Notes on ZeroMorph

- Yu Guo yu.guo@secbit.io

ZeroMorph [KT23] is an MLE polynomial commitment scheme based on KZG10. In fact, the ZeroMorph scheme is a more general framework that can be based on different Univariate Polynomial Commitment schemes, such as the FRI-based ZeroMorph scheme.

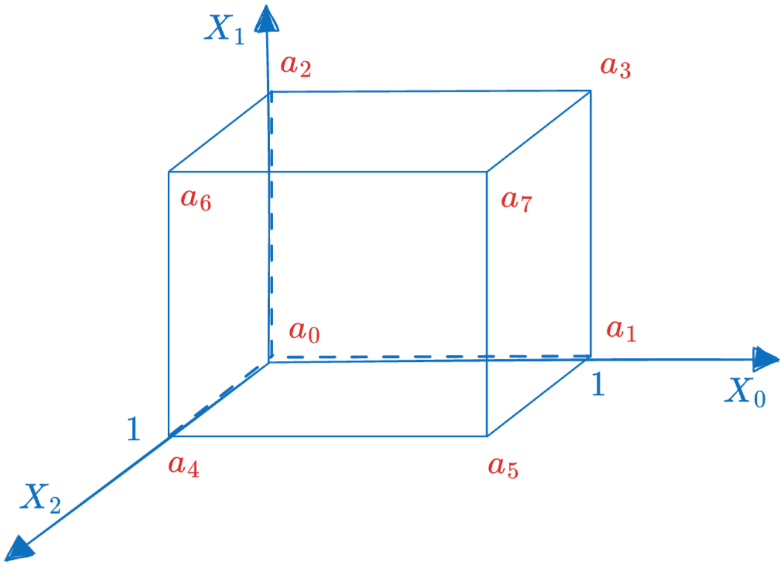
The core idea of ZeroMorph is to use the Evaluations of MLE polynomials, i.e., the "point value vector", as the "coefficient vector" of Univariate polynomials. This approach may seem strange, but the framework remains clear and concise.

The key to understanding ZeroMorph lies in understanding the transformations of values on high-dimensional Boolean HyperCubes and how they correspond to operations on Univariate polynomials.

MLE Polynomials

An MLE (Multilinear Extension) polynomial $\tilde{f}$ is a class of Multivariate polynomials defined on the Boolean HyperCube. The degree of any variable in each term does not exceed 1. For example, $\tilde{f} = 1 + 2X_0 + 3X_1X_0$ is an MLE polynomial, while $\tilde{f}' = 1 + 2X_0^2 + 3X_1X_0 + X_1$ is not, because the degree of X_0^2 is greater than 1.

An MLE polynomial can correspond to a function from Boolean vectors to a finite field, i.e., $f : \{0,1\}^n \to \mathbb{F}_q$, and we call its dimension n. The following figure is an example of a three-dimensional MLE polynomial $\tilde{f}(X_0, X_1, X_2)$, which can be uniquely represented by the "point value vector" $(a_0, a_1, \ldots, a_7)$. This corresponds to the "point value form" representation in Univariate polynomials, i.e., the Evaluations form.



Of course, an MLE polynomial can also be represented in "coefficient form", i.e., Coefficients form, as follows:

$$\tilde{f}(X_0, X_1, \ldots, X_{n-1}) = \sum_{i_0=0}^{1}\sum_{i_1=0}^{1}\cdots\sum_{i_{n-1}=0}^{1} f_{i_0 i_1 \cdots i_{n-1}} X_0^{i_0} X_1^{i_1} \cdots X_{n-1}^{i_{n-1}} \tag{1}$$

For the example of the three-dimensional MLE polynomial above, we can write it as:

$$\tilde{f}(X_0, X_1, X_2) = f_0 + f_1X_0 + f_2X_1 + f_3X_2 + f_4X_0X_1 + f_5X_0X_2 + f_6X_1X_2 + f_7X_0X_1X_2 \tag{2}$$

where $(f_0, f_1, \ldots, f_7)$ is the coefficient vector of the MLE polynomial. Note that because MLE polynomials belong to multivariate polynomials, any representation requires determining the ordering of terms in the polynomial in advance. In this article and subsequent discussions, we will base our approach on Lexicographic Order. For the "point value form" representation of MLE polynomials, we can define it as:

$$\tilde{f}(X_0, X_1, \ldots, X_{n-1}) = \sum_{i_0=0}^{1}\sum_{i_1=0}^{1}\cdots\sum_{i_{n-1}=0}^{1} a_{i_0 i_1 \cdots i_{n-1}} \cdot eq(i_0, i_1, \ldots, i_{n-1}, X_0, X_1, \ldots, X_{n-1}) \tag{3}$$

where eq is a set of Lagrange Polynomials for the n-dimensional Boolean HyperCube $\{0,1\}^n$:

$$eq(i_0, i_1, \ldots, i_{n-1}, X_0, X_1, \ldots, X_{n-1}) = \prod_{j=0}^{n-1} \Big((1 - i_j) \cdot (1 - X_j) + i_j \cdot X_j \Big) \tag{4}$$

There exists an $N\log(N)$ conversion algorithm between the "point value form" and "coefficient form" of MLE polynomials, which we will not discuss in depth here.

We can use ZeroMorph to map an MLE polynomial to a Univariate polynomial, more specifically, to map the "point value vector" of the MLE polynomial on the Boolean HyperCube to the "coefficient vector" of a Univariate polynomial.

