n-RIM:

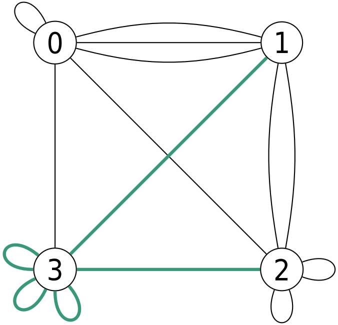
- **Multigraph:** multiple edges and loops



$\Gamma = (V, E)$ is a n-RIM:

▷ **Multigraph:** multiple edges and loops

▶ **Regular:** same number n of edges



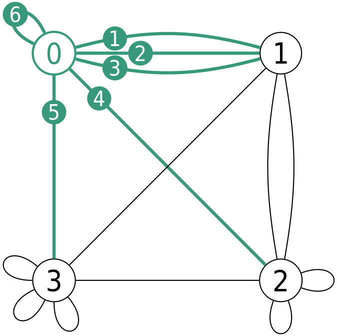
Regular Indexed Multigraphs (RIM)

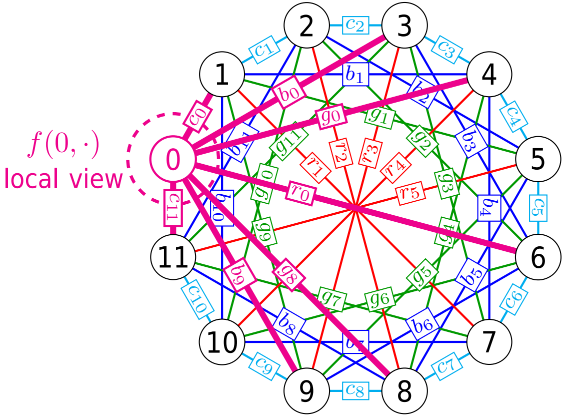
$\Gamma = (V, E)$ is a n-RIM:

▷ **Multigraph:** multiple edges and loops

▷ **Regular:** same number n of edges

▶ **Indexed:** edge is $(v, \ell) \in V \times [n]$
Write $\boldsymbol{E(v, \ell)} \in V$ the neighbor of v by ℓ





Word $f : V \times [n] \to \mathbb{F}$ on a graph Γ

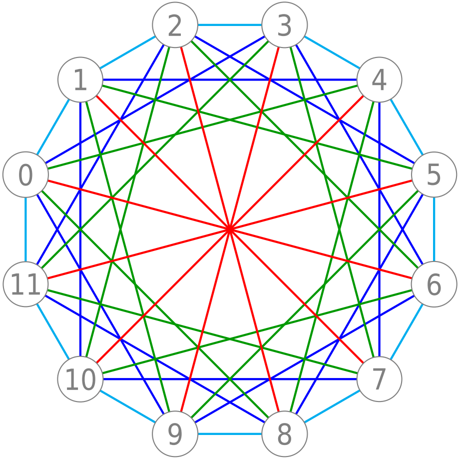
Definition **Code $\mathcal{C}[\Gamma, C_0]$**

Given Γ a n-RIM and $C_0 \subseteq \mathbb{F}^n$,

$$f \in \mathcal{C}[\Gamma, C_0] \iff \forall v,\ f(v,\cdot) \in C_0.$$

We'll only use $C_0 = \mathrm{RS}[n,k]$.

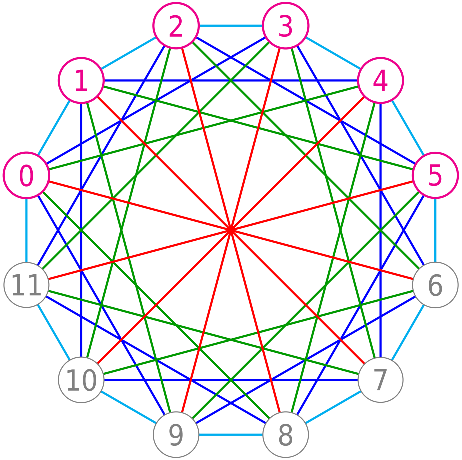
Cut-graph $\Gamma' = \mathrm{Cut}[\Gamma, V']$:



[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Cut-graph $\Gamma' = \mathrm{Cut}[\Gamma, V']$:

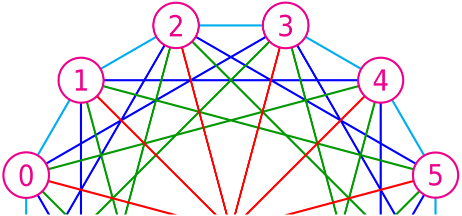
- Choose vertices $V' \subseteq V$



[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Cut-graph $\Gamma' = \mathrm{Cut}[\Gamma, V']$:

- Choose vertices $V' \subseteq V$
- Cut the rest

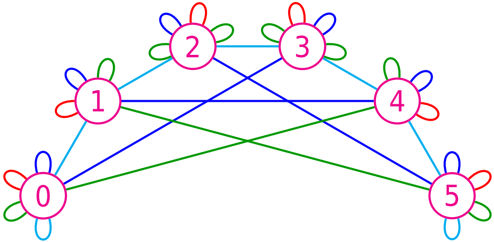


[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025.
Submitted

Cut-graph $\Gamma' = \text{Cut}[\Gamma, V']$:

