

Revisiting the IPA-sumcheck connection

Ariel Gabizon (based on work with Liam Eagen)

Aztec Labs

Outline

- ▶ A few slides of motivation and context.
- ▶ Definitions reg multilinear polynomials.
- ▶ PCS landscape and where this work fits in.
- ▶ Construction.

Succinct arguments in a nutshell

Public program T , public output z .

Succinct arguments in a nutshell

Public program T , public output z .

Want to prove “I know input x for program T that generates output z .”

Succinct arguments in a nutshell

Public program T , public output z .

Want to prove “I know input x for program T that generates output z .”

Want proof size and verification time to be much smaller than run time of T .

(SNARK:=Succinct Non-Interactive Argument of Knowledge)

Succinct arguments in a nutshell

Public program T , public output z .

Want to prove “I know input x for program T that generates output z .”

Want proof size and verification time to be much smaller than run time of T .

(SNARK:=Succinct Non-Interactive Argument of Knowledge)

Arithmetization [LFKN,.....]: Reduce claim to claim of form “I know polynomials that satisfy some identity”

Succinct arguments in a nutshell

Public program T , public output z .

Want to prove “I know input x for program T that generates output z .”

Want proof size and verification time to be much smaller than run time of T .

(SNARK:=Succinct Non-Interactive Argument of Knowledge)

Arithmetization [LFKN,.....]: Reduce claim to claim of form “I know polynomials that satisfy some identity”

Succinct arguments in a nutshell

Advantage of claims about polynomials is that suffice to check at one random point

Succinct arguments in a nutshell

Advantage of claims about polynomials is that suffice to check at one random point

But need to solve "chicken and egg problem":
Prover must commit to polynomials before knowing the challenge point.

Multilinear polynomials

Let $\mathbf{X} = (X_1, \dots, X_k)$. By a (k -variate) multilinear polynomial $f(\mathbf{X})$ over \mathbb{F} we mean of *individual* degree at most one. e.g.

$$f(\mathbf{X}) = X_1 X_2 + X_3 - 5.$$

Multilinear polynomials

Let $\mathbf{X} = (X_1, \dots, X_k)$. By a (k -variate) multilinear polynomial $f(\mathbf{X})$ over \mathbb{F} we mean of *individual* degree at most one. e.g.

$$f(\mathbf{X}) = X_1X_2 + X_3 - 5.$$

We define the well-known **eq** multilinear polynomial in **2k** variables.

$$\text{eq}(\mathbf{x}, \mathbf{y}) := \prod_{i=1}^k (\mathbf{x}_i \mathbf{y}_i + (\mathbf{1} - \mathbf{x}_i)(\mathbf{1} - \mathbf{y}_i))$$

We define the well-known **eq** multilinear polynomial in **2k** variables.

$$\text{eq}(\mathbf{x}, \mathbf{y}) := \prod_{i=1}^k (\mathbf{x}_i \mathbf{y}_i + (\mathbf{1} - \mathbf{x}_i)(\mathbf{1} - \mathbf{y}_i))$$

We have for $\mathbf{x}, \mathbf{y} \in \{0, 1\}^k$, $\text{eq}(\mathbf{x}, \mathbf{y}) = \mathbf{1}$ when $\mathbf{x} = \mathbf{y}$ and $\text{eq}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$ otherwise.

We define the well-known **eq** multilinear polynomial in **2k** variables.

$$\mathbf{eq}(\mathbf{x}, \mathbf{y}) := \prod_{i=1}^k (\mathbf{x}_i \mathbf{y}_i + (\mathbf{1} - \mathbf{x}_i)(\mathbf{1} - \mathbf{y}_i))$$

We have for $\mathbf{x}, \mathbf{y} \in \{0, 1\}^k$, $\mathbf{eq}(\mathbf{x}, \mathbf{y}) = \mathbf{1}$ when $\mathbf{x} = \mathbf{y}$ and $\mathbf{eq}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$ otherwise.