MLE Polynomial to Univariate Polynomial

Let's use a simple example to quickly understand this mapping relationship. Consider a MLE polynomial of dimension 2:

$$\tilde{f}(X_0, X_1) = 2 + X_1 + X_0 X_1 \tag{5}$$

It's easy to verify that its point value representation on the Boolean HyperCube is:

$$\begin{aligned} \tilde{f}(0,0) &= 2 \\ \tilde{f}(1,0) &= 2 \\ \tilde{f}(0,1) &= 3 \\ \tilde{f}(1,1) &= 4 \end{aligned} \tag{6}$$

If we adopt the ZeroMorph scheme, it can be mapped to the following Univariate polynomial:

$$\hat{f}(X) = 2 + 2 \cdot X + 3 \cdot X^2 + 4 \cdot X^3 \tag{7}$$

Assuming we have a Univariate polynomial commitment scheme, we can then calculate the commitment of the mapped Univariate polynomial. For example, suppose we have the following KZG10 commitment scheme SRS:

$$SRS = ([1]_1, [\tau]_1, [\tau^2]_1, [\tau^3]_1, \ldots, [\tau^D]_1, [1]_2, [\tau]_2, [\tau^2]_2, [\tau^3]_2, \ldots, [\tau^D]_2) \tag{8}$$

According to the KZG10 commitment algorithm, we calculate the commitment of $\hat{f}(X)$ as follows:

$$\mathsf{cm}(\hat{f}) = 2 \cdot [1]_1 + 2 \cdot [\tau]_1 + 3 \cdot [\tau^2]_1 + 4 \cdot [\tau^3]_1 \tag{9}$$

In the following sections, we will use the symbol $[[\tilde{f}]]$ to represent the Univariate polynomial corresponding to the mapping of the MLE polynomial $\tilde{f}$.

Polynomial Mapping

In this section, we will discuss more mapping situations. For simplicity, let's first consider the case of three-dimensional MLE, i.e., $\tilde{f} \in \mathbb{F}_q[X_0, X_1, X_2]^{\leq 1}$.

Suppose $\tilde{f}$ is just a constant polynomial, meaning its coefficient vector only has the first term non-zero, and all other elements are zero. The polynomial can be represented as:

$$\tilde{c}(X_0, X_1, X_2) = c_0 \tag{10}$$

Let's consider what kind of Univariate polynomial this constant polynomial would map to. First, we need to convert it to point value form. Consider a three-dimensional Boolean HyperCube, regardless of how $X_0, X_1, X_2 \in \{0, 1\}$ are valued, this polynomial always evaluates to c_0 on the Boolean HyperCube. This means its point value form is $(c_0, c_0, c_0, \ldots, c_0)$, so its corresponding Univariate polynomial is:

$$\begin{aligned} [[\tilde{c}]] &= c_0 + c_0 X + c_0 X^2 + c_0 X^3 + \ldots + c_0 X^7 \\ &= c_0 \cdot (1 + X + X^2 + X^3 + \ldots + X^7) \end{aligned} \tag{11}$$

Now let's consider a two-dimensional MLE polynomial $\tilde{c}'(X_0, X_1)$, which is also a constant polynomial, i.e., $\tilde{c}'(X_0, X_1) = c_0$. Its corresponding Univariate polynomial is:

$$\begin{aligned} [[\tilde{c}']] &= c_0 + c_0 X + c_0 X^2 + c_0 X^3 \\ &= c_0 \cdot (1 + X + X^2 + X^3) \end{aligned} \tag{12}$$

We can see that although the coefficient form representations of the two MLE polynomials $\tilde{c}$ and $\tilde{c}'$ are completely the same, the Univariate polynomials they map to are different. This is because for both Univariate and Multivariate polynomials, their point value form representations implicitly include the selection of the Evaluation Domain. The Evaluation Domain of $\tilde{c}$ is a 3-dimensional Boolean HyperCube, while the Evaluation Domain of $\tilde{c}'$ is a 2-dimensional Boolean HyperCube. Therefore, when we calculate the point value form of polynomials, we need to clarify the choice of Evaluation Domain. For MLE polynomials, if their Evaluation Domain is an n-dimensional Boolean HyperCube, we modify the mapping notation by adding a subscript n to the mapping brackets, i.e., $[[\tilde{f}]]_n$. Below are the two different Univariate polynomials produced by the mapping of $\tilde{c}$ on two different Evaluation Domains:

$$\begin{aligned}
[[\tilde{c}]]_2 &= c_0 + c_0X + c_0X^2 + c_0X^3 \\
[[\tilde{c}]]_3 &= c_0 + c_0X + c_0X^2 + c_0X^3 + c_0X^4 + c_0X^5 + c_0X^6 + c_0X^7
\end{aligned} \tag{13}$$

Additive Homomorphism of Mapping

For any two MLE polynomials, if they have the same dimension, such as $\tilde{f}_1(X_0, X_1)$ and $\tilde{f}_2(X_0, X_1)$, suppose the point value form representation of the former is

$$\tilde{f}_1(X_0, X_1) = v_0 \cdot eq(0,0,X_0,X_1) + v_1 \cdot eq(0,1,X_0,X_1) + v_2 \cdot eq(1,0,X_0,X_1) + v_3 \cdot eq(1,1,X_0,X_1) \tag{14}$$

$$\tilde{f}_2(X_0, X_1) = v_0' \cdot eq(0,0,X_0,X_1) + v_1' \cdot eq(0,1,X_0,X_1) + v_2' \cdot eq(1,0,X_0,X_1) + v_3' \cdot eq(1,1,X_0,X_1) \tag{15}$$

Then their sum is: $\tilde{f}_1(X_0, X_1) + \tilde{f}_2(X_0, X_1)$, and its point value form is:

$$\begin{aligned}
\tilde{f}_1(X_0, X_1) + \tilde{f}_2(X_0, X_1) = (v_0 + v_0') \cdot eq(0,0,X_0,X_1) + (v_1 + v_1') \cdot eq(0,1,X_0,X_1) \\
+ (v_2 + v_2') \cdot eq(1,0,X_0,X_1) + (v_3 + v_3') \cdot eq(1,1,X_0,X_1)
\end{aligned} \tag{16}$$