- ▷ Choose vertices $V' \subseteq V$
- ▷ Cut the rest
- ▶ Enjoy your new graph

$$E_{V'}(v,\ell) = \begin{cases} E(v,\ell) & \text{if } E(v,\ell) \in V' \\ v & \text{otherwise} \end{cases}$$



[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Definition **Graph isomorphism**

A bijection $\varphi : V' \to V''$ is an **isomorphism** $\Gamma' \to \Gamma''$ if

$$\forall (v', \ell) \in V' \times [n], \quad \varphi(E'(v', \ell)) = E''(\varphi(v'), \ell).$$

Definition **Flowering cut**

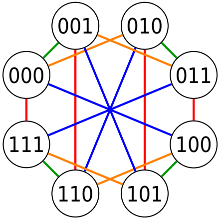
With $V'' = V \setminus V'$, if $\mathsf{Cut}[\Gamma, V'] \sim \mathsf{Cut}[\Gamma, V'']$, the cut is **flowering**.

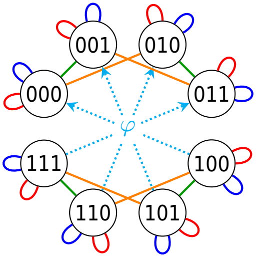
The cut-word $\mathsf{Cut}[f, V']$ is the restriction $f_{|V' \times [n]}$.

Definition **Folding**

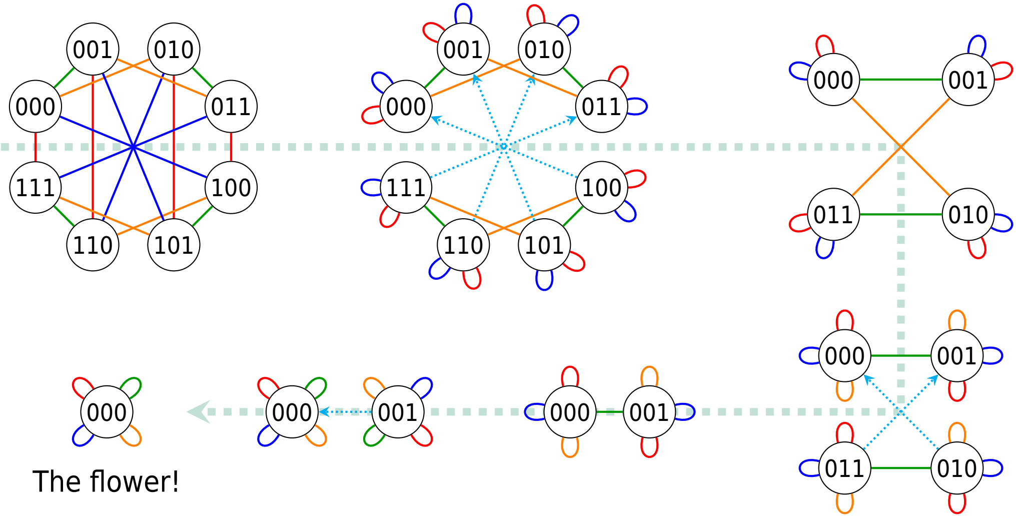
For $\alpha \in \mathbb{F}$, $(v', \ell) \in V' \times [n]$,

$$\mathsf{Fold}[f, \alpha](v', \ell) := \mathsf{Cut}[f, V'](v, \ell) + \alpha \mathsf{Cut}[f, V''](\varphi(v'), \ell).$$





[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025.
Submitted


001
010
000
011
111
100
110
101
001
010
000
011
111
100
110
101
000
001
011
010
000
001
011
010
000
001
000
001
000

The flower!

Flowering protocol

$$f_0 : V_0 \times [n] \to \mathbb{F}$$

$\mathcal{P}$ $\mathcal{V}$

$$\alpha_0 \xleftarrow{\$} \mathbb{F}$$

$$f_1 : V_1 \times [n] \to \mathbb{F}$$

$$f_1 = \mathsf{Fold}[f_0, \alpha_0]$$

$$\vdots$$

$$\alpha_{r-1} \xleftarrow{\$} \mathbb{F}$$

$$f_r : V_r \times [n] \to \mathbb{F}$$

$$f_r = \mathsf{Fold}[f_{r-1}, \alpha_{r-1}]$$

1 commit phase

$$v_0 \xleftarrow{\$} V_0,\ v_1 := \pi_1(v_0)$$

$$f_1(v_1, \cdot) \stackrel{?}{=} \mathsf{Fold}[f_0, \alpha_0](v_1, \cdot)$$

$$\vdots$$

$$v_r := \pi_r(v_{r-1})$$

$$f_r(v_r, \cdot) \stackrel{?}{=} \mathsf{Fold}[f_{r-1}, \alpha_{r-1}](v_r, \cdot)$$

$$f_r \stackrel{?}{\in} \mathsf{RS}[n, k]$$

repeat m times

2 query phase

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Proposition **Flowering complexities**

Flowering with m repetitions has complexities: (recall FRI)

- Prover complexity: $< 3N$ — $< 6N$
- Verifier complexity: $< 4mn \log N$ — $< 2m \log N$
- Number of queries: $< 2mn \log N$ — $< 2m \log K$

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. **In *45th international colloquium on automata, languages, and programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018**

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Proposition **Flowering completeness**

If $f \in \mathcal{C}[\Gamma, \mathsf{RS}[n,k]]$ then $\mathcal{V}$ accepts with probability 1.

