

# Notes on FRI and STIR

arnaucube

February 2023

## Abstract

Notes taken from Vincenzo Iovino [1] explanations about FRI [2], [3], [4].

These notes are for self-consumption, are not complete, don't include all the steps neither all the proofs.

An implementation of FRI can be found at <https://github.com/arnaucube/fri-commitment> [5].

Update(2024-03-22): notes on STIR [6] from explanations by Héctor Masip Ardevol [7].

## Contents

<b>1 Preliminaries</b>	<b>1</b>
1.1 General degree $d$ test . . . . .	1
<b>2 FRI protocol</b>	<b>2</b>
2.1 Intuition . . . . .	2
2.2 FRI-LDT . . . . .	2
2.3 Parameters . . . . .	5
<b>3 FRI as polynomial commitment scheme</b>	<b>5</b>
<b>4 STIR (main idea)</b>	<b>5</b>

## 1 Preliminaries

### 1.1 General degree $d$ test

Query at points  $\{x_i\}_0^{d+1}$ ,  $z$  (with rand  $z \in \mathbb{F}^R$ ). Interpolate  $p(x)$  at  $\{f(x_i)\}_0^{d+1}$  to reconstruct the unique polynomial  $p$  of degree  $d$  such that  $p(x_i) = f(x_i) \forall i = 1, \dots, d + 1$ .

V checks  $p(z) = f(z)$ , if the check passes, then V is convinced with high probability.

This needs  $d + 2$  queries, is linear,  $\mathcal{O}(n)$ . With FRI we will have the test in  $\mathcal{O}(\log d)$ .

## 2 FRI protocol

Allows to test if a function  $f$  is a poly of degree  $\leq d$  in  $\mathcal{O}(\log d)$ .

Note: "P sends  $f(x)$  to V", "sends", in the ideal IOP model means that all the table of  $f(x)$  is sent, in practice is sent a commitment to  $f(x)$ .

### 2.1 Intuition

V wants to check that two functions  $g, h$  are both polynomials of degree  $\leq d$ .

Consider the following protocol:

1. V sends  $\alpha \in \mathbb{F}$  to P. P sends  $f(x) = g(x) + \alpha h(x)$  to V.
2. P sends  $f(x) = g(x) + \alpha h(x)$  to V.
3. V queries  $f(r), g(r), h(r)$  for rand  $r \in \mathbb{F}$ .
4. V checks  $f(r) = g(r) + \alpha h(r)$ . (Schwartz-Zippel lema). If holds, V can be certain that  $f(x) = g(x) + \alpha h(x)$ .
5. P proves that  $\deg(f) \leq d$ .
6. If V is convinced that  $\deg(f) \leq d$ , V believes that both  $g, h$  have  $\deg \leq d$ .

With high probability,  $\alpha$  will not cancel the coeffs with  $\deg \geq d + 1$ .

Let  $g(x) = a \cdot x^{d+1}$ ,  $h(x) = b \cdot x^{d+1}$ , and set  $f(x) = g(x) + \alpha h(x)$ . Imagine that P can chose  $\alpha$  such that  $ax^{d+1} + \alpha \cdot bx^{d+1} = 0$ , then, in  $f(x)$  the coefficients of degree  $d + 1$  would cancel.

Here, P proves  $g, h$  both have  $\deg \leq d$ , but instead of doing  $2 \cdot (d+2)$  queries ( $d+2$  for  $g$ , and  $d+2$  for  $h$ ), it is done in  $d+2$  queries (for  $f$ ). So we halved the number of queries.

### 2.2 FRI-LDT

FRI low degree testing.

Both P and V have oracle access to function  $f$ .

V wants to test if  $f$  is polynomial with  $\deg(f) \leq d$ .

Let  $f_0(x) = f(x)$ .

Each polynomial  $f(x)$  of degree that is a power of 2, can be written as

$$f(x) = f^L(x^2) + x f^R(x^2)$$

for some polynomials  $f^L, f^R$  of degree  $\frac{\deg(f)}{2}$ , each one containing the even and odd degree coefficients as follows:

$$f^L(x) = \sum_0^{\frac{d+1}{2}-1} c_{2i} x^i, \quad f^R(x) = \sum_0^{\frac{d+1}{2}-1} c_{2i+1} x^i$$

eg. for  $f(x) = x^4 + x^3 + x^2 + x + 1$ ,

$$\left. \begin{array}{l} f^L(x) = x^2 + x + 1 \\ f^R(x) = x + 1 \end{array} \right\} \begin{aligned} f(x) &= f^L(x^2) + x \cdot f^R(x^2) \\ &= (x^2)^2 + (x^2) + 1 + x \cdot ((x^2) + 1) \\ &= x^4 + x^2 + 1 + x^3 + x \end{aligned}$$

**Proof generation** (*Commitment phase*) P starts from  $f(x)$ , and for  $i = 0$  sets  $f_0(x) = f(x)$ .

1.  $\forall i \in \{0, \log(d)\}$ , with  $d = \deg f(x)$ ,  
P computes  $f_i^L(x)$ ,  $f_i^R(x)$  for which

$$f_i(x) = f_i^L(x^2) + x f_i^R(x^2) \quad (\text{eq. } A_i)$$

holds.