Thus, the following equation holds:

$$[[\tilde{f}_1(X_0, X_1) + \tilde{f}_2(X_0, X_1)]]_2 = [[\tilde{f}_1(X_0, X_1)]]_2 + [[\tilde{f}_2(X_0, X_1)]]_2 \tag{17}$$

It's not hard to prove that the above equation holds for MLE polynomials of any same dimension. It's also easy to prove:

$$[[\alpha \cdot \tilde{f}]]_n = \alpha \cdot [[\tilde{f}]]_n, \quad \forall \alpha \in \mathbb{F}_q \tag{18}$$

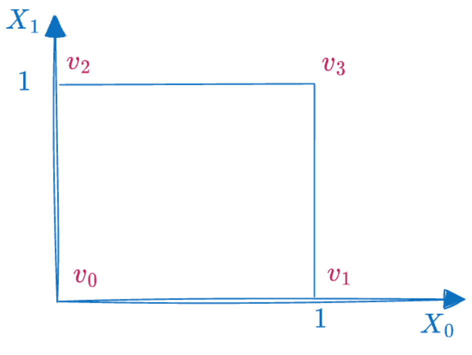
Therefore, we say that the mapping $[[\tilde{f}]]_n$ has polynomial additive homomorphism and is a one-to-one mapping (Injective and Surjective).

Low Dimension to High Dimension Mapping

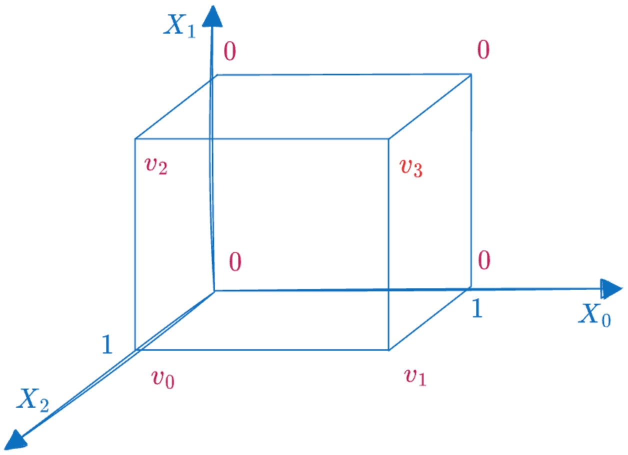
Let's consider a more general polynomial case. Suppose a two-dimensional MLE polynomial $\tilde{c}(X_0, X_1)$ has values (v_0, v_1, v_2, v_3) on a two-dimensional Boolean HyperCube. Then its corresponding Univariate polynomial is:

$$[[\tilde{c}]]_2 = v_0 + v_1X + v_2X^2 + v_3X^3 \tag{19}$$

And $X_2 \cdot \tilde{c}(X_0, X_1)$ is also an MLE polynomial, with dimension 3. Its values on the Boolean HyperCube are: $(0, 0, 0, 0, v_0, v_1, v_2, v_3)$, i.e., the first four terms are zero, and the last four terms are equal to the values of $\tilde{c}(X_0, X_1)$ in the two-dimensional MLE polynomial, as shown in the following figure:



$\tilde{c}(X_0, X_1)$



$X_2 \cdot \tilde{c}(X_0, X_1)$

This is easy to explain because when $X_2 = 0$, the overall polynomial value is zero, so the values at the vertices of the square formed by X_0, X_1 are all zero. When $X_2 = 1$, the polynomial $X_2 \cdot \tilde{c}(X_0, X_1)$ equals $\tilde{c}(X_0, X_1)$. Therefore, the values at the vertices of the square plane where $X_2 = 1$ are equal to $\tilde{c}(X_0, X_1)$. Furthermore, we can draw the following conclusion:

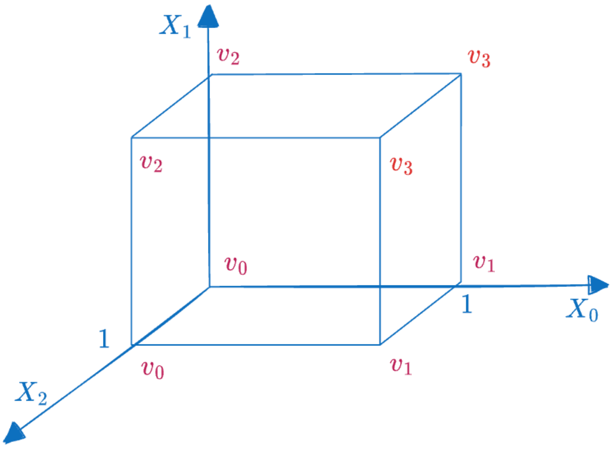
$$[[X_2 \cdot \tilde{c}]]_3 = X^4 \cdot [[\tilde{c}]]_2 \tag{20}$$

Quick derivation as follows:

$$[[X_2 \cdot \tilde{c}]]_3 = v_0X^4 + v_1X^5 + v_2X^6 + v_3X^7 = X^4 \cdot (v_0 + v_1X + v_2X^2 + v_3X^3) = X^4 \cdot [[\tilde{c}]]_2 \tag{21}$$

Here, X^4 raises the degree of $[[\tilde{c}]]_2$, making it fit perfectly in the high-bit region of the 3-dimensional HyperCube (i.e., the region where $X_2 = 1$).

