Fast interpolation, evaluation and change of basis for polynomials over finite fields of characteristic two

Nicholas Coxon

September 24, 2018

SpecFun Seminar

Table of contents

1. Fast transforms over finite fields of characteristic two

2. Fast Hermite interpolation and evaluation over finite fields of characteristic two

Fast transforms over finite fields of characteristic two

Outline

Let $\mathbb{F}$ be a finite field of characteristic two.

Let $\mathbb{F}[x]_\ell$ denote the space of polynomials in $\mathbb{F}[x]$ of degree less than ℓ.

Here's what we're going to do:

1. To $\beta \in \mathbb{F}^n$ with linearly independent entries over $\mathbb{F}_2$, we associate four bases of $\mathbb{F}[x]_{2^n}$.
2. Describe fast algorithms for converting between the bases.
3. Show how to choose β in order to make the algorithms faster.

Subspace enumeration

Define maps $[\,\cdot\,]_k : \mathbb{N} \to \{0,1\}$ for $k \in \mathbb{N}$ by

$$i = \sum_{k \in \mathbb{N}} 2^k [i]_k \quad \text{for } i \in \mathbb{N}.$$

If the entries of $\beta = (\beta_0, \ldots, \beta_{n-1}) \in \mathbb{F}^n$ are linearly independent over $\mathbb{F}_2$, we enumerate the subspace they generate as $\{\omega_{\beta,0}, \ldots, \omega_{\beta,2^n-1}\}$, where

$$\omega_{\beta,i} = \sum_{k=0}^{n-1} [i]_k \beta_k \quad \text{for } i \in \{0, \ldots, 2^n - 1\}.$$

Polynomials bases

The associated Lagrange basis $\{L_{\beta,0}, \ldots, L_{\beta,2^n-1}\}$ of $\mathbb{F}[x]_{2^n}$ is given by

$$L_{\beta,i} = \prod_{\substack{j=0 \\ j \neq i}}^{2^n-1} \frac{x - \omega_{\beta,j}}{\omega_{\beta,i} - \omega_{\beta,j}} \quad \text{for } i \in \{0, \ldots, 2^n - 1\}.$$

The associated Newton basis $\{N_{\beta,0}, \ldots, N_{\beta,2^n-1}\}$ of $\mathbb{F}[x]_{2^n}$ is given by

$$N_{\beta,i} = \prod_{j=0}^{i-1} \frac{x - \omega_{\beta,j}}{\omega_{\beta,i} - \omega_{\beta,j}} \quad \text{for } i \in \{0, \ldots, 2^n - 1\}.$$

To make our life easier, we have normalised the Newton basis so that $N_{\beta,i}(\omega_{\beta,i}) = 1$ for $i \in \{0, \ldots, 2^n - 1\}$.

Polynomials bases

We also consider the basis $\{X_{\beta,0}, \ldots, X_{\beta,2^n-1}\}$ of $\mathbb{F}[x]_{2^n}$ given by

$$X_{\beta,i} = \prod_{k=0}^{n-1} \prod_{j=0}^{2^k[i]_k-1} \frac{x - \omega_{\beta,j}}{\omega_{\beta,2^k} - \omega_{\beta,j}} \quad \text{for } i \in \{0, \ldots, 2^n - 1\}.$$

(Note that $\deg X_{\beta,i} = \sum_{k=0}^{n-1} 2^k[i]_k = i$)

This basis was introduced by Lin, Chung and Han (2014). Consequently, we refer to it as the "LCH basis".

Finally, we also consider the monomial basis $\{1, x, \ldots, x^{2^n-1}\}$ of $\mathbb{F}[x]_{2^n}$.

Conversion problems

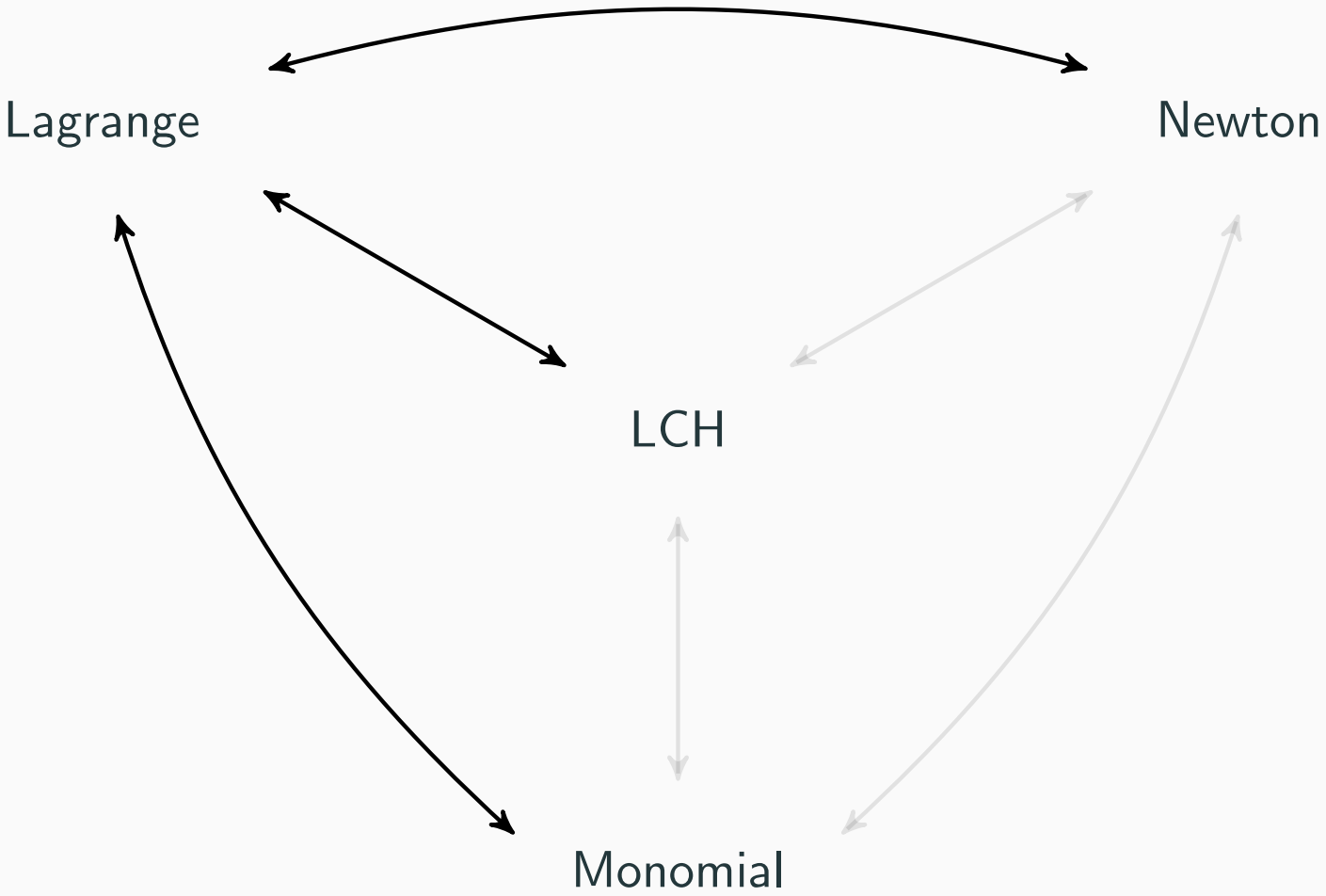
For each pairs of bases associated with a vector $\beta \in \mathbb{F}^n$, we consider the following problem:

Problem

Given the first ℓ coefficients on one of the bases of some polynomial in $\mathbb{F}[x]_\ell$, $\ell \in \{1, \ldots, 2^n\}$, compute its first ℓ coefficients on the other basis.

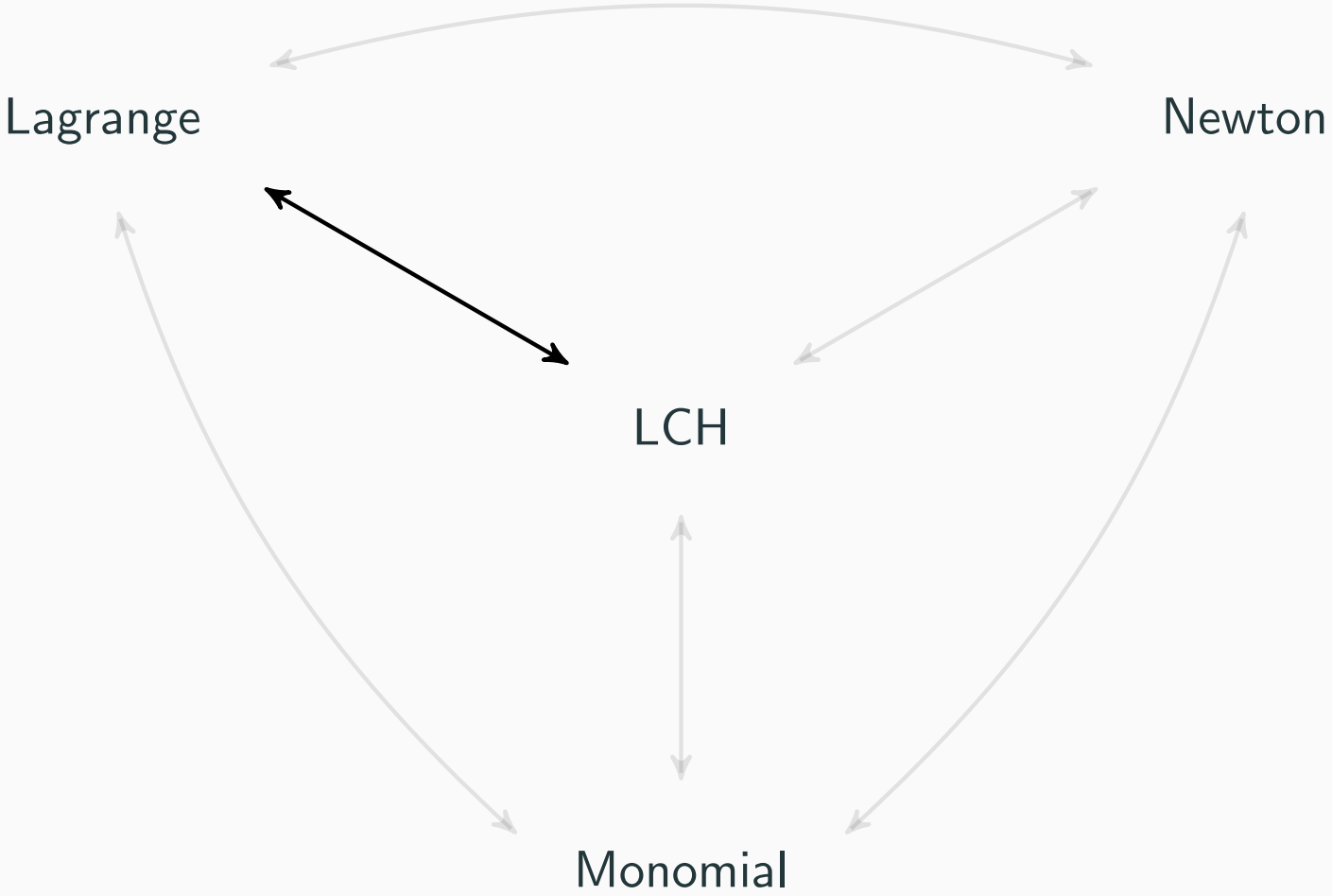
We call ℓ the length of an instance.

Applications



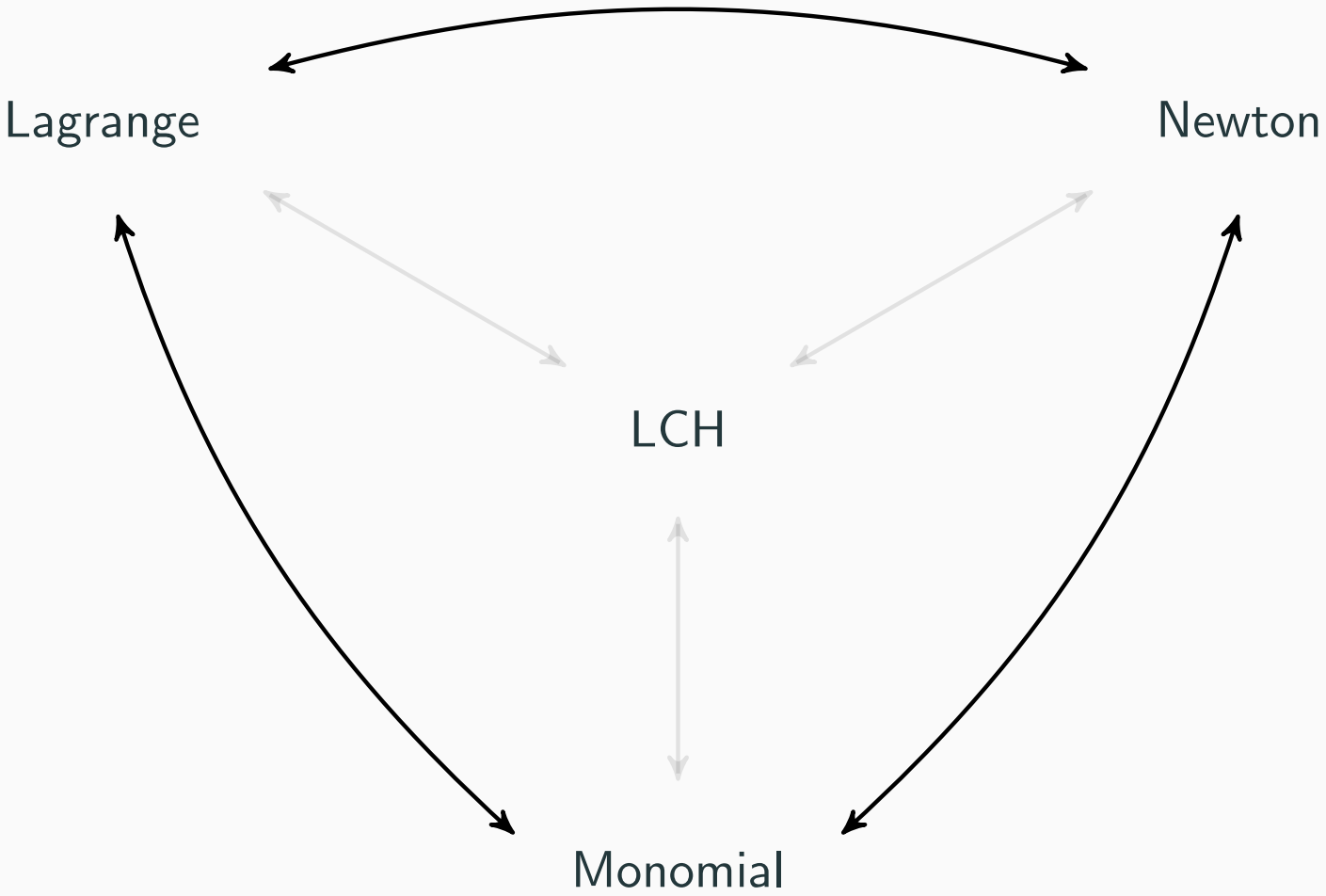
Multipoint evaluation and interpolation; ~~polynomial multiplication~~.