For $\mathbf{z} \in \{0, 1\}^k$, the functions $\mathbf{L}_{\mathbf{z}}(\mathbf{X}) = \mathbf{eq}(\mathbf{z}, \mathbf{X})$ are a “Lagrange basis” for multilinear.

vectors and multilinear

$$n = 2^k.$$

$$f = (f_0, \dots, f_{n-1}) \in \mathbb{F}^n, z \in \mathbb{F}^k$$

$$\hat{f}(z) := \sum_{i < n} \text{eq}(i, z) f_i$$

vectors and multilinear

$$n = 2^k.$$

$$f = (f_0, \dots, f_{n-1}) \in \mathbb{F}^n, z \in \mathbb{F}^k$$

$$\hat{f}(z) := \sum_{i \ll n} \text{eq}(i, z) f_i$$

we identify i with its binary representation.

Polynomial commitment schemes

A PCS is consists of a

Polynomial commitment schemes

A PCS is consists of a

- ▶ Procedure **commit**: $f \in \mathbb{F}^n \rightarrow \mathbf{com}(f)$.

Polynomial commitment schemes

A PCS is consists of a

- ▶ Procedure **commit**: $\mathbf{f} \in \mathbb{F}^n \rightarrow \mathbf{com}(\mathbf{f})$.
- ▶ Protocol **open**($\mathbf{z} \in \mathbb{F}^k, \mathbf{v} \in \mathbb{F}, \mathbf{cm}$): \mathcal{P} convinces \mathcal{V} that it knows \mathbf{f} with $\mathbf{com}(\mathbf{f}) = \mathbf{cm}$ and $\hat{\mathbf{f}}(\mathbf{z}) = \mathbf{v}$.

PCS guarantees

binding: Efficient \mathcal{A} can't find $f_1 \neq f_2$ with $\text{com}(f_1) = \text{com}(f_2)$.

PCS guarantees

binding: Efficient \mathcal{A} can't find $f_1 \neq f_2$ with $\text{com}(f_1) = \text{com}(f_2)$.

Knowledge soundness: If efficient \mathcal{A} makes \mathcal{V} accept in $\text{open}(\text{cm}, z, v)$ it “knows” f with $\text{com}(f) = \text{cm}$ and $\hat{f}(z) = v$.

Discrete log hard groups

Additive group \mathbb{G} with $|\mathbf{G}| = |\mathbf{F}| = \mathbf{r}$ for prime \mathbf{r} .

Discrete log hard groups

Additive group \mathbb{G} with $|\mathbb{G}| = |\mathbb{F}| = r$ for prime r .

Uniformly chosen $G_0, \dots, G_{n-1} \in \mathbb{G}$. For any efficient \mathcal{A} the prob. of finding non-zero $\mathbf{f} \in \mathbb{F}^n$ with

$$\sum f_i G_i = 0$$

is $\text{negl}(\lambda)$.

The multilinear-PCS landscape

Scheme type
hash-based [BaseFold ,WHIR,..]
DL-based(IPA)[Bootle et. al,bulletproofs,..]
pairing-based [Mercury,Samaritan,Dory,KZH-fold,..]

The PCS landscape

Scheme type	Proof size
hash-based	polylog(n)
DL-based(IPA)	log n
pairing-based	$O(1)$

The PCS landscape

Scheme type	Commitment work
hash-based	Encode with ECC over small field
DL-based	MSM in DL-hard group
pairing-based	MSM in Pairing-friendly group

The PCS landscape

Scheme type	verification cost
hash-based	polylog field work
DL-based	$O(n)$-MSM!!
pairing-based	$O(1)$ pairings and scalar mult

The PCS landscape

accumulation := reducing checking several evaluation claims into one.

