Computational Explorations on the Tensor Rank and the Additive Complexity of Finite (Semi)Fields

Jean-Guillaume Dumas Stefano Lia John Sheekey

ISSAC 2026

Contents

Finite extensions in characteristic 2 or 3

Multiplying elements in finite extensions $\mathbb{F}_{q^d}$ or $\mathbb{S}_{q^d}$

Minimal # (base field $\mathbb{F}_q$) operations for the multiplications?

- M = Multiplicative complexity (= tensor rank R)
- A = Additive complexity

Finite extensions in characteristic 2 or 3

Multiplying elements in finite extensions $\mathbb{F}_{q^d}$ or $\mathbb{S}_{q^d}$

Minimal # (base field $\mathbb{F}_q$) operations for the multiplications?

- M = Multiplicative complexity (= tensor rank R)
- A = Additive complexity

NP-hardness: minimization strategy

- Lower bounds: algebraic constraints & exhaustive exploration
- Upper bounds: random walks & optimization of straight-line programs

Example: degree 2 extension via Karatsuba Multiplication

$$\begin{aligned} c_0 + c_1 X + c_2 X^2 &= (a_0 + a_1 X)(b_0 + b_1 X) \\ &= a_0 b_0 + (a_0 b_1 + a_1 b_0) X + a_1 b_1 X^2 \\ &= a_0 b_0 + (a_0 b_0 + (a_0 - a_1)(b_1 - b_0) + a_1 b_1) X + a_1 b_1 X^2 \end{aligned}$$

Example (Multiplication of $d^\circ 1$ polynomials)

Algorithm	Multiplications	Additions
Classical	4 M	1 A
Karatsuba	3 M	4 A

Contents

Finite Semifields

Definition (Finite Field)

Isomorphic to $\mathbb{F}_p[X]/f(X)$ with irreducible f of degree d

Finite Semifields

Definition (Finite Field)

Isomorphic to $\mathbb{F}_p[X]/f(X)$ with irreducible f of degree d

Definition (Finite Semifield)

A finite semifield $\mathbb{S}$ is a finite division algebra, where multiplication is not assumed to be commutative or associative

Finite Semifield example

Let q be a power of an odd prime p, $k \geq 2$, and α a non-square in $\mathbb{F}_{q^k}$.

Finite Semifield example

Let q be a power of an odd prime p, $k \geq 2$, and α a non-square in $\mathbb{F}_{q^k}$.

Example (Finite field as a degree-2 extension)

Then $(a_0 + a_1 X)(b_0 + b_1 X)$ in the finite field $\mathbb{F}_{q^{2k}} \simeq \mathbb{F}_{q^k}[X]/(X^2 - \alpha)$ is:

$$\begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} a_0 b_0 + \alpha\, a_1 b_1 \\ a_0 b_1 + a_1 b_0 \end{bmatrix}$$

Finite Semifield example

Let q be a power of an odd prime p, $k \geq 2$, and α a non-square in $\mathbb{F}_{q^k}$.

Example (Finite field as a degree-2 extension)

Then $(a_0 + a_1 X)(b_0 + b_1 X)$ in the finite field $\mathbb{F}_{q^{2k}} \simeq \mathbb{F}_{q^k}[X]/(X^2 - \alpha)$ is:

$$\begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} a_0 b_0 + \alpha\, a_1 b_1 \\ a_0 b_1 + a_1 b_0 \end{bmatrix}$$

Theorem (Dickson 1906)

The following defines a semifield multiplication, not equivalent to the finite field:

$$\begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} a_0 b_0 + \alpha\, a_1^q b_1 \\ a_0 b_1 + a_1 b_0 \end{bmatrix}$$

Finite Semifield example

Let q be a power of an odd prime p, $k \geq 2$, and α a non-square in $\mathbb{F}_{q^k}$.