Applications



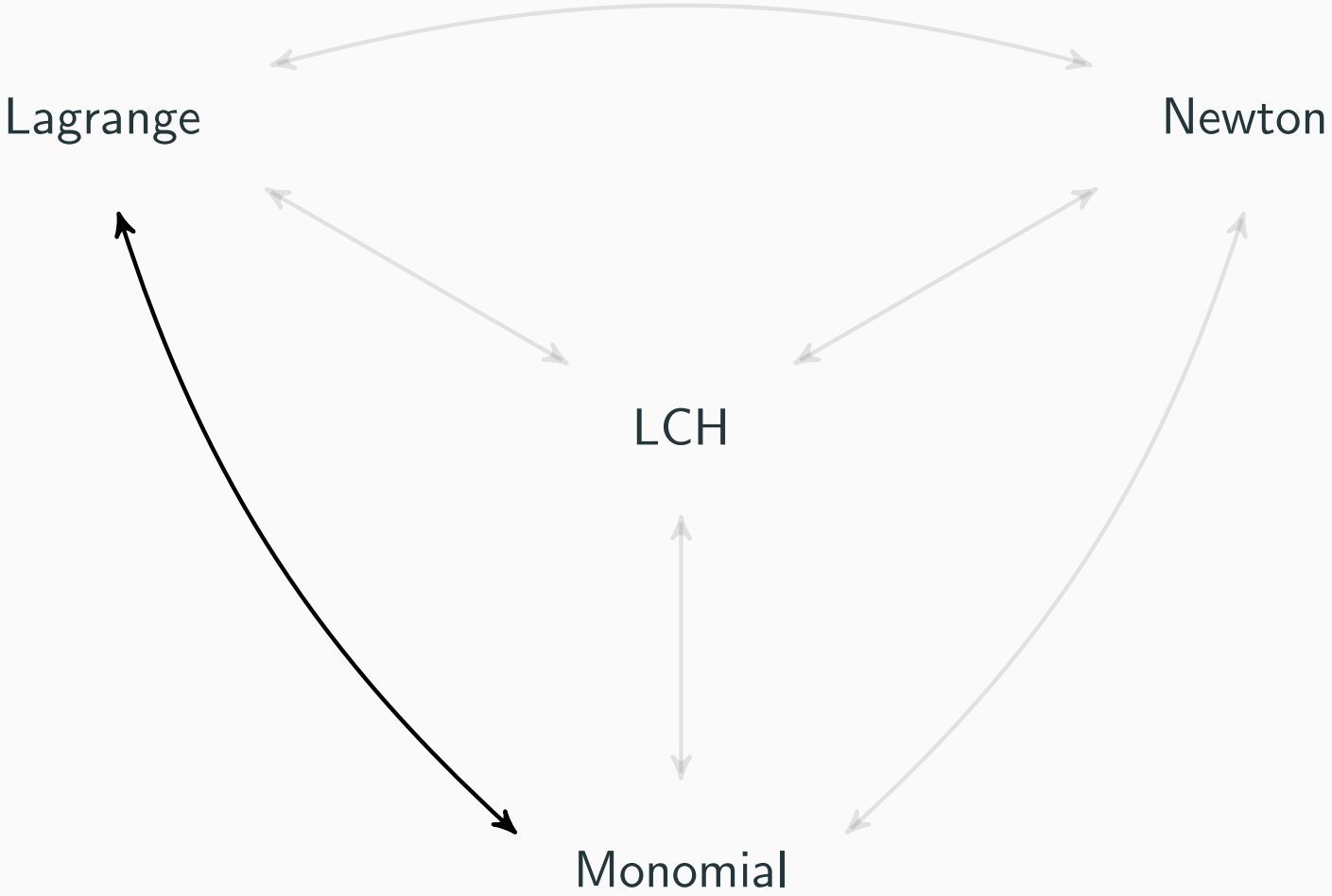
Reed–Solomon code encoding and decoding; zk-STARK.

Applications



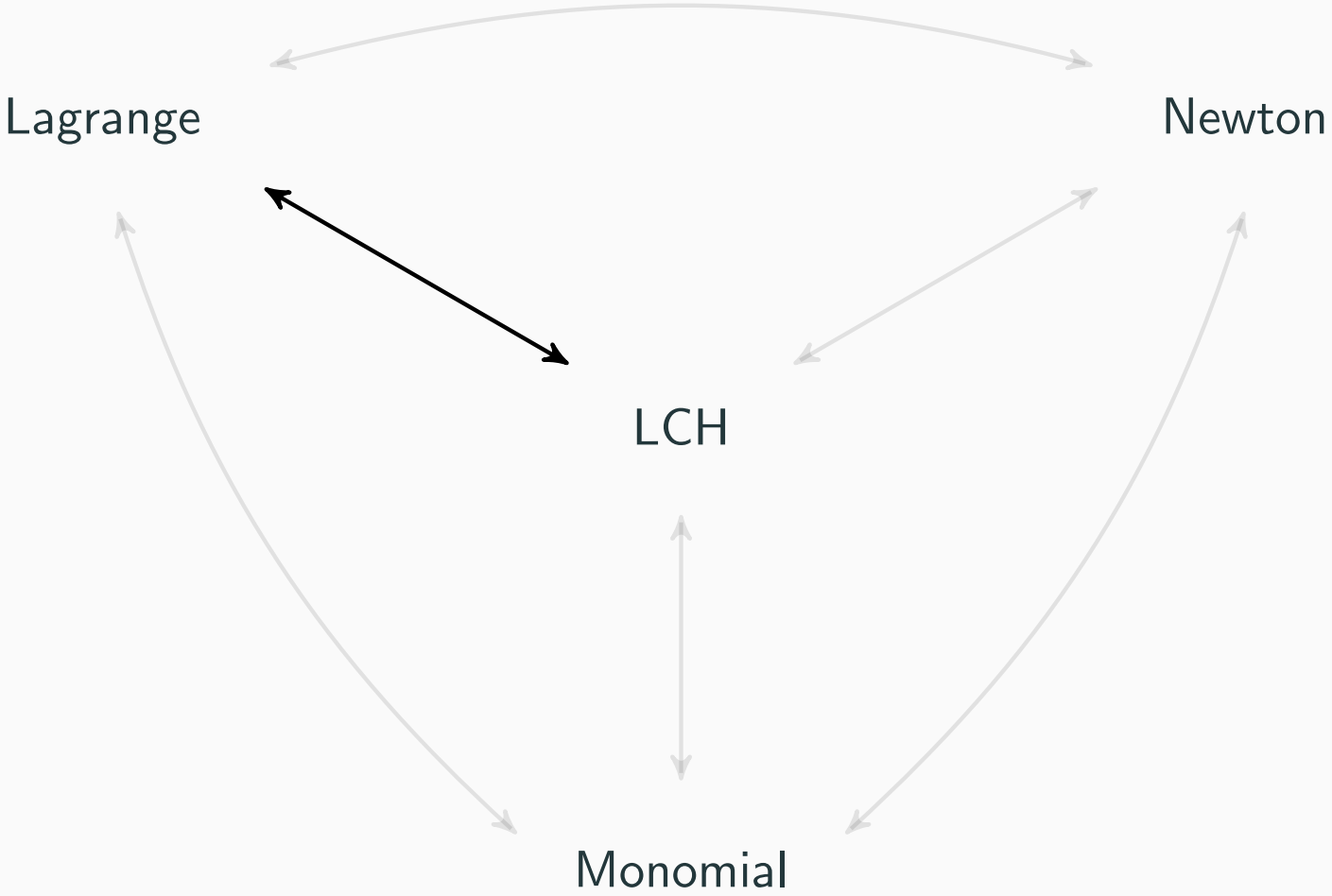
Certain multivariate (Hermite) interpolation and evaluation problems; encoding of Reed–Muller and multiplicity codes.

Previous work



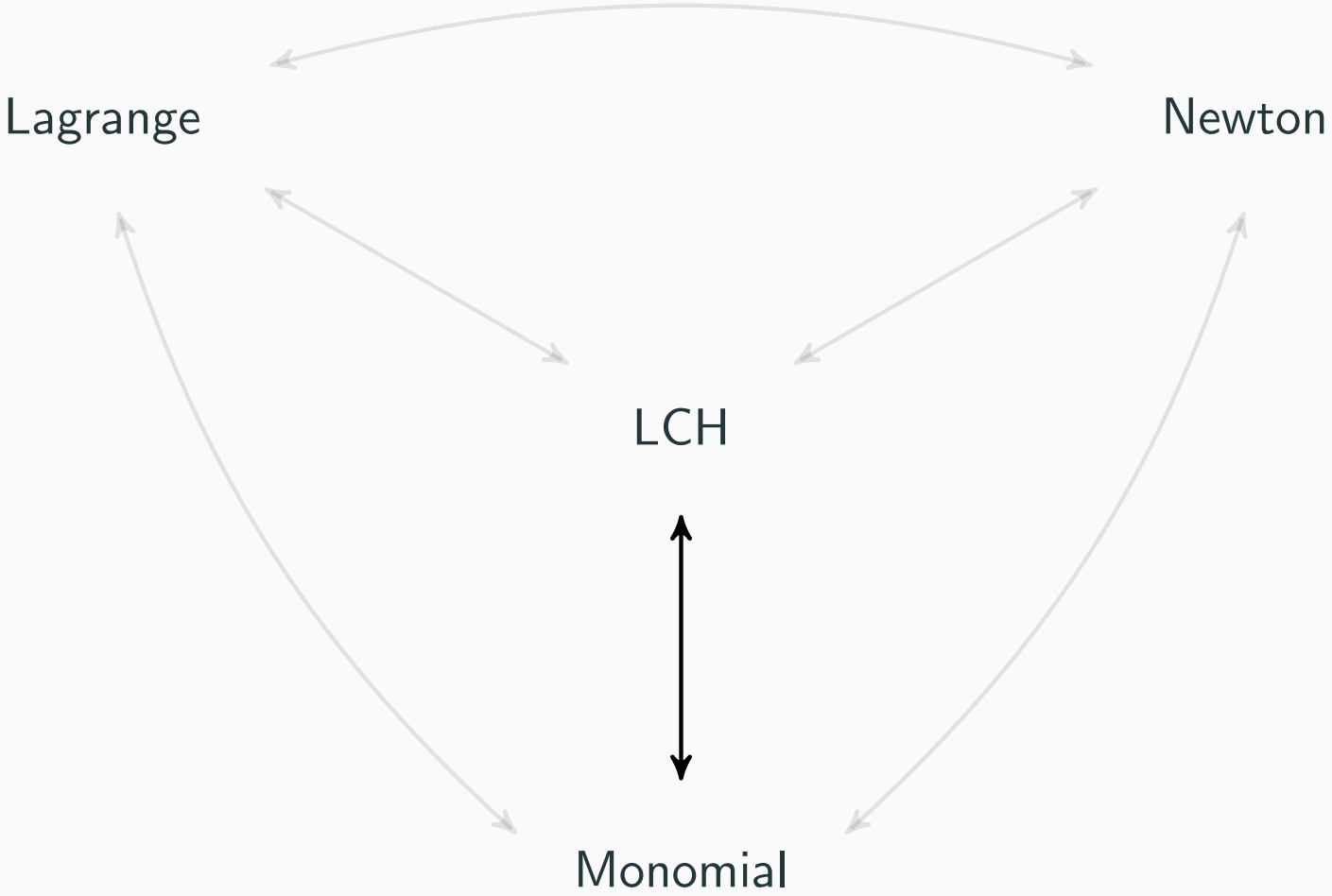
Gao and Mateer 2010 ($\ell = 2^n$): $\mathcal{O}(\ell \log \ell)$ multiplications and $\mathcal{O}(\ell \log^2 \ell)$ additions; only $\mathcal{O}(\ell \log \ell \log\log \ell)$ additions if β is a "Cantor basis".