2. V sends challenge  $\alpha_i \in \mathbb{F}$
3. P commits to the random linear combination  $f_{i+1}$ , for

$$f_{i+1}(x) = f_i^L(x) + \alpha_i f_i^R(x) \quad (\text{eq. } B_i)$$

4. P sets  $f_i(x) := f_{i+1}(x)$  and starts again the iteration.

Notice that at each step,  $\deg(f_i)$  halves.

This is done until the last step, where  $f_i^L(x)$ ,  $f_i^R(x)$  are constant (degree 0 polynomials). For which P does not commit but gives their values directly to V.

(*Query phase*) P would receive a challenge  $z \in D$  set by V (where  $D$  is the evaluation domain,  $D \in \mathbb{F}$ ), and P would open the commitments at  $\{z^{2^i}, -z^{2^i}\}$  for each step  $i$ . (Recall, "opening" means that would provide a proof (MerkleProof) of it).

#### Data sent from P to V

Commitments:  $\{Comm(f_i)\}_0^{\log(d)}$   
eg.  $\{Comm(f_0), Comm(f_1), Comm(f_2), \dots, Comm(f_{\log(d)})\}$

Openings:  $\{f_i(z^{2^i}), f_i(-z^{2^i})\}_0^{\log(d)}$   
for a challenge  $z \in D$  set by V  
eg.  $f_0(z), f_0(-z), f_1(z^2), f_1(-z^2), f_2(z^4), f_2(-z^4), f_3(z^8), f_3(-z^8), \dots$

Constant values of last iteration:  $\{f_k^L, f_k^R\}$ , for  $k = \log(d)$

**Verification** V receives:

Commitments:  $Comm(f_i), \forall i \in \{0, \log(d)\}$

Openings:  $\{o_i, o'_i\} = \{f_i(z^{2^i}), f_i(-(z^{2^i}))\}, \forall i \in \{0, \log(d)\}$

Constant vals:  $\{f_k^L, f_k^R\}$

For all  $i \in \{0, \log(d)\}$ , V knows the openings at  $z^{2^i}$  and  $-(z^{2^i})$  for  $Comm(f_i(x))$ , which are  $o_i = f_i(z^{2^i})$  and  $o'_i = f_i(-(z^{2^i}))$  respectively.

V, from (eq.  $A_i$ ), knows that

$$f_i(x) = f_i^L(x^2) + x f_i^R(x^2)$$

should hold, thus

$$f_i(z) = f_i^L(z^2) + z f_i^R(z^2)$$

where  $f_i(z)$  is known, but  $f_i^L(z^2)$ ,  $f_i^R(z^2)$  are unknown. But, V also knows the value for  $f_i(-z)$ , which can be represented as

$$f_i(-z) = f_i^L(z^2) - z f_i^R(z^2)$$

(note that when replacing  $x$  by  $-z$ , it loses the negative in the power, not in the linear combination).

Thus, we have the system of independent linear equations

$$\begin{aligned} f_i(z) &= f_i^L(z^2) + z f_i^R(z^2) \\ f_i(-z) &= f_i^L(z^2) - z f_i^R(z^2) \end{aligned}$$

for which V will find the value of  $f_i^L(z^{2^i})$ ,  $f_i^R(z^{2^i})$ . Equivalently it can be represented by

$$\begin{pmatrix} 1 & z \\ 1 & -z \end{pmatrix} \begin{pmatrix} f_i^L(z^2) \\ f_i^R(z^2) \end{pmatrix} = \begin{pmatrix} f_i(z) \\ f_i(-z) \end{pmatrix}$$

where V will find the values of  $f_i^L(z^{2^i})$ ,  $f_i^R(z^{2^i})$  being

$$\begin{aligned} f_i^L(z^{2^i}) &= \frac{f_i(z) + f_i(-z)}{2} \\ f_i^R(z^{2^i}) &= \frac{f_i(z) - f_i(-z)}{2z} \end{aligned}$$

Once, V has computed  $f_i^L(z^{2^i})$ ,  $f_i^R(z^{2^i})$ , can use them to compute the linear combination of

$$f_{i+1}(z^{2^i}) = f_i^L(z^{2^i}) + \alpha_i f_i^R(z^{2^i})$$

obtaining then  $f_{i+1}(z^{2^i})$ . This comes from (eq.  $B_i$ ).