Example (Finite field as a degree-2 extension)

Then $(a_0 + a_1 X)(b_0 + b_1 X)$ in the finite field $\mathbb{F}_{q^{2k}} \simeq \mathbb{F}_{q^k}[X]/(X^2 - \alpha)$ is:

$$\begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} a_0 b_0 + \alpha\, a_1 b_1 \\ a_0 b_1 + a_1 b_0 \end{bmatrix}$$

Theorem (Dickson 1906)

The following defines a semifield multiplication, not equivalent to the finite field:

$$\begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} a_0 b_0 + \alpha\, a_1^q b_1 \\ a_0 b_1 + a_1 b_0 \end{bmatrix}$$

non-associative: $(\vec{x}\vec{y})\,\vec{z} = \begin{bmatrix} \ldots^q \ldots^q \ldots \\ \alpha x_1^q y_1 z_1 + \ldots \end{bmatrix}$ versus $\vec{x}\,(\vec{y}\vec{z}) = \begin{bmatrix} \ldots^q \ldots^q \ldots \\ \ldots + x_1 \alpha y_1^q z_1 \end{bmatrix}$

Contents

Bilinear algorithms and LRP matrix representation

Definition (LRP matrix representation of bilinear algorithms)

Linear combinations of the Left input, of the Right input, and Products to form the output:

$$\vec{c} = P \cdot (L \cdot \vec{a}) \odot (R \cdot \vec{b}) \text{ with } P \in \mathbb{F}^{k \times r}, L \in \mathbb{F}^{r \times m}, R \in \mathbb{F}^{r \times n}$$

with $\odot$ the Hadamard (entry-wise) product of vectors

Bilinear algorithms and LRP matrix representation

Definition (LRP matrix representation of bilinear algorithms)

Linear combinations of the Left input, of the Right input, and Products to form the output:

$$\vec{c} = P \cdot (L \cdot \vec{a}) \odot (R \cdot \vec{b}) \text{ with } P \in \mathbb{F}^{k \times r}, L \in \mathbb{F}^{r \times m}, R \in \mathbb{F}^{r \times n}$$

with $\odot$ the Hadamard (entry-wise) product of vectors

Example (Rank-3 polynomial multiplication of Karatsuba)

$$c_0 + c_1 X + c_2 X^2 = a_0 b_0 + (a_0 b_0 + (a_0 - a_1)(b_1 - b_0) + a_1 b_1) X + a_1 b_1 X^2$$

$$\begin{bmatrix} a_0 b_0 \\ a_0 b_1 + a_1 b_0 \\ a_1 b_1 \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}}_{P} \left(\underbrace{\begin{bmatrix} 1 & 0 \\ 1 & -1 \\ 0 & 1 \end{bmatrix}}_{L} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \odot \underbrace{\begin{bmatrix} 1 & 0 \\ -1 & 1 \\ 0 & 1 \end{bmatrix}}_{R} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \right)$$

Bilinear algorithms and Straight-Line Programs (SLP)

$$\vec{c} = P \cdot (L \cdot \vec{a}) \odot (R \cdot \vec{b}) \text{ with } P \in \mathbb{F}^{k \times r}, L \in \mathbb{F}^{r \times m}, R \in \mathbb{F}^{r \times n}$$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}}_{P} \left(\underbrace{\begin{bmatrix} 1 & 0 \\ 1 & -1 \\ 0 & 1 \end{bmatrix}}_{L} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \odot \underbrace{\begin{bmatrix} 1 & 0 \\ -1 & 1 \\ 0 & 1 \end{bmatrix}}_{R} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \right)$$

Bilinear algorithms and Straight-Line Programs (SLP)

$$\vec{c} = P \cdot (L \cdot \vec{a}) \odot (R \cdot \vec{b}) \text{ with } P \in \mathbb{F}^{k \times r}, L \in \mathbb{F}^{r \times m}, R \in \mathbb{F}^{r \times n}$$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}}_{P} \left(\underbrace{\begin{bmatrix} 1 & 0 \\ 1 & -1 \\ 0 & 1 \end{bmatrix}}_{L} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \odot \underbrace{\begin{bmatrix} 1 & 0 \\ -1 & 1 \\ 0 & 1 \end{bmatrix}}_{R} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \right)$$

Example (Straight-Line Program)

```
l1:=a0+a1; r1:=b1-b0;
c0:=a0*b0; p1:=l1*r1; c2:=a1*b1;
c1:=c0+p1+c2;
```

Bilinear algorithms and Straight-Line Programs (SLP)

$$\vec{c} = P \cdot (L \cdot \vec{a}) \odot (R \cdot \vec{b}) \text{ with } P \in \mathbb{F}^{k \times r}, L \in \mathbb{F}^{r \times m}, R \in \mathbb{F}^{r \times n}$$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}}_{P} \left(\underbrace{\begin{bmatrix} 1 & 0 \\ 1 & -1 \\ 0 & 1 \end{bmatrix}}_{L} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \odot \underbrace{\begin{bmatrix} 1 & 0 \\ -1 & 1 \\ 0 & 1 \end{bmatrix}}_{R} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \right)$$