Previous work



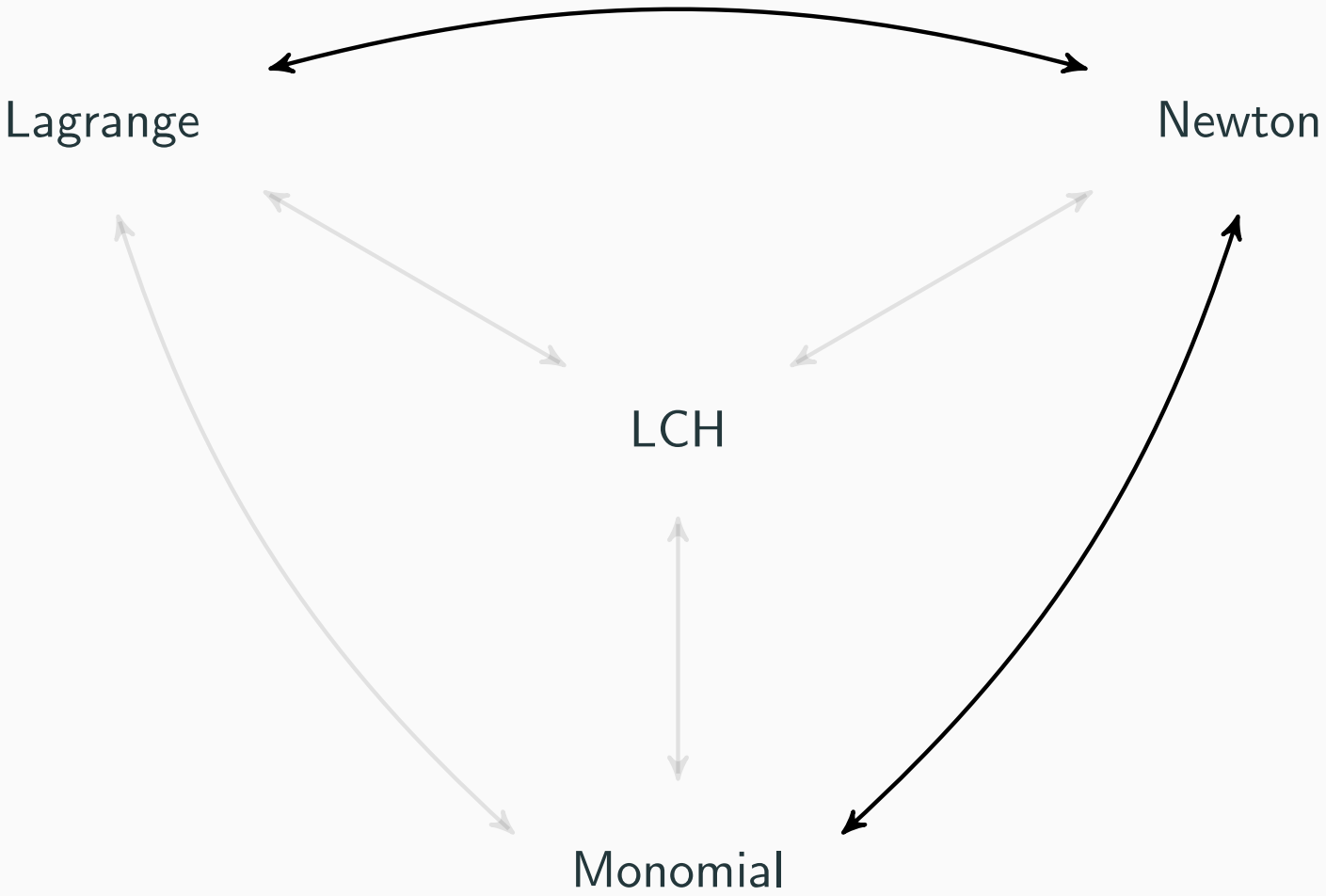
Lin, Chung and Han 2014 ($\ell = 2^n$): $\mathcal{O}(\ell \log \ell)$ multiplications/additions.

Previous work



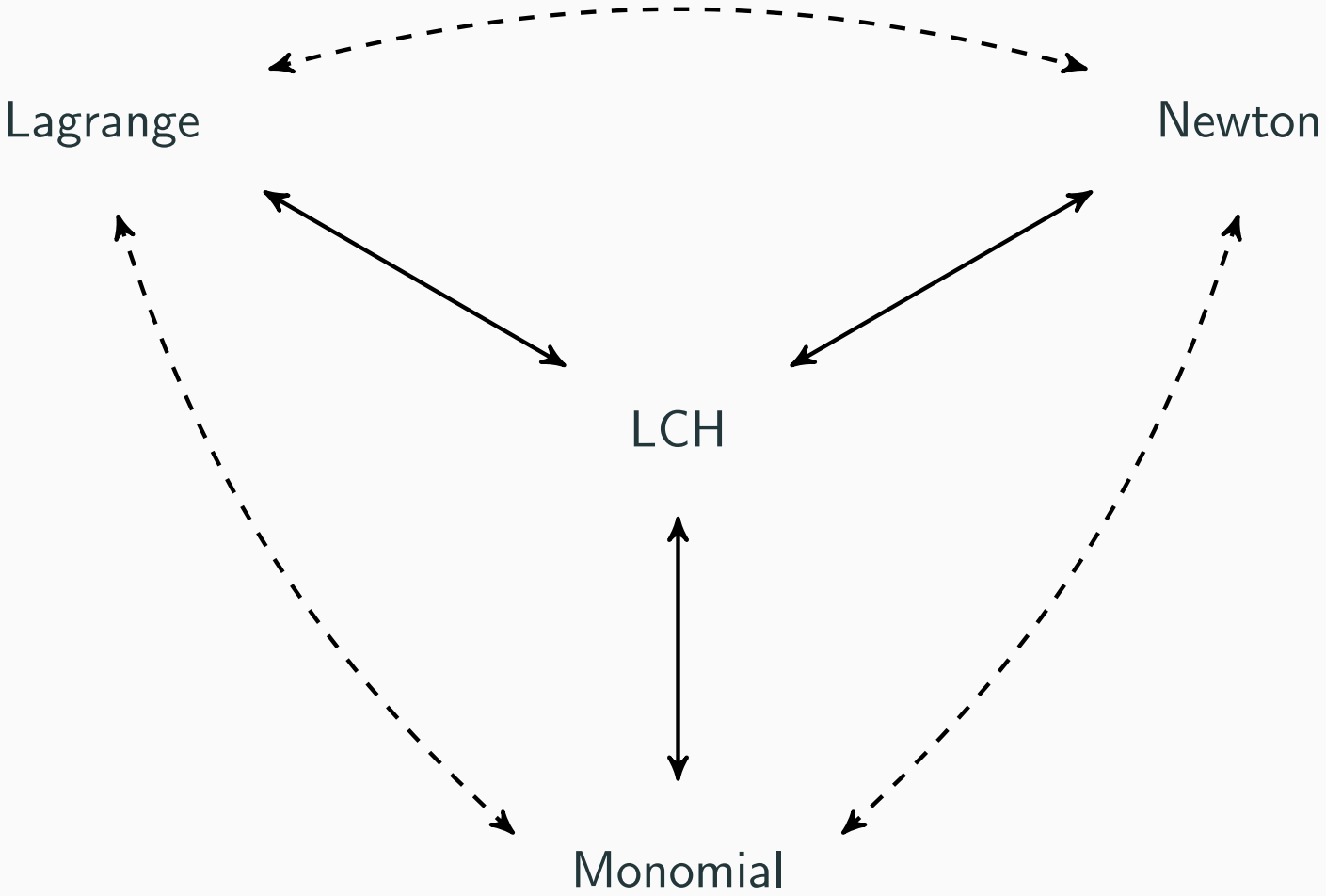
Lin et al. 2016 ($\ell = 2^n$): $\mathcal{O}(\ell \log \ell)$ multiplications and $\mathcal{O}(\ell \log^2 \ell)$ additions; only $\mathcal{O}(\ell \log \ell \log\log \ell)$ additions if β is a Cantor basis.

Previous work



Probably $\mathcal{O}(\mathsf{M}(\ell)\log\ell)$ using product trees, transposition principle, etc.

This work



Let us fix an arbitrary vector $\beta = (\beta_0, \ldots, \beta_{n-1}) \in \mathbb{F}^n$ that has linearly independent entries over $\mathbb{F}_2$.

If $n = 1$, then

$$N_{\beta,0} = X_{\beta,0} = 1, \quad N_{\beta,1} = X_{\beta,1} = \frac{x}{\beta_0}, \quad L_{\beta,0} = \frac{x}{\beta_0} + 1, \quad L_{\beta,1} = \frac{x}{\beta_0}.$$

It is straightforward to perform the conversions in this case.

Subspace vanishing polynomials

Define

$$S_{\beta,k} = \prod_{i=0}^{2^k-1} x - \omega_{\beta,i} \quad \text{for } k \in \{0, \ldots, n\}.$$

Properties

- $S_{\beta,k}$ vanishes on the subspace generated by $\beta_0, \ldots, \beta_{k-1}$.
- $S_{\beta,0} = x$ and $S_{\beta,k} = S^2_{\beta,k-1} - S_{\beta,k-1}(\beta_{k-1})S_{\beta,k-1}$ for $k \geq 1$.
- $S_{\beta,k} = x^{2^k} + \sum_{i=0}^{k-1} s_{k,i} x^{2^i}$ with $s_{k,0}, \ldots, s_{k,k-1} \in \mathbb{F}$.
- $S_{\beta,k}(x + \omega) = S_{\beta,k}(x) + S_{\beta,k}(\omega)$ for $\omega \in \mathbb{F}$.
- $X_{\beta,i} = \prod_{k=0}^{n-1} (S_{\beta,k}(x)/S_{\beta,k}(\beta_k))^{[i]_k}$ for $i \in \{0, \ldots, 2^n - 1\}$.

Main lemma

Lemma

Suppose that $n \geq 2$. For some $k \in \{1, \ldots, n-1\}$, let $\alpha = (\beta_0, \ldots, \beta_{k-1})$, $\gamma = (\beta_k, \ldots, \beta_{n-1})$ and $\delta = (S_{\beta,k}(\beta_k), \ldots, S_{\beta,k}(\beta_{n-1}))$. Then the entries of δ are linearly independent over $\mathbb{F}_2$, and

$$\begin{aligned} L_{\beta,2^k i+j} &= L_{\delta,i}\left(S_{\beta,k}(x)\right) L_{\alpha,j}\left(x - \omega_{\gamma,i}\right), \\ N_{\beta,2^k i+j} &= N_{\delta,i}\left(S_{\beta,k}(x)\right) N_{\alpha,j}\left(x - \omega_{\gamma,i}\right), \\ X_{\beta,2^k i+j} &= X_{\delta,i}\left(S_{\beta,k}(x)\right) X_{\alpha,j}\left(x\right) \end{aligned}$$

for $i \in \{0, \ldots, 2^{n-k} - 1\}$ and $j \in \{0, \ldots, 2^k - 1\}$.

Allows us to reduce conversion problems for the bases associated to β to conversion problems for the bases associated to α and δ.

But we need to introduce a shift parameter to allow recursion.

Conversion between the Newton and LCH bases

Corollary

Suppose that $f_0, \ldots, f_{\ell-1}, \lambda, h_0, \ldots, h_{\ell-1} \in \mathbb{F}$ *satisfy*

$$\sum_{i=0}^{\ell-1} f_i N_{\beta,i}(x-\lambda) = \sum_{i=0}^{\ell-1} h_i X_{\beta,i}.$$

Then there exist unique elements $g_0, \ldots, g_{\ell-1} \in \mathbb{F}$ *such that*

$$\sum_{j=0}^{\min(\ell-2^k i, 2^k)-1} f_{2^k i+j} N_{\alpha,j}(x-\lambda-\omega_{\gamma,i}) = \sum_{j=0}^{\min(\ell-2^k i, 2^k)-1} g_{2^k i+j} X_{\alpha,j}$$

for $i \in \{0, \ldots, \lceil \ell/2^k \rceil - 1\}$, *and*

$$\sum_{i=0}^{\lceil(\ell-j)/2^k\rceil-1} g_{2^k i+j} N_{\delta,i}(x-S_{\beta,k}(\lambda)) = \sum_{i=0}^{\lceil(\ell-j)/2^k\rceil-1} h_{2^k i+j} X_{\delta,i}$$

for $j \in \{0, \ldots, \min(2^k, \ell) - 1\}$.

Conversion between the Newton and LCH bases

How should we choose k?

- $k \approx n/2$ for "cache-friendliness",
- $k = n - 1$ if computing the shifts on the fly.

By always taking $k = n - 1$, we obtain the following complexity bounds.