Scheme type	accumulation cost
hash-based	$O(\text{polylog } n)$
DL-based	$O(\log n)$
pairing-based	$O(1)$

This work:

Scheme type	verification cost
hash-based	polylog
DL-based	$O(n)$-MSM!! \rightarrow polylog
pairing-based	$O(1)$ pairings and scalar mult

The sumcheck protocol [LFKN]

k -variate poly $A(\mathbf{X})$ with ind. degree d , and
“target value” C . \mathcal{P} wants to prove

$$\sum_{\mathbf{b} \in \{0,1\}^k} A(\mathbf{b}) = C.$$

The sumcheck protocol [LFKN]

k -variate poly $A(\mathbf{X})$ with ind. degree d , and
“target value” C . \mathcal{P} wants to prove

$$\sum_{\mathbf{b} \in \{0,1\}^k} A(\mathbf{b}) = C.$$

Sumcheck reduces this with $O(dk)$ communication
to checking $A(\mathbf{r}) = V$ for random $\mathbf{r} \in \mathbb{F}^k$ and some
 V .

The sumcheck protocol [LFKN]

k -variate poly $A(\mathbf{X})$ with ind. degree d , and
“target value” C . \mathcal{P} wants to prove

$$\sum_{\mathbf{b} \in \{0,1\}^k} A(\mathbf{b}) = C.$$

Sumcheck reduces this with $O(dk)$ communication
to checking $A(\mathbf{r}) = V$ for random $\mathbf{r} \in \mathbb{F}^k$ and some
 V .

*[BCS21] showed sumcheck makes sense for polys over
groups/rings/modules...*

Polynomials over \mathbb{G}

Polynomials over \mathbb{G} $\mathbb{G}[X]$ consists of elements of the form $G(X) = \sum_{i < n} a_i X^i$ for integer $n \geq 0$ and $a_i \in \mathbb{G}$. Such G can be *evaluated* at $x \in \mathbb{F}$ -

$$G(x) := \sum_{i < n} a_i x^i \in \mathbb{G},$$

where $a_i x^i$ is defined as scalar multiplication of $a_i \in \mathbb{G}$ by $x^i \in \mathbb{F}$.

It's easy to check that non-zero $G(X) \in \mathbb{G}[X]$ of degree at most d evaluates to zero on at most d $x \in \mathbb{F}$. In abstract algebra terms, $\mathbb{G}[X]$ is a *module* over $\mathbb{F}[X]$. This concretely means for us that

1. $f, g \in \mathbb{G}[X]$ can be *added*:

$$\sum_{i < n} a_i X^i + \sum_{i < n} b_i X^i := \sum_{i < n} (a_i + b_i) X^i$$

2. More interestingly, $f \in \mathbb{F}[X]$ and $g \in \mathbb{G}[X]$ can be *multiplied* to get

$$\left(\sum_{i < n_1} a_i X^i \right) \cdot \left(\sum_{i < n_2} b_i X^i \right) := \sum_{0 \leq \ell < n_1 + n_2} \left(\sum_{i+j=\ell} a_i b_j \right) X^\ell \in \mathbb{G}[X]$$

MLPCS based on DL hardness as sumcheck [Bulletproofs, BCS21]:

Setup: Choose random non-zero

$$\mathbf{G} = (\mathbf{G}_0, \dots, \mathbf{G}_{n-1}) \in \mathbb{G}^n, \mathbf{P} \in \mathbb{G}.$$

MLPCS based on DL hardness as sumcheck [Bulletproofs, BCS21]:

Setup: Choose random non-zero

$$\mathbf{G} = (\mathbf{G}_0, \dots, \mathbf{G}_{n-1}) \in \mathbb{G}^n, \mathbf{P} \in \mathbb{G}.$$

Commitment: $\mathbf{f} \in \mathbb{F}^n$, $\mathbf{com}(\mathbf{f}) = \sum_{i < n} f_i \mathbf{G}_i$.

MLPCS based on DL hardness as sumcheck [Bulletproofs, BCS21]:

Setup: Choose random non-zero

$$\mathbf{G} = (\mathbf{G}_0, \dots, \mathbf{G}_{n-1}) \in \mathbb{G}^n, \mathbf{P} \in \mathbb{G}.$$

Commitment: $\mathbf{f} \in \mathbb{F}^n$, $\mathbf{com}(\mathbf{f}) = \sum_{i < n} f_i \mathbf{G}_i$.

Openings: next slide.