Next, let's consider the values of $\tilde{c}$ on a three-dimensional HyperCube. We'll find that regardless of whether the newly added variable X_2 is 0 or 1, the polynomial's value only depends on X_0, X_1. Therefore, its point value form is equal to the two-dimensional point value vector copied once, filling up the 3-dimensional HyperCube, as shown in the following figure:



$\tilde{c}(X_0, X_1)$

In other words, the point value form of $\tilde{c}$ on a three-dimensional HyperCube is $(v_0, v_1, v_2, v_3, v_0, v_1, v_2, v_3)$, so the Univariate polynomial it maps to is:

$$\begin{aligned} [[\tilde{c}]]_3 &= v_0 + v_1X + v_2X^2 + v_3X^3 + v_0X^4 + v_1X^5 + v_2X^6 + v_3X^7 \\ &= (1 + X^4) \cdot (v_0 + v_1X + v_2X^2 + v_3X^3) \\ &= (1 + X^4) \cdot [[\tilde{c}]]_2 \end{aligned} \tag{22}$$

The above equation can be explained as follows: the values on the three-dimensional HyperCube are composed of two parts, $[[\tilde{c}]]_2$ and $[[\tilde{c}]]_2$ with its degree raised by X^4.

Similarly, the values of $\tilde{c}$ on a four-dimensional HyperCube are $(v_0, v_1, v_2, v_3, \quad v_0, v_1, v_2, v_3, \quad v_0, v_1, v_2, v_3, \quad v_0, v_1, v_2, v_3)$, so the Univariate polynomial it maps to is:

$$\begin{aligned} [[\tilde{c}]]_4 &= v_0 + v_1X + v_2X^2 + v_3X^3 + v_0X^4 + v_1X^5 + v_2X^6 + v_3X^7 \\ &= (1 + X^4 + X^8 + X^{12}) \cdot (v_0 + v_1X + v_2X^2 + v_3X^3) \\ &= (1 + X^4 + X^8 + X^{12}) \cdot [[\tilde{c}]]_2 \end{aligned} \tag{23}$$

When raising a low-dimensional MLE to a high-dimensional HyperCube, we see the phenomenon of the low-dimensional HyperCube constantly copying itself. We can define a new polynomial function, $\Phi_k(X)$, to represent this repetitive operation:

$$\Phi_k(X^h) = 1 + X^h + X^{2h} + \ldots + X^{(2^k-1)h} \tag{24}$$

Therefore, we can define a general relation, i.e.

$$[[\tilde{c}]]_n = \Phi_{n-k}(X^{2^k}) \cdot [[\tilde{c}]]_k \tag{25}$$

Consequently, it can be verified that : $[[\tilde{c}]]_4 = \Phi_2(X^4) \cdot [[\tilde{c}]]_2$.

MLE Polynomial Remainder Theorem

> TODO: What's the correct name for this remainder theorem?

The next question is how to use this MLE to Univariate polynomial mapping to implement the MLE Evaluation Argument protocol. Specifically, the problem is how to use $\mathsf{cm}(\tilde{f})$ to verify the correctness of $\tilde{f}$'s value at a certain point, such as $\tilde{f}(u_0, u_1)$? Although we already have an Evaluation Argument protocol based on KZG10, unfortunately it's based on Univariate polynomials, not MLE polynomials. KZG10 uses the polynomial remainder theorem, as in the following formula:

$$\hat{f}(X) - \hat{f}(z) = q(X) \cdot (X - z) \tag{26}$$

It uses the commitment $\mathsf{cm}(q)$ of the quotient polynomial $q(X)$ as the proof for the Evaluation Argument. So how do we transform the problem of proving MLE's evaluation at a multi-dimensional point, such as $(u_0, u_1, \ldots, u_{n-1})$, into proving the evaluation of a Univariate polynomial at one or more points?

The paper [PST13] gives a multivariate polynomial version of the above theorem:

$$f(X_0, X_1, \ldots, X_{n-1}) - f(u_0, u_1, \ldots, u_{n-1}) = \sum_{k=0}^{n-1} q_k(X_0, X_1, \ldots, X_{n-1}) \cdot (X_k - u_k) \tag{27}$$

If $f(X_0, X_1, \ldots, X_{n-1})$ is an MLE polynomial, it can be simplified to the following formula:

$$\begin{aligned} \tilde{f}(X_0, X_1, \ldots, X_{n-1}) - \tilde{f}(u_0, u_1, \ldots, u_{n-1}) &= \tilde{q}_{n-1}(X_0, X_1, \ldots, X_{n-2}) \cdot (X_{n-1} - u_{n-1}) \\ &+ \tilde{q}_{n-2}(X_0, X_1, \ldots, X_{n-3}) \cdot (X_{n-2} - u_{n-2}) \\ &+ \cdots \\ &+ \tilde{q}_1(X_0) \cdot (X_1 - u_1) \\ &+ \tilde{q}_0 \cdot (X_0 - u_0) \end{aligned} \tag{28}$$

This is because in the MLE polynomial $f(X_0, X_1, \ldots, X_{n-1})$, the highest degree of each variable is 1. For $f(X_0, X_1, \ldots, X_k)$, after dividing by the factor $(X_k - u_k)$, the remainder polynomial will no longer contain the variable X_k. So when $f(X_0, X_1, \ldots, X_{n-1})$ is divided by factors $(X_{n-1} - u_{n-1})$ to $(X_0 - u_0)$ in sequence, the number of variables in the quotient polynomials and remainder polynomials keeps decreasing one by one, until we finally get a constant quotient polynomial $\tilde{q}_0$, and of course a constant remainder polynomial, which is exactly the evaluation of the MLE polynomial at $(u_0, u_1, \ldots, u_{n-1})$.