Example (Straight-Line Program)

```
l1:=a0+a1; r1:=b1-b0;
c0:=a0*b0; p1:=l1*r1; c2:=a1*b1;
c1:=c0+p1+c2;
```

Straight-Line Programs

- Multiplicative complexity = common dimension r

Bilinear algorithms and Straight-Line Programs (SLP)

$$\vec{c} = P \cdot (L \cdot \vec{a}) \odot (R \cdot \vec{b}) \text{ with } P \in \mathbb{F}^{k \times r}, L \in \mathbb{F}^{r \times m}, R \in \mathbb{F}^{r \times n}$$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}}_{P} \left(\underbrace{\begin{bmatrix} 1 & 0 \\ 1 & -1 \\ 0 & 1 \end{bmatrix}}_{L} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \odot \underbrace{\begin{bmatrix} 1 & 0 \\ -1 & 1 \\ 0 & 1 \end{bmatrix}}_{R} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \right)$$

Example (Straight-Line Program)

```
l1:=a0+a1; r1:=b1-b0;
c0:=a0*b0; p1:=l1*r1; c2:=a1*b1;
c1:=c0+p1+c2;
```

Straight-Line Programs

- Multiplicative complexity = common dimension r
- Additive complexity = Shortest Straight-Line Program (SSLP) for L, R and P

Bilinear algorithms and Straight-Line Programs (SLP)

$$\vec{c} = P \cdot (L \cdot \vec{a}) \odot (R \cdot \vec{b}) \text{ with } P \in \mathbb{F}^{k \times r}, L \in \mathbb{F}^{r \times m}, R \in \mathbb{F}^{r \times n}$$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}}_{P} \left(\underbrace{\begin{bmatrix} 1 & 0 \\ 1 & -1 \\ 0 & 1 \end{bmatrix}}_{L} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \odot \underbrace{\begin{bmatrix} 1 & 0 \\ -1 & 1 \\ 0 & 1 \end{bmatrix}}_{R} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \right)$$

Example (Straight-Line Program)

```
l1:=a0+a1; r1:=b1-b0;
c0:=a0*b0; p1:=l1*r1; c2:=a1*b1;
c1:=c0+p1+c2;
```

Straight-Line Programs

- Multiplicative complexity = common dimension r
- Additive complexity = Shortest Straight-Line Program (SSLP) for L, R and P
  - includes the scalar multiplications by (base field) constants (if char. $\notin \{2, 3\}$)

Contents

Known tensor ranks over $\mathbb{F}_2$ and $\mathbb{F}_3$

Set	t. rank	reference
$\mathbb{F}_{2^2}$, $\mathbb{F}_{3^2}$	3	[Karatsuba 1963,Lempel-Seroussi-Winograd 1983]
$\mathbb{F}_{2^3}$, $\mathbb{F}_{3^3}$	6	[Lempel-Seroussi-Winograd 1983]
$\mathbb{F}_{2^4}$	9	[Karatsuba 1963,Chudnovsky-Chudnovsky 1988]
$\mathbb{S}_{2^4}$ #2, #3	9	[Lavrauw-Sheekey 2022]
$\mathbb{F}_{3^4}$,$\mathbb{S}_{3^4}$ #9	9	[Karatsuba 1963,Lavrauw 2013]
$\mathbb{S}_{3^4}$ #1,...,#8,#10,#11	8	[Lavrauw-Sheekey 2022]
$\mathbb{F}_{2^5} = \mathbb{S}_{2^5}$ #5	13	[Montgomery 2005]
$\mathbb{S}_{2^5}$ #1,...,#4,#6	13	Here
$\mathbb{F}_{3^5}$	11	Here
$\mathbb{S}_{3^5}$ #2, #4	10	Here
$\mathbb{S}_{3^5}$ #3, #5,#6	11	Here
$\mathbb{S}_{3^5}$ #7,...,#12	$\geq$11	Here
$\mathbb{F}_{2^6}$	15	
$\mathbb{F}_{3^6}$	$\in [12, 15]$	CodeTables.de, [Cenk-Özbudak 2010]
$\mathbb{F}_{2^7}$	$\in [18, 22]$	
$\mathbb{F}_{2^8}$	$\in [20, 24]$	