Given $\mathbf{cm} \in \mathbb{G}, z \in \mathbb{F}^k, \mathbf{v} \in \mathbb{F}$ want to prove $\mathbf{com}(f) = \mathbf{cm}$ and $\hat{f}(z) = \mathbf{v}$.

Given $\mathbf{cm} \in \mathbb{G}$, $z \in \mathbb{F}^k$, $\mathbf{v} \in \mathbb{F}$ want to prove $\mathbf{com}(\mathbf{f}) = \mathbf{cm}$ and $\hat{\mathbf{f}}(z) = \mathbf{v}$.

Define the polynomial

$$\mathbf{A}(\mathbf{X}) := \hat{\mathbf{f}}(\mathbf{X})\hat{\mathbf{G}}(\mathbf{X}) + \mathbf{eq}(\mathbf{X}, z)\hat{\mathbf{f}}(\mathbf{X})\mathbf{P}$$

Given $\mathbf{cm} \in \mathbb{G}$, $\mathbf{z} \in \mathbb{F}^k$, $\mathbf{v} \in \mathbb{F}$ want to prove $\mathbf{com}(\mathbf{f}) = \mathbf{cm}$ and $\hat{\mathbf{f}}(\mathbf{z}) = \mathbf{v}$.

Define the polynomial

$$\mathbf{A}(\mathbf{X}) := \hat{\mathbf{f}}(\mathbf{X})\hat{\mathbf{G}}(\mathbf{X}) + \mathbf{eq}(\mathbf{X}, \mathbf{z})\hat{\mathbf{f}}(\mathbf{X})\mathbf{P}$$

When claim holds:

$$\begin{aligned} & \sum_{\mathbf{b} \in \{0,1\}^k} \hat{\mathbf{f}}(\mathbf{b})\hat{\mathbf{G}}(\mathbf{b}) + \mathbf{eq}(\mathbf{b}, \mathbf{z})\hat{\mathbf{f}}(\mathbf{b})\mathbf{P} \\ &= \sum_{\mathbf{i} < \mathbf{n}} \mathbf{f}_i \mathbf{G}_i + \mathbf{eq}(\mathbf{i}, \mathbf{z})\mathbf{f}_i \mathbf{P} = \mathbf{cm} + \hat{\mathbf{f}}(\mathbf{z})\mathbf{P} \end{aligned}$$

\mathcal{P} and \mathcal{V} will run sumcheck on \mathbf{A} with target value $\mathbf{cm} + \mathbf{vP}$.

\mathcal{P} and \mathcal{V} will run sumcheck on \mathbf{A} with target value $\mathbf{cm} + \mathbf{vP}$.

At end of sumcheck, \mathcal{V} needs to evaluate $\mathbf{A}(\mathbf{r}) = \hat{\mathbf{f}}(\mathbf{r})\hat{\mathbf{G}}(\mathbf{r}) + \mathbf{eq}(\mathbf{r}, \mathbf{z})\hat{\mathbf{f}}(\mathbf{r})\mathbf{P}$ for some $\mathbf{r} \in \mathbb{F}^k$.

\mathcal{P} and \mathcal{V} will run sumcheck on \mathbf{A} with target value $\mathbf{cm} + \mathbf{vP}$.

At end of sumcheck, \mathcal{V} needs to evaluate $\mathbf{A}(\mathbf{r}) = \hat{\mathbf{f}}(\mathbf{r})\hat{\mathbf{G}}(\mathbf{r}) + \mathbf{eq}(\mathbf{r}, \mathbf{z})\hat{\mathbf{f}}(\mathbf{r})\mathbf{P}$ for some $\mathbf{r} \in \mathbb{F}^k$.

\mathcal{V} computes $\mathbf{eq}(\mathbf{r}, \mathbf{z}), \hat{\mathbf{G}}(\mathbf{r})$. \mathcal{P} *simply sends* $\mathbf{a} = \hat{\mathbf{f}}(\mathbf{r})$.

\mathcal{P} and \mathcal{V} will run sumcheck on \mathbf{A} with target value $\mathbf{cm} + \mathbf{vP}$.