Let's assume this final evaluation is v, i.e.,

$$\tilde{f}(u_0, u_1, \ldots, u_{n-1}) = v \tag{29}$$

Then we apply the Zeromorph mapping to both sides of the remainder theorem equation (both viewed as MLE polynomials) to obtain the corresponding Univariate polynomials.

$$[[\tilde{f}(X_0, X_1, \ldots, X_{n-1}) - v]]_n = [[\sum_{k=0}^{n-1} \tilde{q}_k(X_0, X_1, \ldots, X_{k-1}) \cdot (X_k - u_k)]]_n \tag{30}$$

Due to the additive homomorphism of the mapping, we can continue to simplify the above equation:

$$\begin{aligned} [[\tilde{f}(X_0, X_1, \ldots, X_{n-1})]]_n - [[v]]_n &= \sum_{k=0}^{n-1} [[\tilde{q}_k(X_0, X_1, \ldots, X_{k-1}) \cdot (X_k - u_k)]]_n \\ &= \sum_{k=0}^{n-1} \Big([[X_k \cdot \tilde{q}_k(X_0, X_1, \ldots, X_{k-1})]]_n - u_k [[\tilde{q}_k(X_0, X_1, \ldots, X_{k-1})]]_n \Big) \end{aligned} \tag{31}$$

First, look at the term $[[\tilde{f}(X_0, X_1, \ldots, X_{n-1})]]_n$ on the left side of the equation, which directly maps to $\hat{f}(X)$. Then look at the term $[[v]]_n$, which maps to $\hat{v}(X)$,

$$[[v]]_n = \hat{v}(X) = v + vX + vX^2 + \ldots + vX^{n-1} \tag{32}$$

Or we can use the $\Phi_n(X)$ function to represent it:

$$[[v]]_n = v \cdot \Phi_n(X) \tag{33}$$

Looking at the term $[[\tilde{q}_k(X_0, X_1, \ldots, X_{k-1})]]_n$ on the right side of the equation, this term is mapping a k-dimensional HyperCube to an n-dimensional HyperCube, and then performing the mapping. According to the previous discussion, we need to copy the k-dimensional HyperCube 2^{n-k} times consecutively to fill the n-dimensional HyperCube:

$$[[f(X_0, X_1, \ldots, X_{k-1})]]_n = \Phi_{n-k}(X^{2^k}) \cdot [[f(X_0, X_1, \ldots, X_{k-1})]]_k \tag{34}$$

To explain further, because $\Phi_{n-k}(X^{2^k})$ represents a coefficient vector with an interval of 2^k, its definition expands as follows:

$$\Phi_{n-k}(X^{2^k}) = 1 + X^{2^k} + X^{2 \cdot 2^k} + \ldots + X^{(2^{n-k}-1) \cdot 2^k} \tag{35}$$

Its coefficient vector is:

$$(1, 0, 0, \ldots, 0, \quad 1, 0, \ldots, 0, \quad 1, 0, \ldots, 0, \quad 1) \tag{36}$$

Suppose there is a degree-limited polynomial $g(X) \in \mathbb{F}_q[X]$, satisfying $\deg(g) < 2^k$, then the polynomial $\Phi_{n-k}(X^{2^k}) \cdot g$ represents a polynomial $g(X)$ of degree $2^k - 1$ repeated 2^{n-k} times with an interval of 2^k, ultimately resulting in a polynomial of degree $2^n - 1$.

Finally, there's the term $[[X_k \cdot \tilde{f}(X_0, X_1, \ldots, X_{k-1})]]_n$, how do we continue to simplify it?

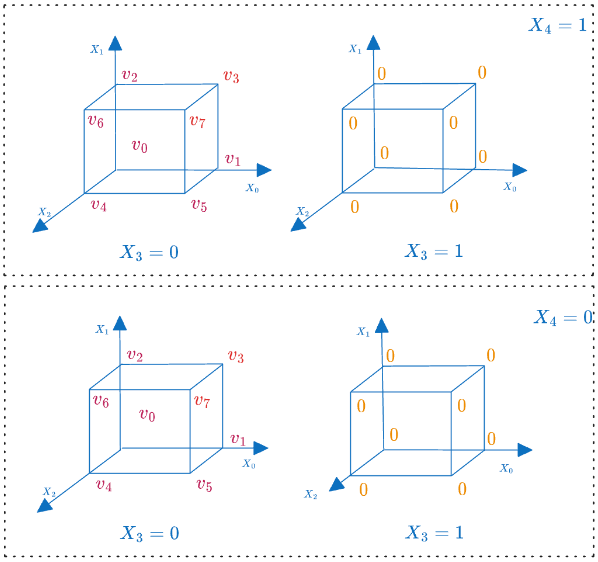
We can construct its mapping in two steps. First, look at $\tilde{f}(X_0, X_1, \ldots, X_{k-1})$ which can be represented by a k-dimensional Hypercube, then when multiplied by a new variable X_k, it becomes a $(k+1)$-dimensional HyperCube. This new Hypercube can be divided into two parts, one part is all zeros (when $X_k = 0$), and the other part is exactly $\tilde{f}(X_0, X_1, \ldots, X_{k-1})$. So we first use the $\Phi_n(X)$ function to construct a repetition pattern of HyperCube with an interval of 2^{k+1}, then repeat the k-dimensional HyperCube 2^{n-k-1} times, so we get the following polynomial.

$$\Phi_{n-k-1}(X^{2^{k+1}}) \cdot [[\tilde{f}(X_0, X_1, \ldots, X_{k-1})]]_k \tag{37}$$

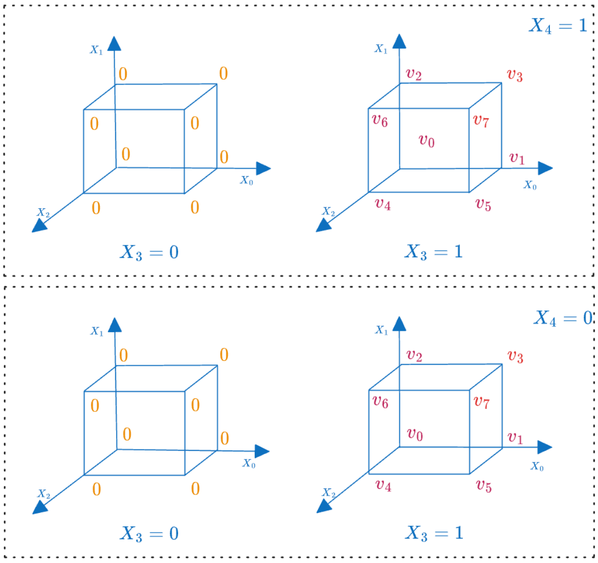
However, this is only the first step. The above Univariate polynomial is not equal to $[[X_k \cdot \tilde{f}(X_0, X_1, \ldots, X_{k-1})]]_n$, because in each repeated $(k+1)$-dimensional HyperCube of the former, the part where $X_k = 1$ is zero, while the part where $X_k = 0$ is filled with the k-dimensional HyberCube $\tilde{f}(X_0, X_1, \ldots, X_{k-1})$, which is opposite to the HyperCube we want. We need to add a shift factor like X^{2^k} to it, so that we can swap the position of the k-dimensional HyperCube corresponding to X_k (from the low-bit region to the high-bit region):