Line of proof

1. Exhaustive search for tensor rank 1, 2, 3, ...

Line of proof

1. Exhaustive search for tensor rank 1, 2, 3, ...
2.  *[Barbulescu-Detrey-Estibals-Zimmermann 2012]* reduces the search into linear subspaces

Line of proof

1. Exhaustive search for tensor rank 1, 2, 3, ...
2. *[Barbulescu-Detrey-Estibals-Zimmermann 2012]* reduces the search into linear subspaces
3. here: further restricts the search using (semi)fields correcting codes properties, e.g.:

Line of proof

1. Exhaustive search for tensor rank 1, 2, 3, ...
2. *[Barbulescu-Detrey-Estibals-Zimmermann 2012]* reduces the search into linear subspaces
3. here: further restricts the search using (semi)fields correcting codes properties, e.g.:

Theorem (Maximum Rank Distance = Semifield Spread Set  [Rúa-Combarro-Ranilla-2009])

an LRP *matrix representation defines a (semi)field multiplication of order k and rank r iff span* $\langle \sum_{i=1}^{r} P_{ji}(\mathsf{L_i}^{\mathsf{T}} \cdot R_i) : j \in [1,k] \rangle$ *is k-dimensional and every non-zero is invertible.*

Line of proof

1. Exhaustive search for tensor rank 1, 2, 3, ...
2. *[Barbulescu-Detrey-Estibals-Zimmermann 2012]* reduces the search into linear subspaces
3. here: further restricts the search using (semi)fields correcting codes properties, e.g.:

Theorem (Maximum Rank Distance = Semifield Spread Set [Rúa-Combarro-Ranilla-2009])

an LRP *matrix representation defines a (semi)field multiplication of order k and rank r iff span $\langle \sum_{i=1}^{r} P_{ji}(\mathsf{L}_i^{\mathsf{T}} \cdot R_i) : j \in [1,k] \rangle$ is k-dimensional and every non-zero is invertible.*

Theorem (Lempel-Winograd 1977, Brockett-Dobkin 1978, Lavrauw-Sheekey 2012)

The rowspace of each of L^{T}, R^{T}, P, *of the* LRP *matrix representation of the multiplication in a (semi)field, is an $[r, k, d \geq k]_q$ error-correcting linear code.*

Line of proof

1. Exhaustive search for tensor rank 1, 2, 3, ...
2. *[Barbulescu-Detrey-Estibals-Zimmermann 2012]* reduces the search into linear subspaces
3. here: further restricts the search using (semi)fields correcting codes properties, e.g.:

Theorem (Maximum Rank Distance = Semifield Spread Set [Rúa-Combarro-Ranilla-2009])

an LRP *matrix representation defines a (semi)field multiplication of order k and rank r iff span* $\langle \sum_{i=1}^{r} P_{ji}(\mathsf{L}_i^{\mathsf{T}} \cdot R_i) : j \in [1,k] \rangle$ *is k-dimensional and every non-zero is invertible.*

Theorem (Lempel-Winograd 1977, Brockett-Dobkin 1978, Lavrauw-Sheekey 2012)

The rowspace of each of L^{T}, R^{T}, P, *of the* LRP *matrix representation of the multiplication in a (semi)field, is an* $[r, k, d \geq k]_q$ *error-correcting linear code.*

github.com/jsheekey/tensor_semifield

Contents

Straight-Line Programs (SLP) for linear codes

Definition (SSLP: shortest straight-line program)

$\mathrm{SSLP}_q[r,k,d]$: minimum number of additions (and scalar multiplications) required to compute the matrix-vector product Lv, where $L^\intercal$ generates an $[r,k,d]_q$ Hamming metric linear code.

Straight-Line Programs (SLP) for linear codes

Definition (SSLP: shortest straight-line program)

$\mathrm{SSLP}_q[r,k,d]$: minimum number of additions (and scalar multiplications) required to compute the matrix-vector product Lv, where $L^\intercal$ generates an $[r,k,d]_q$ Hamming metric linear code.

Lemma (Additive complexity lower bound for the multiplication in a (semi)field)

The minimum number of operations needed to compute multiplication in a field or semifield of degree k over $\mathbb{F}_q$ using r multiplications is at least $r\ M + (3\mathrm{SSLP}_q[r,k,k] + r - k)\ A$.