At end of sumcheck, \mathcal{V} needs to evaluate $\mathbf{A}(\mathbf{r}) = \hat{\mathbf{f}}(\mathbf{r})\hat{\mathbf{G}}(\mathbf{r}) + \mathbf{eq}(\mathbf{r}, \mathbf{z})\hat{\mathbf{f}}(\mathbf{r})\mathbf{P}$ for some $\mathbf{r} \in \mathbb{F}^k$.

\mathcal{V} computes $\mathbf{eq}(\mathbf{r}, \mathbf{z}), \hat{\mathbf{G}}(\mathbf{r})$. \mathcal{P} simply sends $\mathbf{a} = \hat{\mathbf{f}}(\mathbf{r})$.

The strange (and useful) thing: When \mathbb{G} has hard discrete-log this is sound!

\mathcal{P} and \mathcal{V} will run sumcheck on \mathbf{A} with target value $\mathbf{cm} + \mathbf{vP}$.

At end of sumcheck, \mathcal{V} needs to evaluate $\mathbf{A}(\mathbf{r}) = \hat{\mathbf{f}}(\mathbf{r})\hat{\mathbf{G}}(\mathbf{r}) + \mathbf{eq}(\mathbf{r}, \mathbf{z})\hat{\mathbf{f}}(\mathbf{r})\mathbf{P}$ for some $\mathbf{r} \in \mathbb{F}^k$.

\mathcal{V} computes $\mathbf{eq}(\mathbf{r}, \mathbf{z}), \hat{\mathbf{G}}(\mathbf{r})$. \mathcal{P} simply sends $\mathbf{a} = \hat{\mathbf{f}}(\mathbf{r})$.

The strange (and useful) thing: When \mathbb{G} has hard discrete-log this is sound!

Drawback: Computing $\hat{\mathbf{G}}(\mathbf{r})$ is n -size MSM for \mathcal{V} !

Mitigation from Halo: defer MSM

Given claims $\hat{\mathbf{G}}(\mathbf{r}_1) = \mathbf{V}_1, \hat{\mathbf{G}}(\mathbf{r}_2) = \mathbf{V}_2$. Can reduce them into one using sumcheck:

Mitigation from Halo: defer MSM

Given claims $\hat{\mathbf{G}}(\mathbf{r}_1) = \mathbf{V}_1, \hat{\mathbf{G}}(\mathbf{r}_2) = \mathbf{V}_2$. Can reduce them into one using sumcheck:

Recall

$$\hat{\mathbf{G}}(\mathbf{r}) = \sum_{\mathbf{b} \in \{0,1\}^k} \text{eq}(\mathbf{b}, \mathbf{r}) \hat{\mathbf{G}}(\mathbf{b})$$

Mitigation from Halo: defer MSM

Given claims $\hat{\mathbf{G}}(\mathbf{r}_1) = \mathbf{V}_1$, $\hat{\mathbf{G}}(\mathbf{r}_2) = \mathbf{V}_2$. Can reduce them into one using sumcheck:

Recall

$$\hat{\mathbf{G}}(\mathbf{r}) = \sum_{\mathbf{b} \in \{0,1\}^k} \text{eq}(\mathbf{b}, \mathbf{r}) \hat{\mathbf{G}}(\mathbf{b})$$

\mathcal{V} chooses random γ . Let

$$\mathbf{A}(\mathbf{X}) := (\text{eq}(\mathbf{X}, \mathbf{r}_1) + \gamma \cdot \text{eq}(\mathbf{X}, \mathbf{r}_2)) \hat{\mathbf{G}}(\mathbf{X}).$$

Mitigation from Halo: defer MSM

Given claims $\hat{\mathbf{G}}(\mathbf{r}_1) = \mathbf{V}_1$, $\hat{\mathbf{G}}(\mathbf{r}_2) = \mathbf{V}_2$. Can reduce them into one using sumcheck:

Recall

$$\hat{\mathbf{G}}(\mathbf{r}) = \sum_{\mathbf{b} \in \{0,1\}^k} \text{eq}(\mathbf{b}, \mathbf{r}) \hat{\mathbf{G}}(\mathbf{b})$$