$$[[X_k \cdot \tilde{f}(X_0, X_1, \ldots, X_{k-1})]]_n = X^{2^k} \cdot \Phi_{n-k-1}(X^{2^{k+1}}) \cdot [[\tilde{f}(X_0, X_1, \ldots, X_{k-1})]]_k \tag{38}$$

The following figure demonstrates with a specific example where $k = 3, n = 5$. The left side is the 5-dimensional HyperCube before shifting, where the upper and lower half fields represent the fifth dimension, and each half field has two three-dimensional cubes representing the fourth dimension. We can see that only the three-dimensional cube when $X_3 = 0$ corresponds to $\tilde{f}(X_0, X_1, X_2)$, while when $X_3 = 1$, the three-dimensional cube is all zero. The right side of the figure below is the 5-dimensional HyperCube after shifting, where the $\tilde{f}(X_0, X_1, X_2)$ cube has been shifted to the right, that is, to the region corresponding to $X_3 = 1$.



$\Phi_2(X^{2^3}) \cdot [[\tilde{f}(X_0, X_1, X_2)]]_3$



$X^{2^3} \cdot \Phi_2(X^{2^3}) \cdot [[\tilde{f}(X_0, X_1, X_2)]]_3$

At this point, we can obtain the key equation of the Zeromorph protocol:

$$[[\tilde{f}(X_0, X_1, \ldots, X_{n-1})]]_n - v \cdot \Phi_n(X) = \sum_{k=0}^{n-1} \left(X^{2^k} \cdot \Phi_{n-k-1}(X^{2^{k+1}}) - u_k \cdot \Phi_{n-k}(X^{2^k}) \right) \cdot [[\tilde{q}_k(X_0, X_1, \ldots, X_{k-1})]]_k \tag{39}$$

KZG10-based Evaluation Argument

Note that the Zeromorph equation we derived in the previous section is an equation about Univariate polynomials. We can write it briefly as:

$$\hat{f}(X) - v \cdot \Phi_n(X) = \sum_k \left(X^{2^k} \cdot \Phi_{n-k-1}(X^{2^{k+1}}) - u_k \cdot \Phi_{n-k}(X^{2^k}) \right) \cdot \hat{q}_k(X) \tag{40}$$

Here $\hat{f}(X)$ and $\hat{q}_k(X)$ are defined as follows:

$$\begin{aligned}\hat{f}(X) &= [[\tilde{f}(X_0, X_1, \ldots, X_{n-1})]]_n \\ \hat{q}_k(X) &= [[\tilde{q}_k(X_0, X_1, \ldots, X_{k-1})]]_k\end{aligned} \tag{41}$$

To prove that the value of $\tilde{f}(X_0, X_1, \ldots, X_{n-1})$ at the point $(u_0, u_1, \ldots, u_{n-1})$ is v, we only need to check if the above polynomials are equal. Here, we use the idea of the Schwartz-Zippel lemma: let the Verifier randomly choose a point $X = \zeta$, and then let the Prover provide the values of $\hat{f}(\zeta)$ and $\hat{q}_k(\zeta)$, so that the Verifier can verify whether the following equation holds:

$$\hat{f}(\zeta) - v \cdot \Phi_n(\zeta) = \sum_k \Big(\zeta^{2^k} \cdot \Phi_{n-k-1}(\zeta^{2^{k+1}}) - u_k \cdot \Phi_{n-k}(\zeta^{2^k})\Big) \cdot \hat{q}_k(\zeta) \tag{42}$$

However, this is not enough, because what the Prover actually commits to is $\hat{q}_k(X)$. To ensure that the MLE remainder polynomial relation holds, we must enforce that the degrees of all quotient polynomials $\hat{q}_k(X)$ are less than 2^k, i.e., $\deg(\hat{q}_k) < 2^k$, to ensure that the Prover has no room for cheating.

Both FRI and KZG10 provide methods to prove $\deg(\hat{q}_k) < 2^k$. In this article, we only consider the Zeromorph protocol based on KZG10. A simple Degree Bound proof protocol based on KZG10 is as follows:

- The Prover provides $\mathsf{cm}(\hat{q}_k)$ and additionally $\mathsf{cm}(X^{D-2^k+1} \cdot \hat{q}_k(X))$ and sends them to the Verifier,
- The Verifier verifies the following equation:

$$e\Big(\mathsf{cm}(\hat{q}_k),\ [\tau^{D-2^k+1}]_2\Big) = e\Big(\mathsf{cm}(X^{D-2^k+1} \cdot \hat{q}_k(X)),\ [1]_2\Big) \tag{43}$$

Here, the role of $X^{D-2^k+1} \cdot \hat{q}_k(X)$ is to align the Degree of $\hat{q}_k(X)$ to D. Because in the KZG10 SRS, the maximum Degree of polynomials that can be committed is D, so if the Degree of $\hat{q}_k(X)$ exceeds 2^k, then $\deg(X^{D-2^k+1} \cdot \hat{q}) > D$, making it impossible to commit using the KZG10 SRS. Conversely, if the Prover can correctly commit to $X^{D-2^k+1} \cdot \hat{q}_k(X)$, it proves that $\deg(\hat{q}_k) < 2^k$.