Multiplications:	$\lfloor \ell/2 \rfloor \lceil \log_2 \ell \rceil$	$\lfloor \ell/2 \rfloor \lceil \log_2 \ell \rceil$
Additions:	$(\lfloor \ell/2 \rfloor - 1) \lceil \log_2 \ell \rceil + \ell - 1$	$\lfloor \ell/2 \rfloor \lceil \log_2 \ell \rceil$
Precomputations:	$\mathcal{O}(\log^2 \ell)$ operations	$\mathcal{O}(\ell)$ operations
Auxiliary space:	$\mathcal{O}(\log^2 \ell)$ field elements	$\mathcal{O}(\ell)$ field elements

Conversion between the Lagrange and LCH bases

Corollary ($\ell = 2^n$)

Suppose that $f_0, \ldots, f_{2^n-1}, \lambda, h_0, \ldots, h_{2^n-1} \in \mathbb{F}$ *satisfy*

$$\sum_{i=0}^{2^n-1} f_i L_{\beta,i}(x-\lambda) = \sum_{i=0}^{2^n-1} h_i X_{\beta,i}.$$

Then there exist unique elements $g_0, \ldots, g_{2^n-1} \in \mathbb{F}$ *such that*

$$\sum_{j=0}^{2^k-1} f_{2^k i+j} L_{\alpha,j}(x-\lambda-\omega_{\gamma,i}) = \sum_{j=0}^{2^k-1} g_{2^k i+j} X_{\alpha,j}$$

for $i \in \{0, \ldots, 2^{n-k}-1\}$, *and*

$$\sum_{i=0}^{2^{n-k}-1} g_{2^k i+j} L_{\delta,i}(x-S_{\beta,k}(\lambda)) = \sum_{i=0}^{2^{n-k}-1} h_{2^k i+j} X_{\delta,i}$$

for $j \in \{0, \ldots, 2^k-1\}$.

Conversion between the Lagrange and LCH bases

By always taking $k = n - 1$, we obtain the following complexity bounds.

Multiplications:	$\min\left((\ell-1)\left(\lceil\log_2 \ell\rceil+1\right)/2, 2^{n-1}n\right)$
Additions:	$\min\left((\ell-1)\left(\lceil\log_2 \ell\rceil+2\right), (2^n-1)(n+1)\right)$
Precomputations:	$\mathcal{O}(n^2)$ operations
Auxiliary space:	$2^n - \ell + \mathcal{O}(n^2)$ field elements

The case $\ell < 2^n$ is handled by applying ideas from truncated FFTs.

Conversion between the monomial and LCH bases

Corollary

Suppose that $f_0, \ldots, f_{\ell-1}, h_0, \ldots, h_{\ell-1} \in \mathbb{F}$ *satisfy*

$$\sum_{i=0}^{\ell-1} f_i X_{\beta,i} = \sum_{i=0}^{\lceil \ell/2^k \rceil - 1} \left(\sum_{j=0}^{\min(\ell-2^k i, 2^k)-1} h_{2^k i+j} x^j \right) S^i_{\beta,k}.$$

Then there exist unique elements $g_0, \ldots, g_{\ell-1} \in \mathbb{F}$ *such that*

$$\sum_{j=0}^{\min(\ell-2^k i, 2^k)-1} f_{2^k i+j} X_{\alpha,j} = \sum_{j=0}^{\min(\ell-2^k i, 2^k)-1} g_{2^k i+j} x^j$$

for $i \in \{0, \ldots, \lceil \ell/2^k \rceil - 1\}$, *and*

$$\sum_{i=0}^{\lceil (\ell-j)/2^k \rceil - 1} g_{2^k i+j} X_{\delta,i} = \sum_{i=0}^{\lceil (\ell-j)/2^k \rceil - 1} h_{2^k i+j} x^i$$

for $j \in \{0, \ldots, \min(2^k, \ell) - 1\}$.

Problem

We want to perform $\mathcal{O}(\ell \log \ell)$ multiplications overall, so we need to perform $\mathcal{O}(\ell)$ multiplications in the Taylor expansion step.

But

$$S_{\beta,k} = x^{2^k} + \sum_{i=0}^{k-1} s_{k,i} x^{2^i} \quad \text{with } s_{k,0}, \ldots, s_{k,k-1} \in \mathbb{F}.$$

For arbitrarily chosen k, it is unclear how to obtain the desired complexity for the Taylor expansion step.

Not a solution

If $\beta_0, \ldots, \beta_{k-1} \in \mathbb{F}_{2^k}$ for some $k \in \{1, \ldots, n-1\}$, then

$$S_{\beta,k} = \prod_{\omega \in \mathbb{F}_{2^k}} x - \omega = x^{2^k} - x.$$

Using the algorithms of Gao and Mateer, the Taylor expansion step then requires at most $\lfloor \ell/2 \rfloor \lceil \log_2 \lceil \ell/2^k \rceil \rceil$ additions and zero multiplications.

But we cannot guarantee that such a value of k exists.

A solution

If $\beta_0/\beta_0, \ldots, \beta_{k-1}/\beta_0 \in \mathbb{F}_{2^k}$ for some $k \in \{1, \ldots, n-1\}$, then

$$S_{\beta,k} = x^{2^k} - \beta_0^{2^k-1} x = \beta_0^{2^k} \left(\left(\frac{x}{\beta_0} \right)^{2^k} - \frac{x}{\beta_0} \right).$$

We can always take $k = 1$.

$\mathcal{O}(\ell)$ multiplications allow us to reduce to Taylor expansion at $x^{2^k} - x$.

Using the algorithms of Gao and Mateer, the Taylor expansion step then requires at most $\lfloor \ell/2 \rfloor \lceil \log_2 \lceil \ell/2^k \rceil \rceil$ additions and $\mathcal{O}(\ell)$ multiplications.

A better solution

Lemma

Suppose that $n \geq 2$. *Let* $k \in \{1, \ldots, n-1\}$ *s.t.* $\beta_0/\beta_0, \ldots, \beta_{k-1}/\beta_0 \in \mathbb{F}_{2^k}$, $\alpha = (\beta_0, \ldots, \beta_{k-1})$ *and*

$$\delta = \left(\left(\frac{\beta_k}{\beta_0} \right)^{2^k} - \frac{\beta_k}{\beta_0}, \ldots, \left(\frac{\beta_{n-1}}{\beta_0} \right)^{2^k} - \frac{\beta_{n-1}}{\beta_0} \right).$$

Then the entries of δ *are linearly independent over* $\mathbb{F}_2$, *and*

$$X_{\beta, 2^k i + j}(\beta_0 x) = X_{\delta, i}\left(\delta_0 \frac{x^{2^k} - x}{\delta_0} \right) X_{\alpha, j}(\alpha_0 x)$$

for $i \in \{0, \ldots, 2^{n-k} - 1\}$ *and* $j \in \{0, \ldots, 2^k - 1\}$.

Conversion between the monomial and LCH bases

Corollary

Suppose that $f_0, \ldots, f_{\ell-1}, h_0, \ldots, h_{\ell-1} \in \mathbb{F}$ *satisfy*

$$\sum_{i=0}^{\ell-1} f_i X_{\beta,i}(\beta_0 x) = \sum_{i=0}^{\lceil \ell/2^k \rceil - 1} \left(\sum_{j=0}^{\min(\ell-2^k i, 2^k)-1} h_{2^k i+j} x^j \right) \left(\frac{x^{2^k} - x}{\delta_0} \right)^i .$$

Then there exist unique elements $g_0, \ldots, g_{\ell-1} \in \mathbb{F}$ *such that*

$$\sum_{j=0}^{\min(\ell-2^k i, 2^k)-1} f_{2^k i+j} X_{\alpha,j}(\alpha_0 x) = \sum_{j=0}^{\min(\ell-2^k i, 2^k)-1} g_{2^k i+j} x^j$$

for $i \in \{0, \ldots, \lceil \ell/2^k \rceil - 1\}$, *and*

$$\sum_{i=0}^{\lceil (\ell-j)/2^k \rceil - 1} g_{2^k i+j} X_{\delta,i}(\delta_0 x) = \sum_{i=0}^{\lceil (\ell-j)/2^k \rceil - 1} h_{2^k i+j} x^i$$

for $j \in \{0, \ldots, \min(2^k, \ell) - 1\}$.

Conversion between the monomial and LCH bases

We instead convert between the monomial basis and the "twisted" LCH basis $\{X_{\beta,0}(\beta_0 x), \ldots, X_{\beta,2^n-1}(\beta_0 x)\}$.

Converting to and from the LCH basis then requires an additional $\mathcal{O}(\ell)$ multiplications for performing the substitution $x \mapsto \beta_0 x$ or $x \mapsto x/\beta_0$ on the monomial basis.

Taylor expansion at $(x^{2^k} - x)/\delta_0$ requires at most $\lfloor \ell/2 \rfloor \left\lceil \log_2 \left\lceil \ell/2^k \right\rceil \right\rceil$ additions and $\mathcal{O}(\ell)$ multiplications.

$X_{\beta,0}(\beta_0 x) = 1$ and $X_{\beta,1}(\beta_0 x) = x$, so all multiplications in the base case are eliminated.

Conversion between the monomial and LCH bases

By always taking $k = 1$, we obtain the following complexity bounds for converting between the twisted LCH basis and the monomial basis.

Multiplications:	$\lfloor \ell/2 \rfloor \, (3 \lceil \log_2 \ell \rceil - 4) + 1$
Additions:	$\lfloor \ell/2 \rfloor \, \binom{\lceil \log_2 \ell \rceil}{2}$
Precomputations:	$\mathcal{O}(\log^2 \ell)$ operations
Auxiliary space:	$\mathcal{O}(\log \ell)$ field elements

After equalising precomputations, our algorithms perform fewer multiplications than those of Lin, Al-Naffouri, Han and Chung.

How should we choose β?

Suppose now that we are free to choose β, but the field $\mathbb{F}$ and the dimension of the vector are fixed.

Then how should we chose β?

Wishlist

- $\beta_0 = 1$,
- Large values of k permitted,
- Multiplications by elements of small subfields (e.g., $\mathbb{F}_2$).

Very special case: $\mathbb{F} \supseteq \mathbb{F}_{2^{2^{\lceil \log_2 n \rceil}}}$

If $\mathbb{F}_{2^{2^{\lceil \log_2 n \rceil}}} \subseteq \mathbb{F}$, then take $\beta = (\beta_0, \ldots, \beta_{n-1})$ to be a "Cantor basis":

$$\beta_0 = 1 \quad \text{and} \quad \beta_i = \beta_{i+1}^2 - \beta_{i+1} \quad \text{for } i \in \{0, \ldots, n-2\}.$$

Then we can always take $k = 2^{\lceil \log_2 n \rceil - 1}$.

$\Rightarrow$ Perform at most $\lfloor \ell/2 \rfloor \lceil \log_2 \ell \rceil \lceil \log_2 \log_2 \ell \rceil$ additions.

Moreover, $\delta = (\beta_0, \ldots, \beta_{n-k-1})$ for this choice of k.

$\Rightarrow$ Perform no multiplications since $\delta_0 = \beta_0 = 1$.

Reduces to the algorithm of Lin, Al-Naffouri, Han and Chung for $\ell = 2^n$.

Less special case: tower of subfields

Suppose there exists a tower of subfields

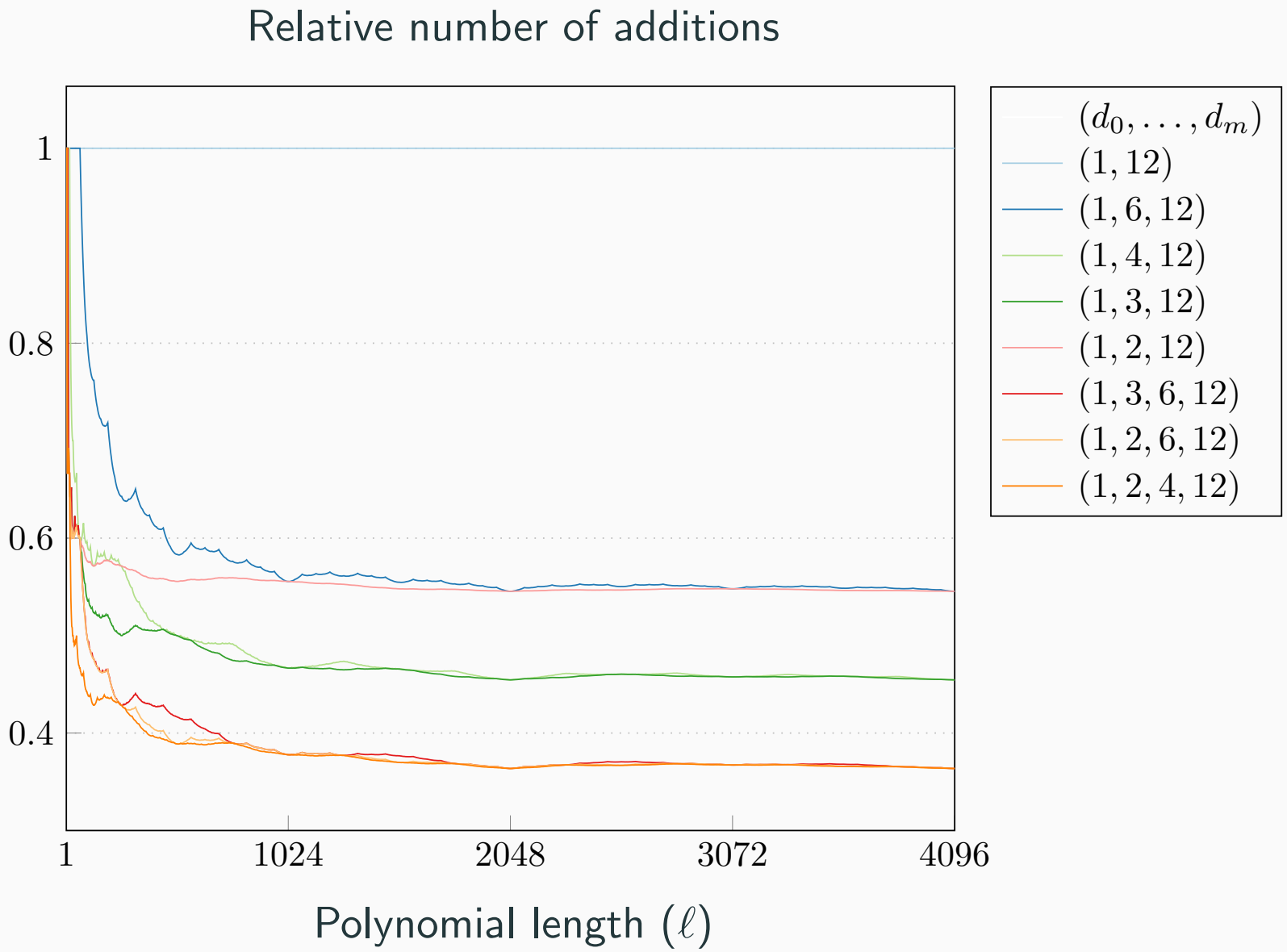
$$\mathbb{F}_2 = \mathbb{F}_{2^{d_0}} \subset \mathbb{F}_{2^{d_1}} \subset \cdots \subset \mathbb{F}_{2^{d_m}} = \mathbb{F}.$$