Now, V checks that the obtained  $f_{i+1}(z^{2^i})$  is equal to the received opening  $o_{i+1} = f_{i+1}(z^{2^i})$  from the commitment done by P. V checks also the commitment of  $Comm(f_{i+1}(x))$  for the opening  $o_{i+1} = f_{i+1}(z^{2^i})$ .  
 If the checks pass, V is convinced that  $f_1(x)$  was committed honestly.

Now, sets  $i := i + 1$  and starts a new iteration.

For the last iteration, V checks that the obtained  $f_i^L(z^{2^i}), f_i^R(z^{2^i})$  are equal to the constant values  $\{f_k^L, f_k^R\}$  received from P.

It needs  $\log(d)$  iterations, and the number of queries (commitments + openings sent and verified) needed is  $2 \cdot \log(d)$ .

### 2.3 Parameters

P commits to  $f_i$  restricted to a subfield  $F_0 \subset \mathbb{F}$ . Let  $0 < \rho < 1$  be the *rate* of the code, such that

$$|F_0| = \rho^{-1} \cdot d$$

**Thm 2.1.** For  $\delta \in (0, 1 - \sqrt{\rho})$ , we have that if V accepts, then w.v.h.p. (with very high probability)  $\Delta(f_0, p^d) \leq \delta$ .

## 3 FRI as polynomial commitment scheme

This section overviews the trick from [4] to convert FRI into a polynomial commitment.

Want to check that the evaluation of  $f(x)$  at  $r$  is  $f(r)$ , which is equivalent to proving that  $\exists Q \in \mathbb{F}[x]$  with  $\deg(Q) = d - 1$ , such that

$$f(x) - f(r) = Q(x) \cdot (x - r)$$

note that  $f(x) - f(r)$  evaluated at  $r$  is 0, so  $(x - r)|(f(x) - f(r))$ , in other words  $(f(x) - f(r))$  is a multiple of  $(x - r)$  for a polynomial  $Q(x)$ .

Let us define  $g(x) = \frac{f(x) - f(r)}{x - r}$ .

Prover uses FRI-LDT 2.2 to commit to  $g(x)$ , and then prove w.v.h.p that  $\deg(g) \leq d - 1$  ( $\iff \Delta(g, p^{d-1}) \leq \delta$ ).

Prover was already proving that  $\deg(f) \leq d$ .

Now, the missing thing to prove is that  $g(x)$  has the right shape. We can relate  $g$  to  $f$  as follows: V does the normal FRI-LDT, but in addition, at the first iteration: V has  $f(z)$  and  $g(z)$  openings, so can verify

$$g(z) = (f(z) - f(r)) \cdot (z - r)^{-1}$$

## 4 STIR (main idea)

*Update from 2024-03-22, notes from Héctor Masip Ardevol (<https://hecmas.github.io>) explanations.*

Let  $p \in \mathbb{F}[x]^{<n}$ .

In FRI we decompose  $p(x)$  as

$$p(x) = p_e(x^2) + x \cdot p_o(x^2)$$

with  $p_e, p_o \in \mathbb{F}[x]^{<n}$  containing the even and odd powers respectively.

The next FRI polynomial is

$$p_1(x) = p_e(x) + \alpha p_o(x)$$

for  $\alpha \in^R \mathbb{F}$ .

In STIR, this would be  $q(x) = x^2$ ,

$$Q(z, y) = p_e(y) + z \cdot p_o(y)$$

and then,  $p(x) = Q(x, q(x))$ . And  $Q$  fullfills the degree from Fact 4.6 from the STIR paper.

We can generalize to a  $q$  with bigger degree, or with another shape, and adapting  $Q$  on the choice of  $q$ .

eg. for  $q(x) = x^3$ , we can take

$$Q(z, y) = p_1(y) + z \cdot p_2(y) + z^2 \cdot p_3(y)$$

with  $p_1, p_2, p_3 \in \mathbb{F}[x]^{<n/3}$  with coefficients taken every 3 powers alternating.

## References

- [1] Vincenzo Iovino. <https://sites.google.com/site/vincenzoiovinoit/>.
- [2] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast reed-solomon interactive oracle proofs of proximity, 2018. <https://eccc.weizmann.ac.il/report/2017/134/>.
- [3] Ulrich Haböck. A summary on the fri low degree test. Cryptology ePrint Archive, Paper 2022/1216, 2022. <https://eprint.iacr.org/2022/1216>.
- [4] Alexander Vlasov and Konstantin Panarin. Transparent polynomial commitment scheme with polylogarithmic communication complexity. Cryptology ePrint Archive, Paper 2019/1020, 2019. <https://eprint.iacr.org/2019/1020>.
- [5] <https://github.com/arnaucube/fri-commitment>.
- [6] Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. STIR: Reed-solomon proximity testing with fewer queries. Cryptology ePrint Archive, Paper 2024/390, 2024. <https://eprint.iacr.org/2024/390>.
- [7] Héctor Masip Ardevol. <https://hecmas.github.io>.