Protocol Description

Below, we first give a simple and naive protocol implementation for easy understanding.

Public Input

- Commitment of MLE polynomial $\tilde{f}$: $\mathsf{cm}([[\tilde{f}]]_n)$
- Evaluation point $\mathbf{u} = (u_0, u_1, \ldots, u_{n-1})$
- Evaluation result $v = \tilde{f}(\mathbf{u})$

Witness

- Point value vector of MLE polynomial $\tilde{f}$ on n-dimensional HyperCube $\mathbf{a} = (a_0, a_1, \ldots, a_{2^n-1})$

Interactive Process

Round 1: Prover sends commitments of remainder polynomials

- Calculate n remainder MLE polynomials, $\{\tilde{q}_k\}_{k=0}^{n-1}$
- Construct Univariate polynomials mapped from remainder MLE polynomials $Q_k = [[\tilde{q}_k]]_k, \quad 0 \le k < n$
- Calculate and send their commitments: $\mathsf{cm}(Q_0), \mathsf{cm}(Q_1), \ldots, \mathsf{cm}(Q_{n-1})$

$$\tilde{f}(X_0, X_1, \ldots, X_{n-1}) - v = \sum_{k=0}^{n-1} (X_k - u_k) \cdot \tilde{q}_k(X_0, X_1, \ldots, X_{k-1}) \tag{44}$$

Round 2: Prover calculates, $\pi_k = \mathsf{cm}(X^{D_{max}-2^k} \cdot Q_k), \quad 0 \le k < n$, as the Degree Bound proof of $\deg(Q_k) < 2^k$, and sends them to the Verifier

Round 3: Verifier sends a random number $\zeta \in \mathbb{F}_p^*$

Round 4: Prover calculates auxiliary polynomial $R(X)$ and quotient polynomial $H(X)$, and sends $\mathsf{cm}(H)$

- Calculate $R(X)$,

$$R(X) = F(X) - v \cdot \Phi_n(\zeta) - \sum_{k=0}^{n-1} \Big(\zeta^{2^k} \cdot \Phi_{n-k-1}(\zeta^{2^{k+1}}) - u_k \cdot \Phi_{n-k}(\zeta^{2^k})\Big) \cdot Q_k(X) \tag{45}$$

- Calculate $H(X)$ and its commitment $\mathsf{cm}(H)$, as proof that $R(X)$ takes the value zero at $X=\zeta$

$$H(X)=\frac{R(X)}{X-\zeta} \tag{46}$$

Round 5: Verifier verifies the following equations

- Construct the commitment of $\mathsf{cm}(R)$:

$$\mathsf{cm}(R)=\mathsf{cm}(F)-\mathsf{cm}(v\cdot\Phi_n(\zeta))-\sum_{i=0}^{n-1}\Big(\zeta^{2^i}\cdot\Phi_{n-i-1}(\zeta^{2^{i+1}})-u_i\cdot\Phi_{n-i}(\zeta^{2^i})\Big)\cdot\mathsf{cm}(Q_i) \tag{47}$$

- Verify $R(\zeta)=0$

$$e(\mathsf{cm}(R),\ [1]_2)=e(\mathsf{cm}(H),[\tau]_2-\zeta\cdot[1]_2) \tag{48}$$

- Verify if $(\pi_0,\pi_1,\ldots,\pi_{n-1})$ are correct, i.e., verify the Degree Bound of all remainder polynomials: $\deg(Q_i)<2^i$, for $0\le i<n$

$$e(\mathsf{cm}(Q_i),[\tau^{D_{max}-2^i}]_2)=e(\pi_i,[1]_2),\quad 0\le i<n \tag{49}$$

Efficiency Overview

- Proof size: $(2n+1)\mathbb{G}_1$
- Verifier computation: $(2n+2)P$, $(n+2)\mathsf{EccMul}^{\mathbb{G}_1}$

Optimized Protocol

In the naive protocol, there are n quotient polynomials, and their Degree Bound proofs have $2n\ \mathbb{G}_1$ elements, which is obviously not efficient enough. However, we can prove these n degree bounds in batch. Here's the traditional batch proof approach:

- Verifier first sends a random number β
- Prover aggregates the n quotient polynomials together to get $P(X)$, and when aggregating, aligns the Degree of these quotient polynomials to the same value, which is the Degree of the largest quotient polynomial 2^{n-1}:

$$P(X)=\sum_{k=0}^{n-1}\beta^k\cdot X^{2^n-2^k}\cdot Q_i(X) \tag{50}$$

- Prover sends the commitment of $P(X)$, $\mathsf{cm}(P)$
- Verifier sends a random number ζ
- Prover constructs polynomial $S(X)$, which takes the value zero at $X=\zeta$, i.e., $S(\zeta)=0$

$$S(X)=P(X)-\sum_{k=0}^{n-1}\beta^k\cdot\zeta^{2^n-2^k}\cdot Q_i(X) \tag{51}$$

- Prover constructs quotient polynomial $H_1(X)$ and aligns its Degree to the maximum Degree Bound D, then proves $S(\zeta)=0$, and sends the commitment $\mathsf{cm}(H_1)$

$$H_1(X)=\frac{S(X)}{X-\zeta}\cdot X^{D-2^n+1} \tag{52}$$

- Verifier has $\mathsf{cm}(P)$ and $\mathsf{cm}(Q_i)$, he can restore the commitment of $\mathsf{cm}(S)$ based on the following equation:

$$\mathsf{cm}(S)=\mathsf{cm}(P)-\sum_{i=0}^{n-1}\beta^i\cdot\zeta^{2^n-2^k}\cdot Q_i(X) \tag{53}$$

- Verifier only needs two Pairing operations to verify $S(\zeta)=0$, thus obtaining the proof that n Degree Bounds hold

$$e\big(\mathsf{cm}(S),\ [\tau^{D_{max}-2^n+1}]_2\big)=e\big(\mathsf{cm}(H),[\tau]_2-\zeta\cdot[1]_2\big) \tag{54}$$

Moreover, Verifier can send a random number α to further aggregate the evaluation proofs of $R(X)$ and $S(X)$, because they both take the value zero at $X=\zeta$.