Theorem **Flowering soundness**

If $\Delta(f, \mathcal{C}[\Gamma, \mathsf{RS}[n,k]]) > \delta$ then for any $\tilde{\mathcal{P}}$ and $\varepsilon > 0$, $\mathcal{V}$ accepts with probability

$$\leqslant \frac{\log N}{\varepsilon q} + (1 - \delta + \varepsilon \log N)^m .$$

Recall for FRI: $$\frac{K^2 \log K}{(2\varepsilon)^7 q} + \left(1 - \min\left(\delta, 1 - \sqrt{K/N} - \varepsilon\right)\right)^m$$

[BCI+23] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity Gaps for Reed–Solomon Codes. ***J. ACM*, 70(5), October 2023**

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

1 Interactive proximity tests

2 Flowering protocol

3 Flowering graphs

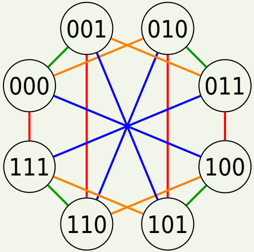
- Cayley graphs
- Our parameters
- Expander graphs
- Conclusion & future work

Definition **Cayley graph Cay(G, S)**

Fix $(G, \cdot)$. Let $S \subseteq G$ symmetric generating. Define $\Gamma = (G, E)$ where $E(g, s) = g{\cdot}s$.

Example

With $G = (\mathbb{F}_2^3, +)$ and $S = \{100, 010, 001, 111\}$,



We consider

- $G = (\mathbb{F}_2^r, +)$
- $S \subseteq G$ of size n

[Cay78] Arthur Cayley. Desiderata and Suggestions: No. 2. The Theory of Groups: Graphical Representation. ***American Journal of Mathematics*, 1(2):174–176, 1878**

Our parameters

We take

- $\Gamma = \mathsf{Cay}((\mathbb{F}_2^r, +), S)$
- S the columns of a parity check matrix of a $[n, n-r, d]_2$ binary code
- $C = \mathcal{C}[\Gamma, \mathsf{RS}[n,k]]$

Proposition **Parameters of the code**

- $N = n2^{r-1}$
- $\mathrm{rate}\, C \geqslant \frac{2k}{n} - 1$
- $\frac{1}{2}\delta \leqslant \Delta(C) \leqslant \delta$, with $\delta = \frac{1}{2^{r-d+1}}\left(1 - \frac{k-1}{n}\right) = \frac{n2^{d-2}}{N}\left(1 - \frac{k-1}{n}\right)$

If $d \ll r$ (i.e. far from MDS), δ is **terrible** ($O(1/N)$).

[DMR25] Hugo Delavenne, Tanguy Medevielle, and Élina Roussel. Interactive Oracle Proofs of Proximity to Codes on Graphs, 2025. **Submitted**

Lemma **Expansion bound [AC88]**

If Γ is λ-expander, with $\delta = 1 - \frac{k+1}{n}$,

$$\Delta(\mathcal{C}[\Gamma, \mathsf{RS}[n,k]]) \geqslant \delta(\delta - \lambda)$$

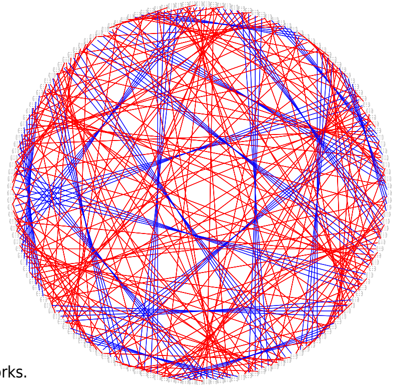
Thus **constant expansion** implies **constant minimal distance**.

Example **Expander graph family**

$\mathsf{Cay}(\mathrm{SL}_3(\mathbb{F}_p), S_p)$ has **constant expansion**.

We obtain $\mathsf{Cay}(\mathrm{SL}_3(\mathbb{F}_2), S_2)$ with

$$S_2 = \left\{ \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}^{\pm 1} \right\}$$



[AC88] Noga Alon and Fan Chung. Explicit construction of linear sized tolerant networks. ***Discrete Mathematics*, 72(1-3):15–19, 1988**

Competing parameters with FRI

- Better soundness
- Complexity could be improved

More possible cuts

- 2 cuts V', V'' $\quad\rightarrow\quad$ m cuts $V_1, V_2, ..., V_m$
- $V = V' \sqcup V''$ $\quad\rightarrow\quad$ $V = V_1 \cup V_2 \cup \cdots \cup V_m$