Construct $\beta = (\beta_0, \ldots, \beta_{n-1})$ as follows:

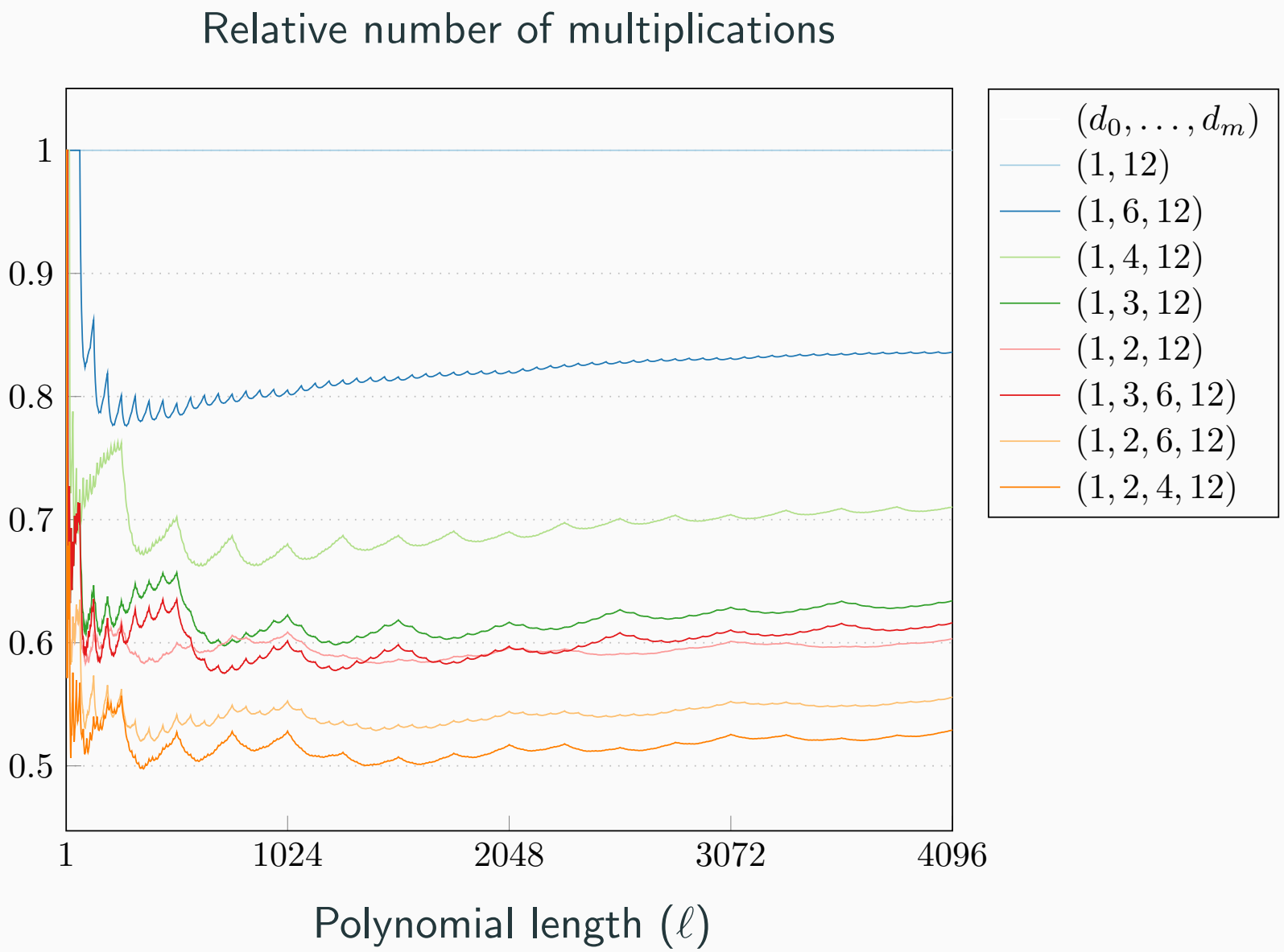
1. Choose a basis $\{\vartheta_{i,0}, \ldots, \vartheta_{i,d_{i+1}/d_i-1}\}$ for each extension $\mathbb{F}_{2^{d_{i+1}}}/\mathbb{F}_{2^{d_i}}$.
2. Set
$$\beta_i = \prod_{j=0}^{m-1} \vartheta_{j,i_j} \quad \text{such that} \quad \sum_{j=0}^{m-1} i_j d_j = i,$$
for $i \in \{0, \ldots, n\}$.

Then we can always take $k = \max\{d_i \mid d_i < n\}$.

Example: $\mathbb{F} = \mathbb{F}_{2^{12}}$, $n = 12$



Example: $\mathbb{F} = \mathbb{F}_{2^{12}}$, $n = 12$



Fewer mutliplications

We have complete freedom in the choice of bases for the extensions.

We should use this freedom to either eliminate multiplications, or force them to be by elements of small subfields.

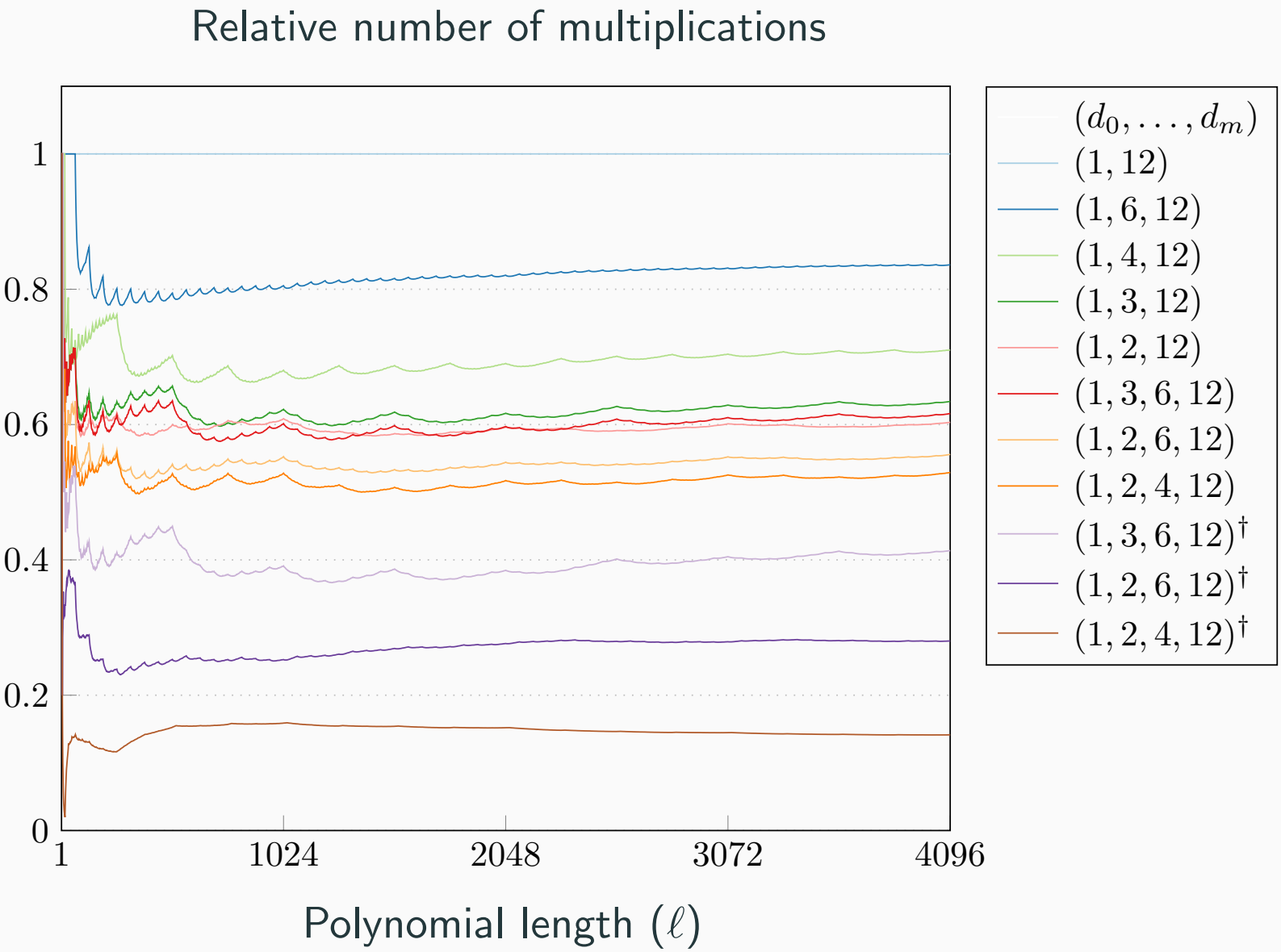
Example

If $\mathbb{F}_{2^{d_{i+1}}}/\mathbb{F}_{2^{d_i}}$ is a quadratic extension, then choose its basis $\{\vartheta_{i,0}, \vartheta_{i,1}\}$ such that

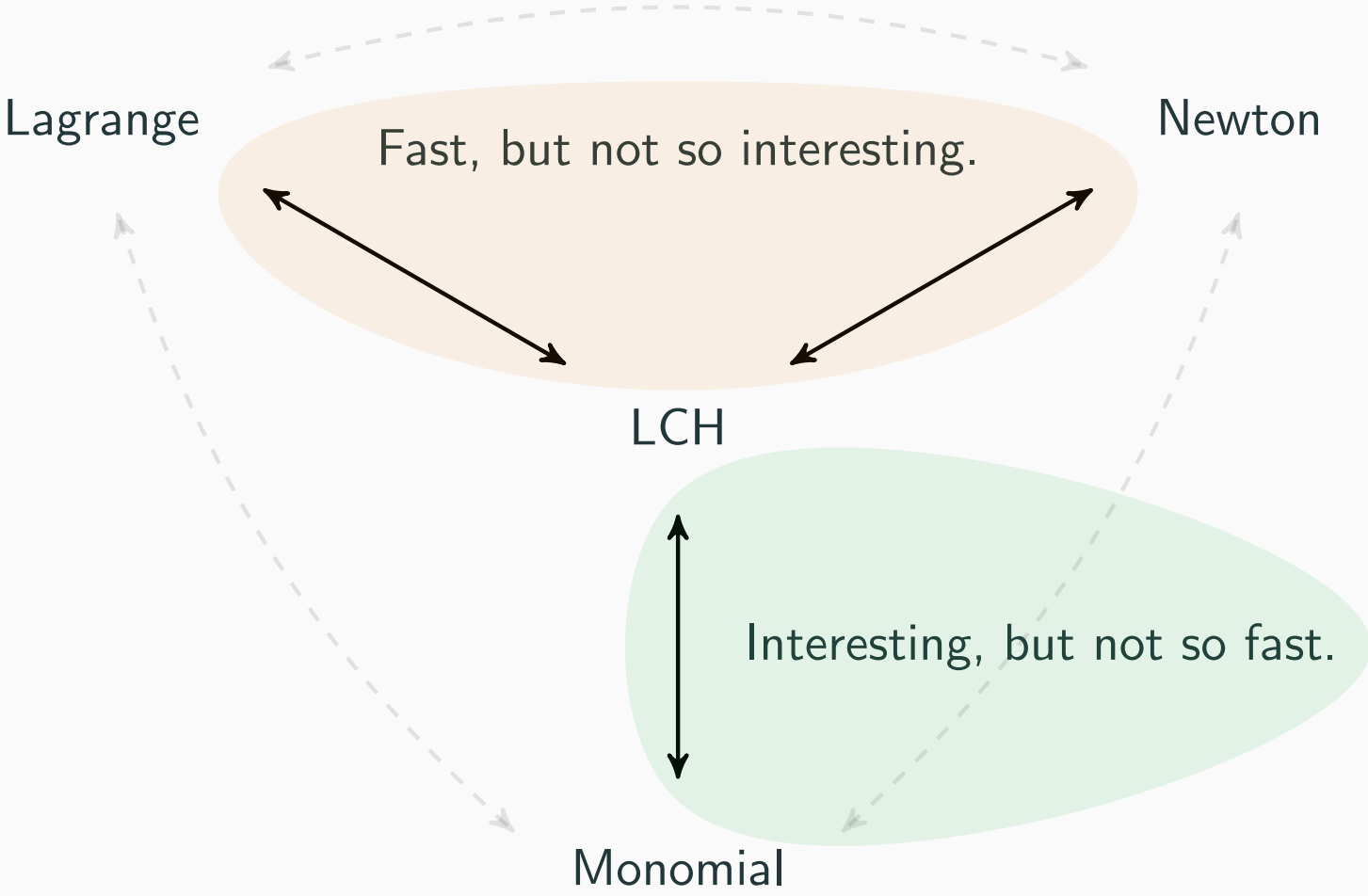
$$\mathrm{Tr}_{\mathbb{F}_{2^{d_{i+1}}}/\mathbb{F}_{2^{d_i}}}\left(\frac{\vartheta_{i,1}}{\vartheta_{i,0}}\right) = 1.$$

Then each time $k = d_i$, the Taylor expansion step is multiplication-free.