\mathcal{V} chooses random γ . Let

$$\mathbf{A}(\mathbf{X}) := (\text{eq}(\mathbf{X}, \mathbf{r}_1) + \gamma \cdot \text{eq}(\mathbf{X}, \mathbf{r}_2)) \hat{\mathbf{G}}(\mathbf{X}).$$

We have

$$\sum_{\mathbf{b} \in \{0,1\}^k} \mathbf{A}(\mathbf{X}) = \hat{\mathbf{G}}(\mathbf{r}_1) + \gamma \hat{\mathbf{G}}(\mathbf{r}_2)$$

Mitigation from Halo: defer MSM

Given claims $\hat{\mathbf{G}}(\mathbf{r}_1) = \mathbf{V}_1$, $\hat{\mathbf{G}}(\mathbf{r}_2) = \mathbf{V}_2$. Can reduce them into one using sumcheck:

Recall

$$\hat{\mathbf{G}}(\mathbf{r}) = \sum_{\mathbf{b} \in \{0,1\}^k} \text{eq}(\mathbf{b}, \mathbf{r}) \hat{\mathbf{G}}(\mathbf{b})$$

\mathcal{V} chooses random γ . Let

$$\mathbf{A}(\mathbf{X}) := (\text{eq}(\mathbf{X}, \mathbf{r}_1) + \gamma \cdot \text{eq}(\mathbf{X}, \mathbf{r}_2)) \hat{\mathbf{G}}(\mathbf{X}).$$

We have

$$\sum_{\mathbf{b} \in \{0,1\}^k} \mathbf{A}(\mathbf{X}) = \hat{\mathbf{G}}(\mathbf{r}_1) + \gamma \hat{\mathbf{G}}(\mathbf{r}_2)$$

Reducing $\hat{\mathbf{G}}(\mathbf{r}_1) \stackrel{?}{=} \mathbf{V}_1, \hat{\mathbf{G}}(\mathbf{r}_2) \stackrel{?}{=} \mathbf{V}_2$ to $\hat{\mathbf{G}}(\mathbf{r}) \stackrel{?}{=} \mathbf{V}$:

1. \mathcal{V} chooses random γ .
2. Let
 $\mathbf{A}(\mathbf{X}) := (\text{eq}(\mathbf{X}, \mathbf{r}_1) + \gamma \cdot \text{eq}(\mathbf{X}, \mathbf{r}_2)) \hat{\mathbf{G}}(\mathbf{X})$.
3. \mathcal{P} and \mathcal{V} run sumcheck on \mathbf{A} with target value $\mathbf{V}_1 + \gamma \mathbf{V}_2$.

Reducing $\hat{\mathbf{G}}(\mathbf{r}_1) \stackrel{?}{=} \mathbf{V}_1, \hat{\mathbf{G}}(\mathbf{r}_2) \stackrel{?}{=} \mathbf{V}_2$ to $\hat{\mathbf{G}}(\mathbf{r}) \stackrel{?}{=} \mathbf{V}$:

1. \mathcal{V} chooses random γ .
2. Let $\mathbf{A}(\mathbf{X}) := (\text{eq}(\mathbf{X}, \mathbf{r}_1) + \gamma \cdot \text{eq}(\mathbf{X}, \mathbf{r}_2)) \hat{\mathbf{G}}(\mathbf{X})$.
3. \mathcal{P} and \mathcal{V} run sumcheck on \mathbf{A} with target value $\mathbf{V}_1 + \gamma \mathbf{V}_2$.
4. Claim is reduced to $\mathbf{A}(\mathbf{r}) = \mathbf{V}$, for some $\mathbf{r} \in \mathbb{F}, \mathbf{V}' \in \mathbb{G}$.