Below is the optimized version of the Zeromorph protocol, refer to Zeromorph paper [KT23] Section 6. The main optimization technique is to aggregate multiple Degree Bound proofs together, and also aggregate the evaluation proof of $R(X)$ together. This way, only two Pairing operations are needed for verification (this version does not consider the Zero-knowledge property for now).

Public Input

- Commitment of MLE polynomial $\tilde{f}$ mapped to Univariate polynomial $F(X) = [[\tilde{f}]]_n$: $\mathsf{cm}([[\tilde{f}]]_n)$
- Evaluation point $\mathbf{u} = (u_0, u_1, \ldots, u_{n-1})$
- Evaluation result $v = \tilde{f}(\mathbf{u})$

Witness

- Evaluation vector of MLE polynomial $\tilde{f}$: $\mathbf{a} = (a_0, a_1, \ldots, a_{2^n-1})$

Protocol

Round 1: Prover sends commitments of remainder polynomials

- Calculate n remainder MLE polynomials, $\{q_i\}_{i=0}^{n-1}$
- Construct Univariate polynomials mapped from remainder MLE polynomials $Q_i = [[q_i]]_i, \quad 0 \leq i < n$
- Calculate and send their commitments: $\mathsf{cm}(Q_0), \mathsf{cm}(Q_1), \ldots, \mathsf{cm}(Q_{n-1})$

$$\tilde{f}(X_0, X_1, \ldots, X_{n-1}) - v = \sum_{i=0}^{n-1}(X_k - u_k) \cdot q_i(X_0, X_1, \ldots, X_{k-1}) \tag{55}$$

Round 2: Verifier sends a random number $\beta \in \mathbb{F}_p^*$ to aggregate multiple Degree Bound proofs

Round 3: Prover calculates $P(X)$ and sends its commitment $\mathsf{cm}(P)$

- Calculate $P(X)$,

$$P(X) = \sum_{i=0}^{n-1} \beta^i \cdot X^{2^n-2^k} Q_i(X) \tag{56}$$

Round 4: Verifier sends a random number $\zeta \in \mathbb{F}_p^*$ to challenge the polynomial evaluation at $X = \zeta$

Round 5: Prover calculates $H_0(X)$ and $H_1(X)$

- Calculate $R(X)$,

$$R(X) = F(X) - v \cdot \Phi_n(\zeta) - \sum_{i=0}^{n-1} \left(\zeta^{2^i} \cdot \Phi_{n-i-1}(\zeta^{2^{i+1}}) - u_i \cdot \Phi_{n-i}(\zeta^{2^i})\right) \cdot Q_i(X) \tag{57}$$

- Calculate $S(X)$,

$$S(X) = P(X) - \sum_{k=0}^{n-1} \beta^k \cdot \zeta^{2^n-2^k} \cdot Q_i(X) \tag{58}$$

- Calculate quotient polynomials $H_0(X)$ and $H_1(X)$

$$H_0(X) = \frac{R(X)}{X-\zeta}, \qquad H_1(X) = \frac{S(X)}{X-\zeta} \tag{59}$$

Round 6: Verifier sends a random number $\alpha \in \mathbb{F}_p^*$ to aggregate $H_0(X)$ and $H_1(X)$

Round 7: Prover calculates $H(X)$ and sends its commitment $\mathsf{cm}(H)$

- Calculate $H(X) = (H_0(X) + \alpha \cdot H_1(X)) \cdot X^{D_{max}-2^n+1}$

Round 8: Verifier verifies the following equations

- Restore the commitment of $\mathsf{cm}(R)$:

$$\mathsf{cm}(R) = \mathsf{cm}(F) - \mathsf{cm}(v \cdot \Phi_n(\zeta)) - \sum_{i=0}^{n-1} \left(\zeta^{2^i} \cdot \Phi_{n-i-1}(\zeta^{2^{i+1}}) - u_i \cdot \Phi_{n-i}(\zeta^{2^i})\right) \cdot \mathsf{cm}(Q_i) \tag{60}$$

- Restore the commitment of $\mathsf{cm}(S)$:

$$\mathsf{cm}(S) = \mathsf{cm}(P) - \sum_{i=0}^{n-1} \beta^i \cdot \zeta^{2^n-2^k} \cdot Q_i(X)) \tag{61}$$

- Verify $R(\zeta) = 0$ and $S(\zeta) = 0$

$$e(\mathsf{cm}(R) + \alpha \cdot \mathsf{cm}(S),\ [\tau^{D-2^n+1}]_2) = e(\mathsf{cm}(H), [\tau]_2 - \zeta \cdot [1]_2) \tag{62}$$

Summary

Overall, Zeromorph is a concise protocol. It directly maps the point value form of MLE to the coefficients of Univariate polynomials, and then uses the KZG10 protocol to complete the Evaluation proof. Subsequent articles will discuss how to combine Zeromorph with the FRI protocol to implement MLE PCS.

Reference:

- [KT23] Kohrita, Tohru, and Patrick Towa. "Zeromorph: Zero-knowledge multilinear-evaluation proofs from homomorphic univariate commitments." Cryptology ePrint Archive (2023). https://eprint.iacr.org/2023/917
- [PST13] Papamanthou, Charalampos, Elaine Shi, and Roberto Tamassia. "Signatures of correct computation." Theory of Cryptography Conference. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. https://eprint.iacr.org/2011/587