Example: $\mathbb{F} = \mathbb{F}_{2^{12}}$, $n = 12$



Conclusion



Future work: answer the (correct) question "How should we choose $S_{\beta,n}$?".

Fast Hermite interpolation and evaluation over finite fields of characteristic two

Hasse derivatives

Since we are working over small characteristic, derivative is taken to mean Hasse derivative.

Definition

For $i \in \mathbb{N}$, the ith Hasse derivative $D^i : \mathbb{F}[x] \to \mathbb{F}[x]$ sends $F \in \mathbb{F}[x]$ to the coefficient of y^i in the polynomial $F(x+y) \in \mathbb{F}[x][y]$.

The Hasse derivatives of $F \in \mathbb{F}[x]$ satisfy

$$F(x+y) = \sum_{i \in \mathbb{N}} (D^i F)(x) y^i.$$

Hermite interpolation and evaluation

Problem

Let $\omega_0, \ldots, \omega_{m-1} \in \mathbb{F}$ be distinct, $\ell_0, \ldots, \ell_{m-1}$ be positive integers, $\ell = \ell_0 + \cdots + \ell_{m-1}$, and $F \in \mathbb{F}[x]_\ell$.

Hermite interpolation: Given $(D^j F)(\omega_i)$ for $j \in \{0, \ldots, \ell_i - 1\}$ and $i \in \{0, \ldots, m-1\}$, compute F.

Hermite evaluation: Given F, compute $(D^j F)(\omega_i)$ for $j \in \{0, \ldots, \ell_i - 1\}$ and $i \in \{0, \ldots, m-1\}$.

Polynomials are assumed to be represented w.r.t. the monomial basis.

We call ℓ the length of an instance.

(A variant of) Chin's algorithm

For $F \in \mathbb{F}[x]_\ell$, we have

$$F \equiv \sum_{j=0}^{\ell_i - 1} (D^j F)(\omega_i)\,(x - \omega_i)^j \pmod{(x - \omega_i)^{\ell_i}}$$

for $i \in \{0, \ldots, m-1\}$.

Taylor expansion to convert between derivatives and residues.

Remainder trees/fast CRT to convert between residues and polynomials.

Performs $\mathcal{O}(\mathsf{M}(\ell) \log \ell)$ field operations.

Our problem

For now, we focus on the special case $m = |\mathbb{F}|$ and $\ell_0 = \cdots = \ell_{m-1} = 2^n$.

Let $|\mathbb{F}| = q$ and fix some enumeration $\mathbb{F} = \{\omega_0, \ldots, \omega_{q-1}\}$.

For $n \in \mathbb{N}$, define $\mathsf{H}_n : \mathbb{F}[x]_{2^n q} \to \mathbb{F}^{2^n q}$ by

$$F \mapsto \left(\left(D^{\lfloor i/q \rfloor} F\right)(\omega_{i \bmod q})\right)_{i=0,\ldots,2^n q-1}.$$

In particular, $\mathsf{H}_0(F) = (F(\omega_0), \ldots, F(\omega_{q-1}))$ for $F \in \mathbb{F}[x]_q$.

The maps H_n^{-1} and H_n respectively capture the Hermite interpolation and evaluation problems that we are considering.

Main lemma

Lemma

Let $n \in \mathbb{N}$ be nonzero and $F \in \mathbb{F}[x]_{2^n q}$. Write

$$F = F_1(x^q - x)^{2^{n-1}} + F_0$$

with $F_0, F_1 \in \mathbb{F}[x]_{2^{n-1}q}$. Then for $\omega \in \mathbb{F}$ and $i \in \{0, \ldots, 2^n - 1\}$,

$$\left(D^i F\right)(\omega) = \begin{cases} \left(D^i F_0\right)(\omega) & \text{if } i < 2^{n-1}, \\ \left(D^{i-2^{n-1}} F_1\right)(\omega) + \left(D^i F_0\right)(\omega) & \text{otherwise.} \end{cases}$$

Proof. For $\omega \in \mathbb{F}$,

$$F(x+\omega) = \sum_{i \in \mathbb{N}} (D^i F)(\omega) x^i = F_1(x+\omega)(x^{2^{n-1}q} + x^{2^{n-1}}) + F_0(x+\omega).$$

Main lemma

Lemma

Let $n \in \mathbb{N}$ be nonzero and $F \in \mathbb{F}[x]_{2^n q}$. Write

$$F = F_1(x^q - x)^{2^{n-1}} + F_0$$

with $F_0, F_1 \in \mathbb{F}[x]_{2^{n-1} q}$. Then for $\omega \in \mathbb{F}$ and $i \in \{0, \ldots, 2^n - 1\}$,

$$\left(D^i F\right)(\omega) = \begin{cases} \left(D^i F_0\right)(\omega) & \text{if } i < 2^{n-1}, \\ \left(D^{i-2^{n-1}} F_1\right)(\omega) + \left(D^i F_0\right)(\omega) & \text{otherwise.} \end{cases}$$

We have $D^i \circ D^j = \binom{i+j}{j} D^{i+j}$ for $i, j \in \mathbb{N}$. It follows by Lucas' lemma that

$$D^{i-2^{n-1}} \circ D^{2^{n-1}} = D^i \quad \text{for } i \in \{2^{n-1}, \ldots, 2^n - 1\}.$$

Main lemma

Lemma

Let $n \in \mathbb{N}$ be nonzero and $F \in \mathbb{F}[x]_{2^n q}$. Write

$$F = F_1(x^q - x)^{2^{n-1}} + F_0$$

with $F_0, F_1 \in \mathbb{F}[x]_{2^{n-1} q}$. Then for $\omega \in \mathbb{F}$ and $i \in \{0, \ldots, 2^n - 1\}$,

$$\left(D^i F\right)(\omega) = \begin{cases} \left(D^i F_0\right)(\omega) & \text{if } i < 2^{n-1}, \\ \left(D^{i-2^{n-1}}(F_1 + D^{2^{n-1}} F_0)\right)(\omega) & \text{otherwise.} \end{cases}$$

We have $D^i \circ D^j = \binom{i+j}{j} D^{i+j}$ for $i, j \in \mathbb{N}$. It follows by Lucas' lemma that

$$D^{i-2^{n-1}} \circ D^{2^{n-1}} = D^i \quad \text{for } i \in \{2^{n-1}, \ldots, 2^n - 1\}.$$

Main lemma

Lemma

Let $n \in \mathbb{N}$ be nonzero and $F \in \mathbb{F}[x]_{2^n q}$. Write

$$F = F_1(x^q - x)^{2^{n-1}} + F_0$$

with $F_0, F_1 \in \mathbb{F}[x]_{2^{n-1} q}$. Then

$$\mathrm{H}_n(F) = \mathrm{H}_{n-1}\big(F_0\big) \oplus \mathrm{H}_{n-1}\big(F_1 + D^{2^{n-1}} F_0\big).$$

Recall that $\mathrm{H}_0(F) = (F(\omega_0), \ldots, F(\omega_{q-1}))$ for $F \in \mathbb{F}[x]_q$.

Cost of the reduction

We have

$$F = F_1(x^q - x)^{2^{n-1}} + F_0 = x^{2^{n-1}q}F_1 + x^{2^{n-1}}F_1 + F_0.$$

Consequently, $F \leftrightarrow (F_0, F_1)$ costs $2^{n-1}q$ additions.

If $F_0 = \sum_{i=0}^{2^{n-1}q-1} f_i x^i$, then Lucas' lemma implies that

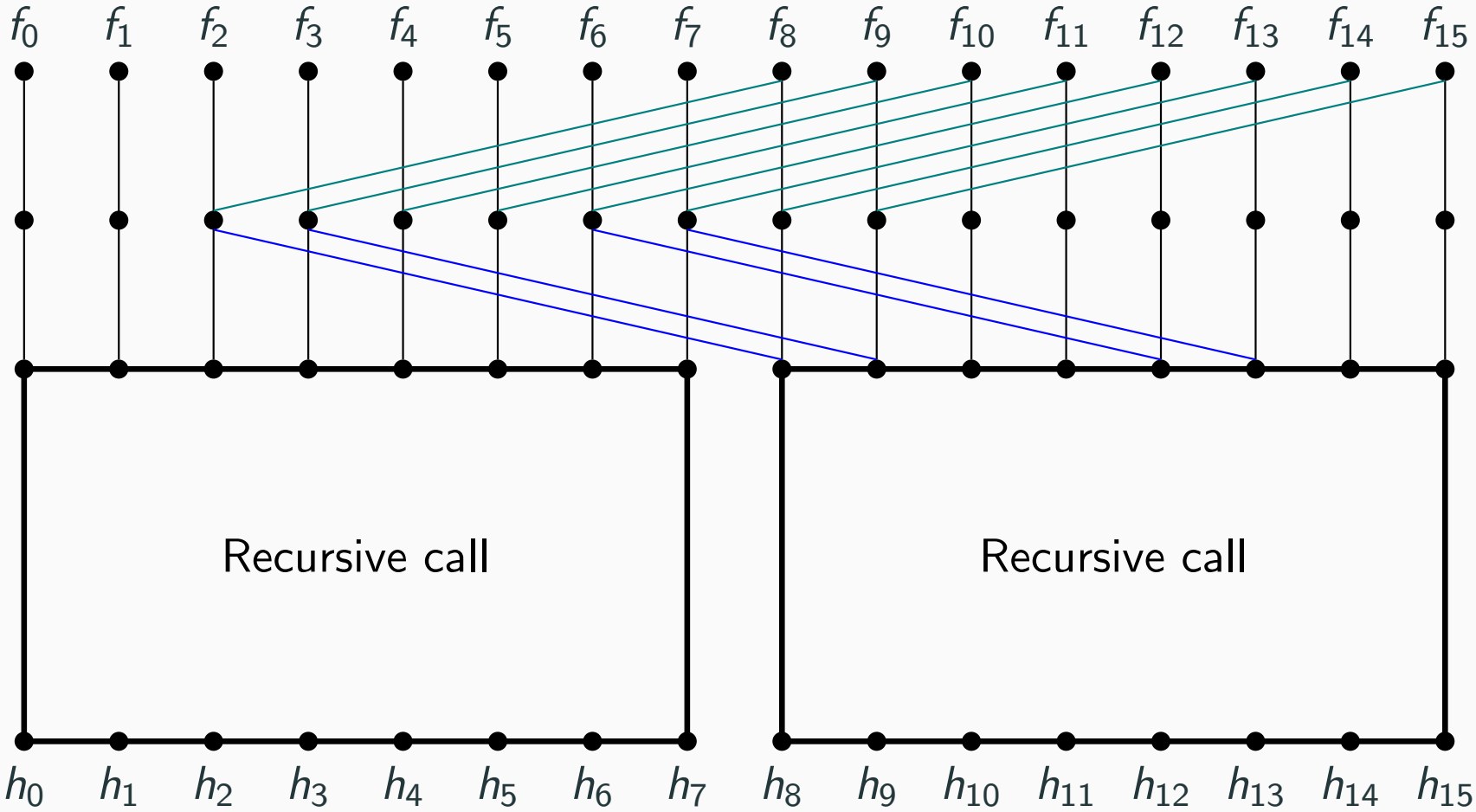
$$D^{2^{n-1}}F_0 = \sum_{i=0}^{q/2-1} x^{2^n i} \sum_{j=0}^{2^{n-1}-1} f_{2^{n-1}(2i+1)+j} x^j.$$

Thus, $(F_0, F_1) \leftrightarrow (F_0, F_1 + D^{2^{n-1}}F_0)$ costs $2^{n-2}q$ additions.