Reducing $\hat{\mathbf{G}}(\mathbf{r}_1) \stackrel{?}{=} \mathbf{V}_1, \hat{\mathbf{G}}(\mathbf{r}_2) \stackrel{?}{=} \mathbf{V}_2$ to $\hat{\mathbf{G}}(\mathbf{r}) \stackrel{?}{=} \mathbf{V}$:

1. \mathcal{V} chooses random γ .
2. Let $\mathbf{A}(\mathbf{X}) := (\text{eq}(\mathbf{X}, \mathbf{r}_1) + \gamma \cdot \text{eq}(\mathbf{X}, \mathbf{r}_2)) \hat{\mathbf{G}}(\mathbf{X})$.
3. \mathcal{P} and \mathcal{V} run sumcheck on \mathbf{A} with target value $\mathbf{V}_1 + \gamma \mathbf{V}_2$.
4. Claim is reduced to $\mathbf{A}(\mathbf{r}) = \mathbf{V}$, for some $\mathbf{r} \in \mathbb{F}, \mathbf{V}' \in \mathbb{G}$.
5. \mathcal{V} computes $\text{eq}(\mathbf{r}, \mathbf{r}_1), \text{eq}(\mathbf{r}, \mathbf{r}_2)$ reducing the claim to $\hat{\mathbf{G}}(\mathbf{r}) = \mathbf{V}$ for some \mathbf{V} .

\mathcal{P} proving correctness of $\hat{\mathbf{G}}(\mathbf{r})$

Observation: If we have mlPCS for field-valued multilinear, where all ops on \mathbf{f} 's vals are \mathbb{F} -linear, can also use on group valued multilinear \mathbf{G} . - e.g.

BaseFold

FRI/ BaseFold

Idea: Think of $(\mathbf{G}_0, \dots, \mathbf{G}_{n-1})$ as *univariate*

$$g(X) := \sum_{i=0}^{n-1} \mathbf{G}_i X^i$$

FRI/ BaseFold

Idea: Think of $(\mathbf{G}_0, \dots, \mathbf{G}_{n-1})$ as *univariate*

$$g(X) := \sum_{i=0}^{n-1} \mathbf{G}_i X^i$$

Send a merkle commitment to the values of g on a $2n$ -order subgroup.

FRI/ BaseFold

Idea: Think of $(\mathbf{G}_0, \dots, \mathbf{G}_{n-1})$ as *univariate*

$$g(X) := \sum_{i=0}^{n-1} G_i X^i$$

Send a merkle commitment to the values of g on a $2n$ -order subgroup.

Recall we have

$$g(X) = g_{\text{even}}(X^2) + X g_{\text{odd}}(X^2)$$

FRI/ BaseFold

Idea: Think of $(\mathbf{G}_0, \dots, \mathbf{G}_{n-1})$ as *univariate*

$$g(\mathbf{X}) := \sum_{i=0}^{n-1} \mathbf{G}_i \mathbf{X}^i$$

Send a merkle commitment to the values of g on a $2n$ -order subgroup.

Recall we have

$$g(\mathbf{X}) = g_{\text{even}}(\mathbf{X}^2) + \mathbf{X}g_{\text{odd}}(\mathbf{X}^2)$$

Let $\mathbf{r} = (\mathbf{r}_1, \dots, \mathbf{r}_k)$. “Fold” g by \mathbf{r}_1 :

$$g_1(\mathbf{X}) = (\mathbf{1} - \mathbf{r}_1)g_{\text{even}}(\mathbf{X}) + \mathbf{r}_1g_{\text{odd}}(\mathbf{X})$$

Let $\mathbf{r} = (\mathbf{r}_1, \dots, \mathbf{r}_k)$. “Fold” \mathbf{g} by \mathbf{r}_1 :

$$\mathbf{g}_1(\mathbf{X}) := (\mathbf{1} - \mathbf{r}_1)\mathbf{g}_{\text{even}}(\mathbf{X}) + \mathbf{r}_1\mathbf{g}_{\text{odd}}(\mathbf{X})$$

Let $\mathbf{r} = (\mathbf{r}_1, \dots, \mathbf{r}_k)$. “Fold” \mathbf{g} by \mathbf{r}_1 :

$$\mathbf{g}_1(\mathbf{X}) := (\mathbf{1} - \mathbf{r}_1)\mathbf{g}_{\text{even}}(\mathbf{X}) + \mathbf{r}_1\mathbf{g}_{\text{odd}}(\mathbf{X})$$

\mathcal{P} sends Merkle commitment to values of \mathbf{g}_1 on \mathbf{n} -order subgroup.

\mathcal{V} checks on random locations \mathbf{g}_1 is the correct folding.

Let $\mathbf{r} = (\mathbf{r}_1, \dots, \mathbf{r}_k)$. “Fold” \mathbf{g} by \mathbf{r}_1 :

$$\mathbf{g}_1(\mathbf{X}) := (\mathbf{1} - \mathbf{r}_1)\mathbf{g}_{\text{even}}(\mathbf{X}) + \mathbf{r}_1\mathbf{g}_{\text{odd}}(\mathbf{X})$$

\mathcal{P} sends Merkle commitment to values of \mathbf{g}_1 on \mathbf{n} -order subgroup.

\mathcal{V} checks on random locations \mathbf{g}_1 is the correct folding.

Now we have

$$\hat{\mathbf{G}}(\mathbf{r}) = \hat{\mathbf{g}}_1(\mathbf{r}_2, \dots, \mathbf{r}_k)$$

Let $\mathbf{r} = (\mathbf{r}_1, \dots, \mathbf{r}_k)$. “Fold” \mathbf{g} by \mathbf{r}_1 :

$$\mathbf{g}_1(\mathbf{X}) := (\mathbf{1} - \mathbf{r}_1)\mathbf{g}_{\text{even}}(\mathbf{X}) + \mathbf{r}_1\mathbf{g}_{\text{odd}}(\mathbf{X})$$

\mathcal{P} sends Merkle commitment to values of \mathbf{g}_1 on \mathbf{n} -order subgroup.

\mathcal{V} checks on random locations \mathbf{g}_1 is the correct folding.

Now we have

$$\hat{\mathbf{G}}(\mathbf{r}) = \hat{\mathbf{g}}_1(\mathbf{r}_2, \dots, \mathbf{r}_k)$$

Now we have

$$\hat{\mathbf{G}}(\mathbf{r}) = \hat{\mathbf{g}}_1(\mathbf{r}_2, \dots, \mathbf{r}_k)$$

Now we have

$$\hat{G}(\mathbf{r}) = \hat{g}_1(r_2, \dots, r_k)$$

Correlated Agreement Thm: This is sound for random \mathbf{r} .

Now we have

$$\hat{G}(\mathbf{r}) = \hat{g}_1(r_2, \dots, r_k)$$

Correlated Agreement Thm: This is sound for random \mathbf{r} .

BaseFold: Interleave with sumcheck to work for *all* \mathbf{r}

Now we have

$$\hat{G}(\mathbf{r}) = \hat{g}_1(\mathbf{r}_2, \dots, \mathbf{r}_k)$$

Correlated Agreement Thm: This is sound for random \mathbf{r} .

BaseFold: Interleave with sumcheck to work for *all* \mathbf{r}

Future work: Improve using WHIR

Revisiting the IPA-sumcheck connection

Liam Eagen^{1,2} and Ariel Gabizon²

¹Alpen Labs

²Aztec Labs

August 24, 2025

Abstract

Inner Product Arguments (IPA) [BCC⁺16, BBB⁺17] are a family of proof systems with $O(\log n)$ sized proofs, $O(n)$ time verifiers, and transparent setup. Bootle, Chiesa and Sotiraki [BCS21] observed that an IPA can be viewed as a sumcheck protocol [LFKN92] *where the summed polynomial is allowed to have coefficients in a group rather than a field*. We leverage this viewpoint to improve the performance of multi-linear polynomial commitments based on IPA. Specifically,

1. We introduce a simplified variant of Halo-style accumulation that works for multilinear evaluation claims, rather than only univariate ones as in [BGH19, BCMS20].
2. We show that the size n MSM the IPA verifier performs can be replaced by a “group variant” of BaseFold [ZCF23]. This reduces the verifier complexity from $O(n)$ to $O_\lambda(\log^2 n)$ time at the expense of an additional $4n$ scalar multiplications for the IPA prover.

See paper on eprint for details. Thanks!