Therefore, $F \leftrightarrow (F_0, F_1 + D^{2^{n-1}}F_0)$ costs $3 \cdot 2^{n-2}q$ additions.***

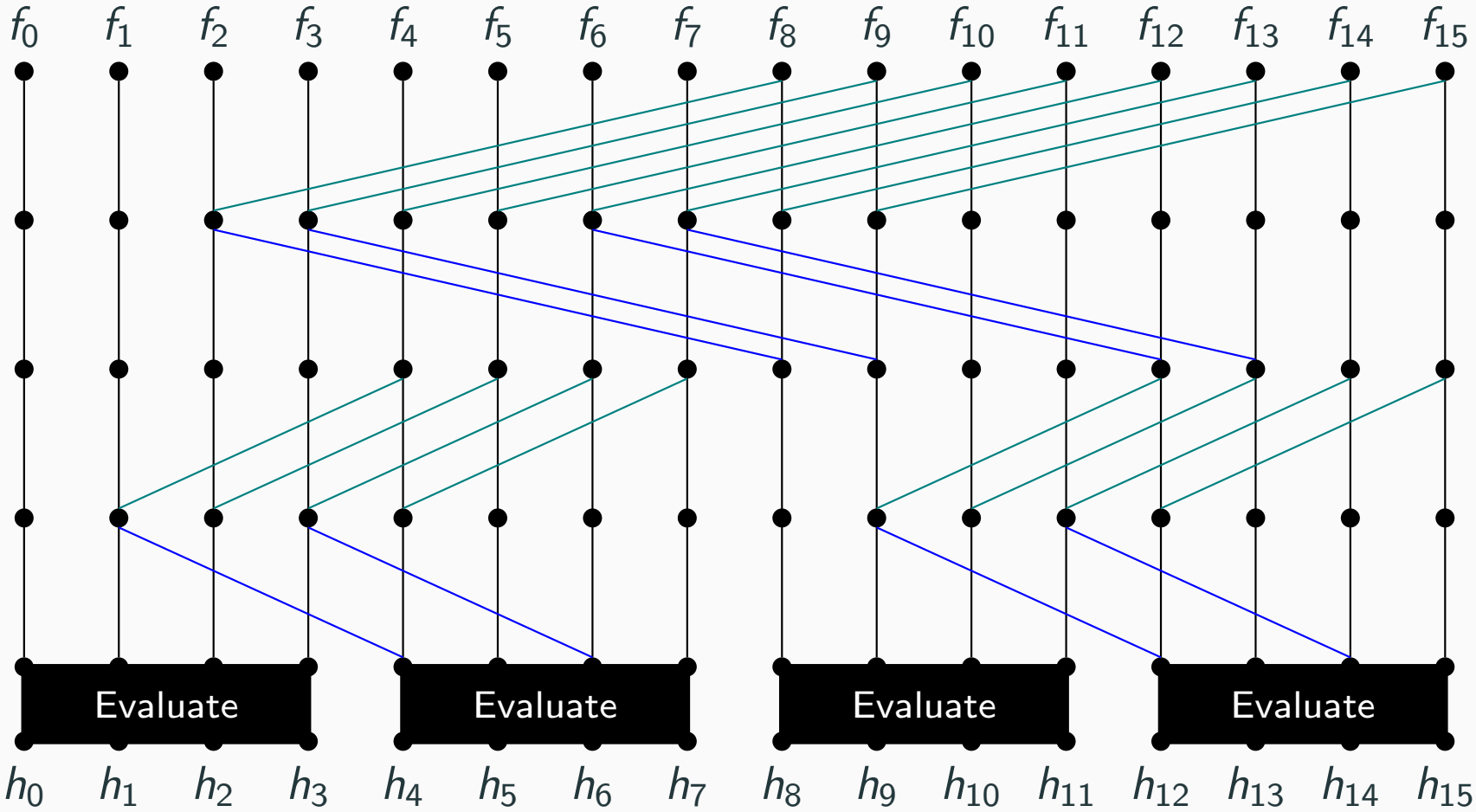
Evaluation example: $q = 4$, $n = 2$

Let $F = \sum_{i=0}^{15} f_i x^i$ and $\mathrm{H}_2(F) = (h_0, \ldots, h_{15})$.



Evaluation example: $q = 4$, $n = 2$

Let $F = \sum_{i=0}^{15} f_i x^i$ and $\mathrm{H}_2(F) = (h_0, \ldots, h_{15})$.



Complexity: $2^n(A_0(q) + 3nq/4)$ additions and $2^n M_0(q)$ multiplications.

Evaluation complexity

For $c \in \{1, \ldots, q\}$, let $A_0(c)$ and $M_0(c)$ respectively denote the number of additions and multiplications needed to compute the first c entries of $\mathrm{H}_0(F)$ when given a polynomial $F \in \mathbb{F}[x]_q$.

Theorem

Given $F \in \mathbb{F}[x]_{2^n q}$ and $c \in \{1, \ldots, 2^n q\}$, the first c entries of $\mathrm{H}_n(F)$ can be computed with $M_0(q)\,(\lceil c/q \rceil - 1) + M_0(c \bmod^* q)$ multiplications, and at most

$$A_0(q)\,(\lceil c/q \rceil - 1) + A_0(c \bmod^* q) + \left(\frac{3}{4}\lceil \log_2 \lceil c/q \rceil \rceil - \frac{1}{4}\right)(\lceil c/q \rceil - 1)\,q + (2^n - 1)\,q$$

additions.

Evaluation complexity

For $c \in \{1, \ldots, q\}$, let $A_0(c)$ and $M_0(c)$ respectively denote the number of additions and multiplications needed to compute the first c entries of $\mathrm{H}_0(F)$ when given a polynomial $F \in \mathbb{F}[x]_q$.

Theorem

Given $F \in \mathbb{F}[x]_{2^n q}$ and $c \in \{1, \ldots, 2^n q\}$, the first c entries of $\mathrm{H}_n(F)$ can be computed with $M_0(q)\,(\lceil c/q \rceil - 1) + M_0(c \bmod^* q)$ multiplications, and at most

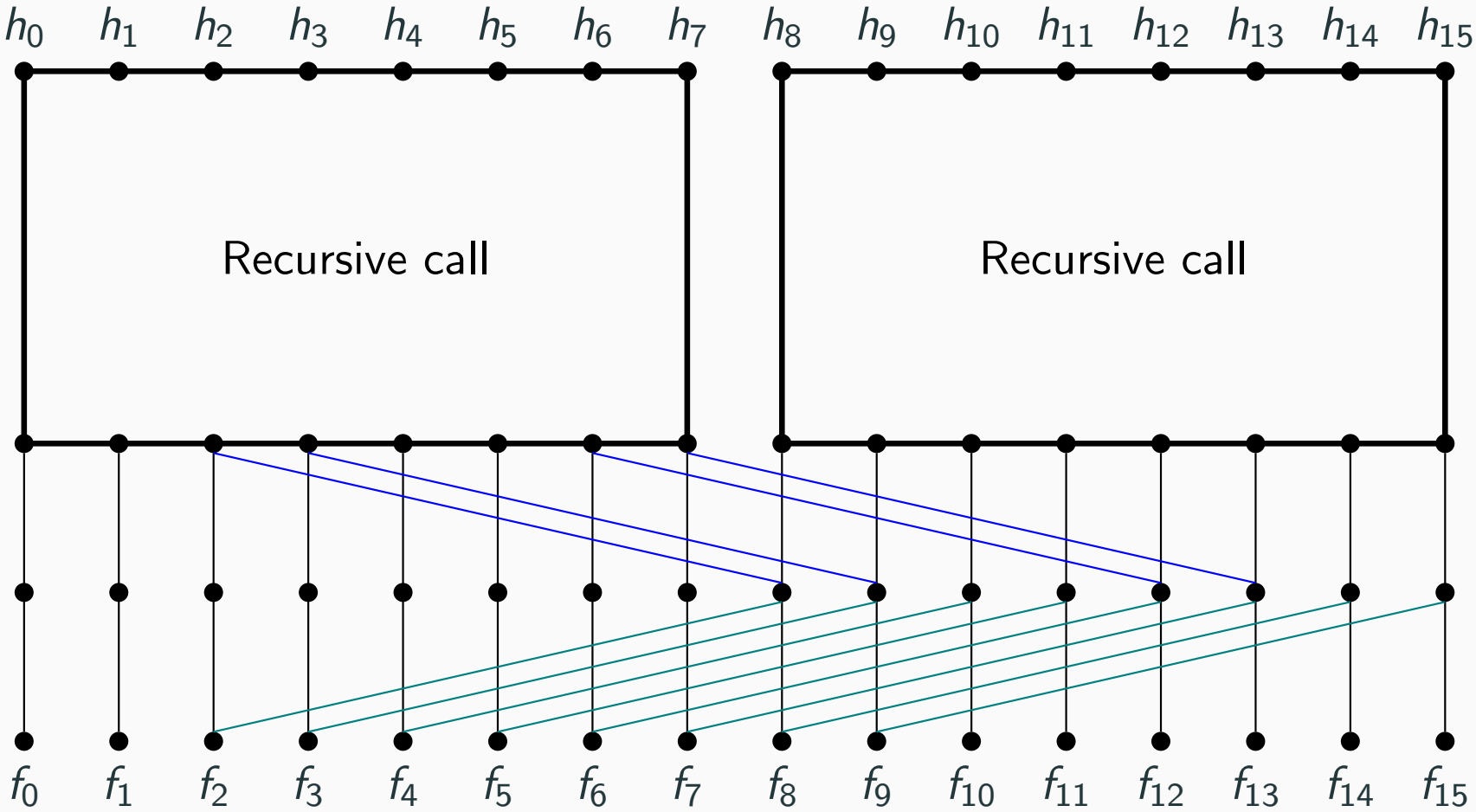
$$A_0(q)\,(\lceil c/q \rceil - 1) + A_0(c \bmod^* q) + \left(\frac{3}{4}\lceil \log_2 \lceil c/q \rceil \rceil - \frac{1}{4}\right)(\lceil c/q \rceil - 1)\,q + (2^n - 1)\,q$$

additions.

If in addition $F \in \mathbb{F}[x]_\ell \subset \mathbb{F}[x]_{2^n q}$, we just discard some more additions.

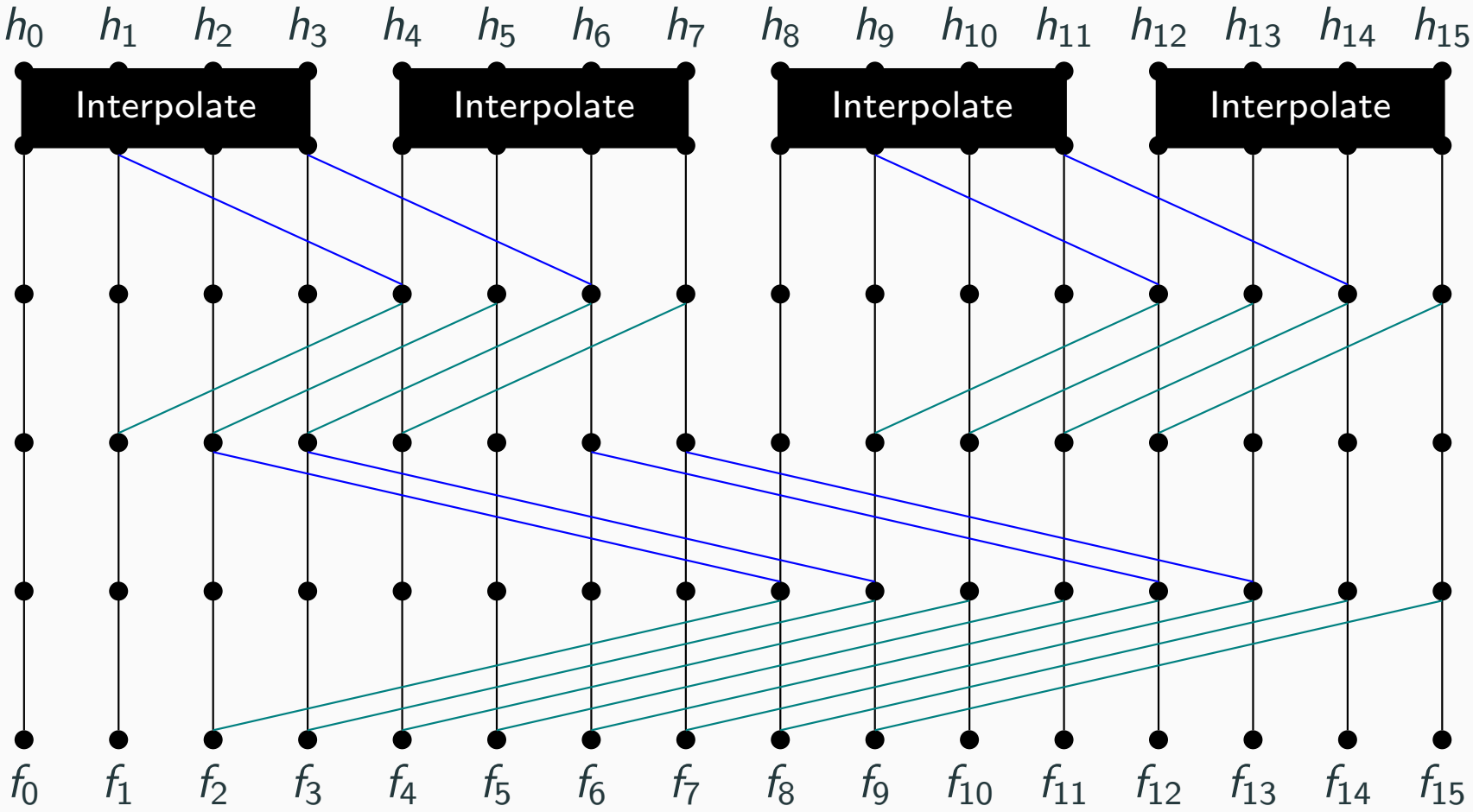
Interpolation example: $q = 4$, $n = 2$

Let $F = \sum_{i=0}^{15} f_i x^i$ and $\mathrm{H}_2(F) = (h_0, \ldots, h_{15})$.



Interpolation example: $q = 4$, $n = 2$

Let $F = \sum_{i=0}^{15} f_i x^i$ and $\mathrm{H}_2(F) = (h_0, \ldots, h_{15})$.



Complexity: $2^n(A_0(q) + 3nq/4)$ additions and $2^n M_0(q)$ multiplications.

New problem

Problem

Given the first ℓ entries of $\mathrm{H}_n(F)$ for some polynomial $F \in \mathbb{F}[x]_\ell \subseteq \mathbb{F}[x]_{2^n q}$, compute F.

This is still a special case of the general Hermite interpolation problem, but it is sufficient for applications in coding theory.

New problem

Problem

Given the first ℓ entries of $\mathsf{H}_n(F)$ for some polynomial $F \in \mathbb{F}[x]_\ell \subseteq \mathbb{F}[x]_{2^n q}$, compute F.

To address the problem, we follow the approach of truncated FFTs by solving the following more general problem.

Problem

Let $F = \sum_{i=0}^{2^n q - 1} f_i x^i \in \mathbb{F}[x]_{2^n q}$. Given the first c entries of $\mathsf{H}_n(F)$ and $f_c, \ldots, f_{2^n q - 1}$, compute $f_0, \ldots, f_{c-1}$.

We can then solve the first problem by taking $c = \ell$, since we then know that $f_c = \cdots = f_{2^n q - 1} = 0$.

Interpolation complexity

Theorem

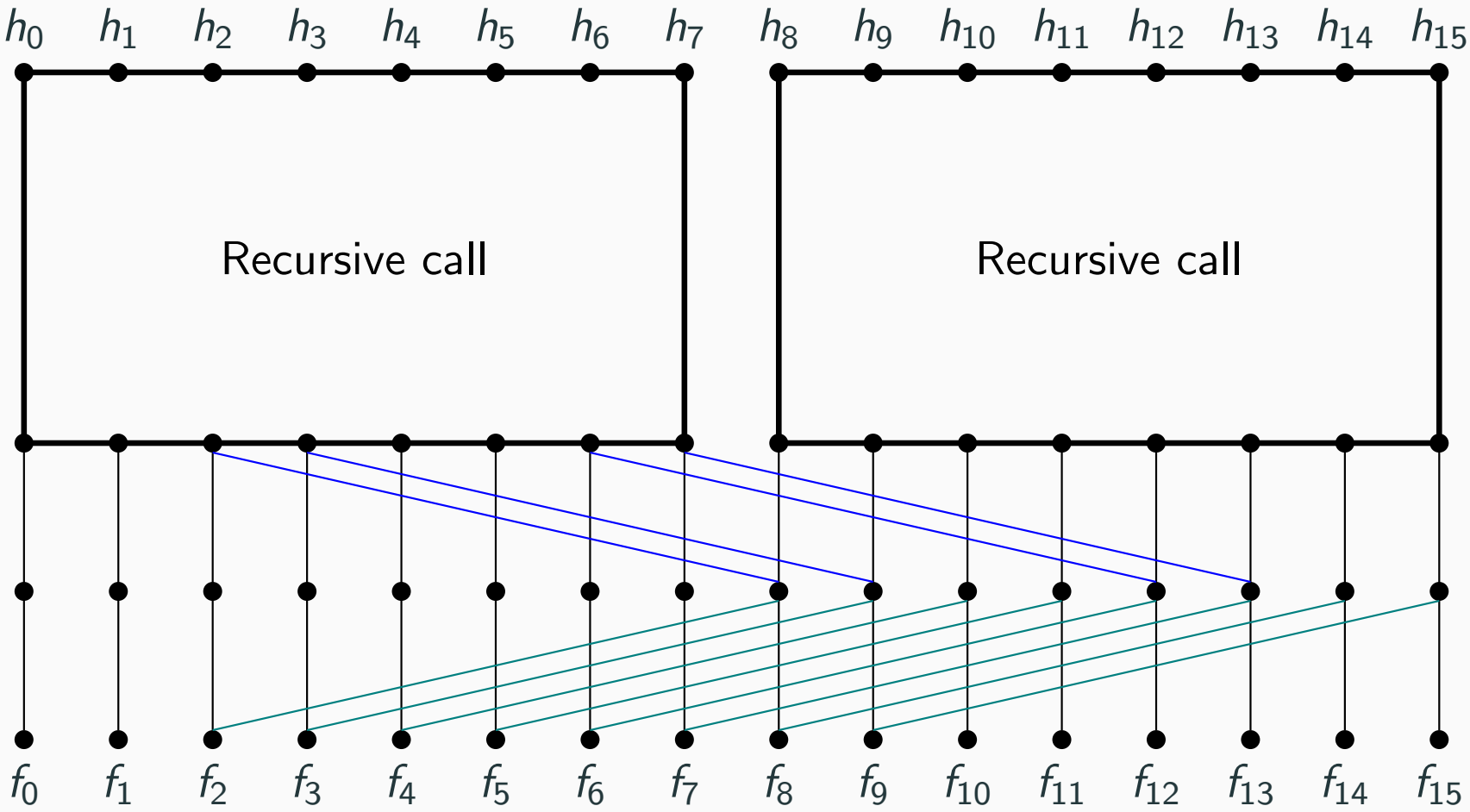
Suppose that we can solve the problem for $n = 0$ with $A_0(c)$ additions and $M_0(c)$ multiplications. Then the problem for $n \geq 1$ can be solved with $M_0(q)\,(\lceil c/q \rceil - 1) + M_0(c \bmod^* q)$ multiplications, and at most

$$A_0(q)\,(\lceil c/q \rceil - 1) + A_0(c \bmod^* q) + \left(\frac{7}{4}\lceil \log_2 \lceil c/q \rceil \rceil - \frac{3}{4}\right)(\lceil c/q \rceil - 1)\,q + (2^n - 1)\,(2q + 1)$$

additions.

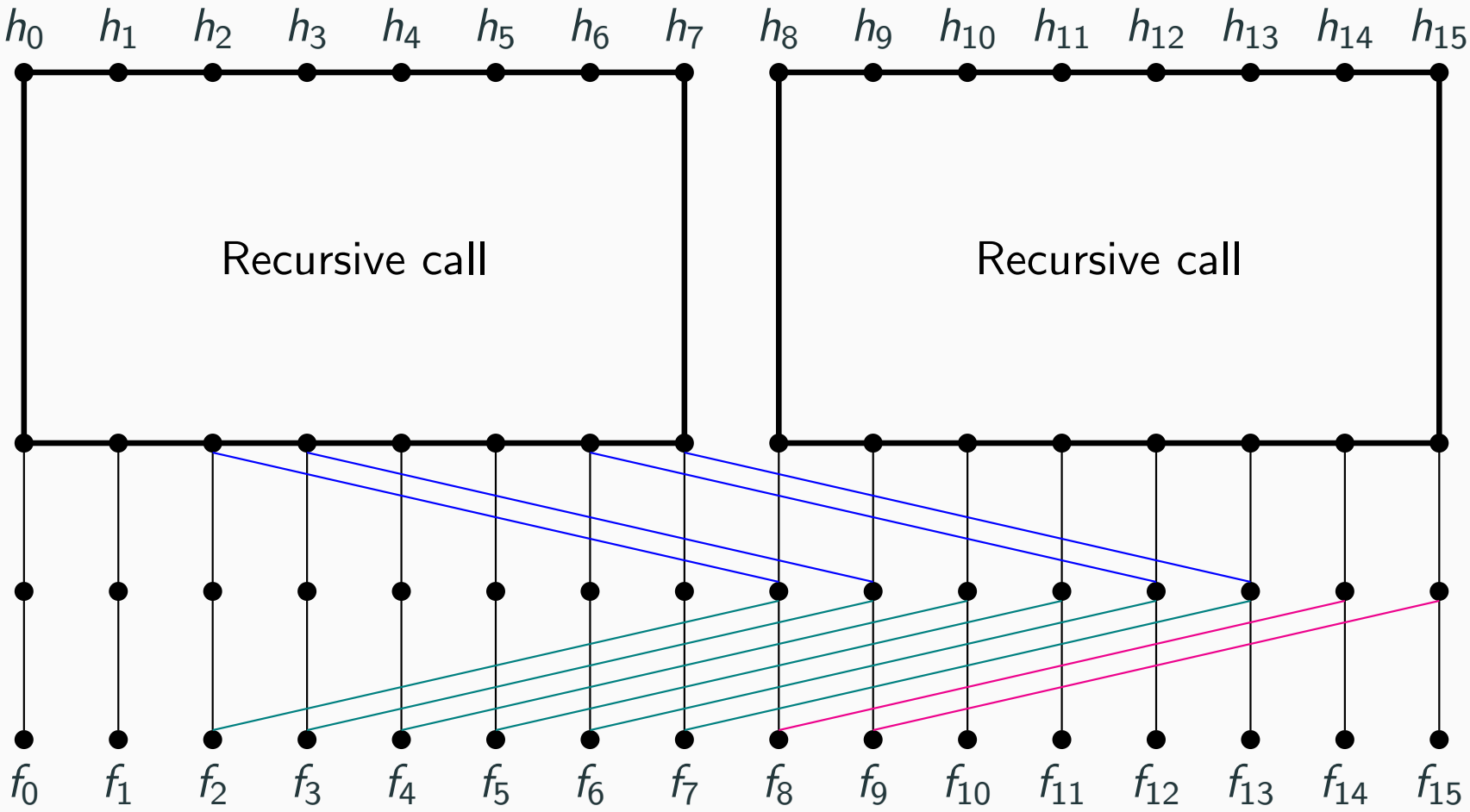
Interpolation example: $q = 4$, $n = 2$

Let $F = \sum_{i=0}^{15} f_i x^i$ and $\mathrm{H}_2(F) = (h_0, \ldots, h_{15})$.



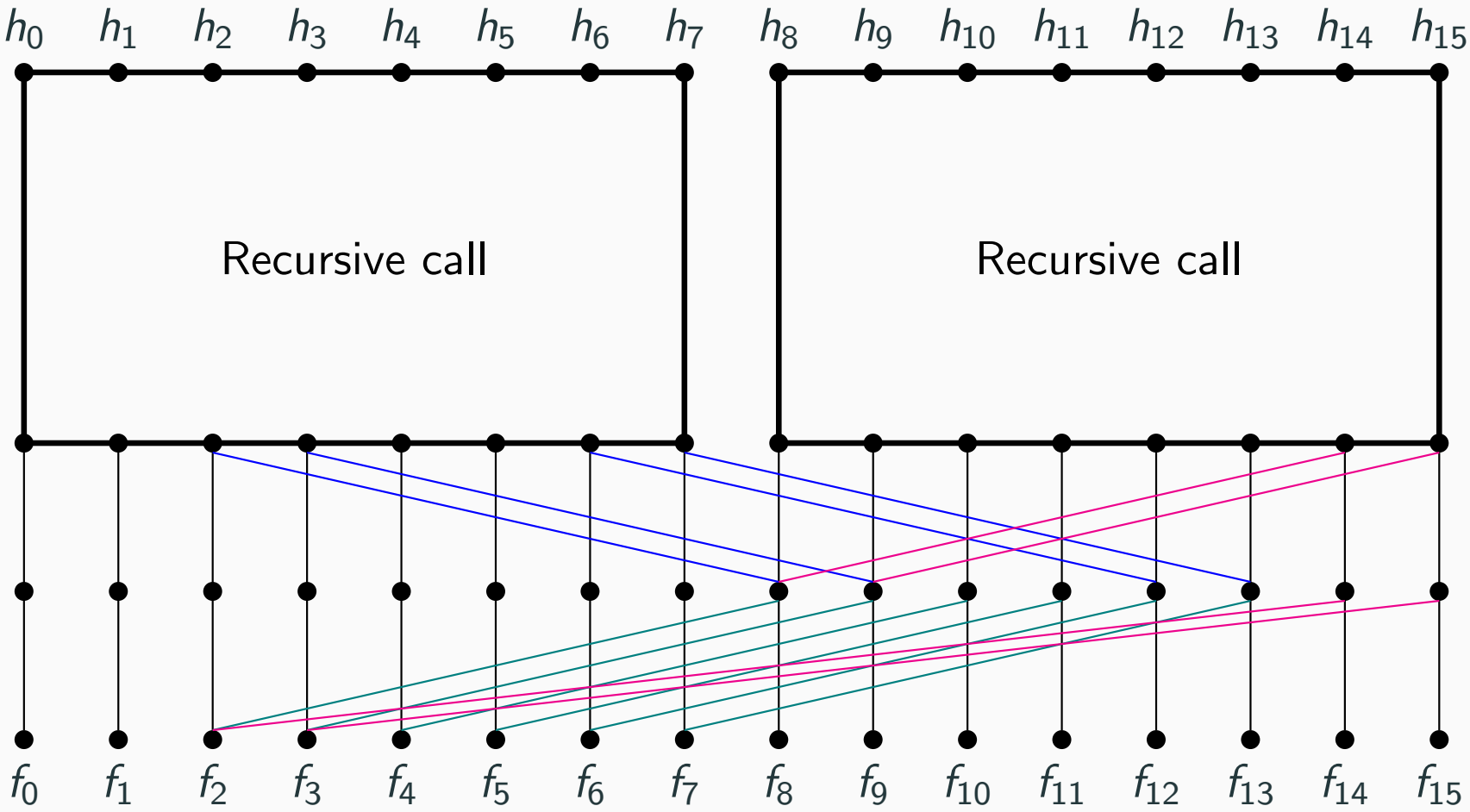
Interpolation example: $q = 4$, $n = 2$

Let $F = \sum_{i=0}^{15} f_i x^i$ and $\mathrm{H}_2(F) = (h_0, \ldots, h_{15})$.



Interpolation example: $q = 4$, $n = 2$

Let $F = \sum_{i=0}^{15} f_i x^i$ and $\mathrm{H}_2(F) = (h_0, \ldots, h_{15})$.



Thank you!