Straight-Line Programs (SLP) for linear codes

Definition (SSLP: shortest straight-line program)

$\mathrm{SSLP}_q[r,k,d]$: minimum number of additions (and scalar multiplications) required to compute the matrix-vector product Lv, where L^T generates an $[r,k,d]_q$ Hamming metric linear code.

Lemma (Additive complexity lower bound for the multiplication in a (semi)field)

The minimum number of operations needed to compute multiplication in a field or semifield of degree k over $\mathbb{F}_q$ using r multiplications is at least $r\ M + (3\mathrm{SSLP}_q[r,k,k] + r - k)\ A$.

Proof.

By *[Brockett-Dobkin 1978,Lavrauw-Sheekey 2012]*, $\mathsf{L}^\mathsf{T} \in \mathbb{F}^{k\times r}$, $\mathsf{R}^\mathsf{T} \in \mathbb{F}^{k\times r}$ and $P \in \mathbb{F}^{k\times r}$ are $[r,k,\geq k]_q$ codes ; then use the *tranposition* principle on P^T. □

Some shortest straight-line programs for linear codes

- Lower bounds: hamming-metric distance constraints

Some shortest straight-line programs for linear codes

- Lower bounds: hamming-metric distance constraints
- Upper bounds: straight-line programs optimizations (github.com/jgdumas/plinopt)

Some shortest straight-line programs for linear codes

- Lower bounds: hamming-metric distance constraints
- Upper bounds: straight-line programs optimizations (github.com/jgdumas/plinopt)

Example ($[9, 4, 4]$ codes)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

Some shortest straight-line programs for linear codes

- Lower bounds: hamming-metric distance constraints
- Upper bounds: straight-line programs optimizations (github.com/jgdumas/plinopt)

Example ($[9, 4, 4]$ codes)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

```
o0:=i0;
o1:=i1;
o2:=i2;
o3:=i3;
o4:=i0;
o5:=i1+i2;
o6:=i3+o5;
t4:=i0+i3;
o7:=i2+t4;
o8:=i1+t4;
```

Some shortest straight-line programs for linear codes

$[r,k,d]_q$	SSLP
$[8,4,4]_2$	6
$[9,4,4]_2$	5
$[10,4,4]_2$	4
$[13,5,5]_2$	8
$[8,4,4]_3$	6
$[9,4,4]_3$	5
$[10,4,4]_3$	4
$[10,5,5]_3$	$\in [8,10]$
$[11,5,5]_3$	≤ 12
$[12,5,5]_3$	≤ 12
$[13,5,5]_3$	≤ 11

- Lower bounds: hamming-metric distance constraints
- Upper bounds: straight-line programs optimizations (github.com/jgdumas/plinopt)

Example ($[9,4,4]$ codes)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

```
o0:=i0;
o1:=i1;
o2:=i2;
o3:=i3;
o4:=i0;
o5:=i1+i2;
o6:=i3+o5;
t4:=i0+i3;
o7:=i2+t4;
o8:=i1+t4;
```

Generic Straight-Line Programs optimizations: PLinOpt

1. Common sub-expression elimination: $\begin{cases} o_0 := i_0 + i_1; \\ o_1 := i_0 + i_1; \end{cases} \Rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \Leftarrow$

```
t:=i0+i1;
o0:=t;
o1:=t;
```

Generic Straight-Line Programs optimizations: PLinOpt

1. Common sub-expression elimination: $\begin{cases} o_0 := i_0 + i_1; \\ o_1 := i_0 + i_1; \end{cases} \Rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \Leftarrow$
```
t:=i0+i1;
o0:=t;
o1:=t;
```
2. Common scalar elimination: $\{o_0 := i_0 * 2 + i_1 * 2 \Rightarrow \begin{bmatrix} 2 & 2 \end{bmatrix} \Leftarrow$
```
t:=i0+i1;
o0:=t*2;
```

Generic Straight-Line Programs optimizations: PLinOpt

1. Common sub-expression elimination: $\begin{cases} o_0 := i_0 + i_1; \\ o_1 := i_0 + i_1; \end{cases} \Rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \Leftarrow$
```
t:=i0+i1;
o0:=t;
o1:=t;
```
2. Common scalar elimination: $\{o_0 := i_0 * 2 + i_1 * 2 \Rightarrow \begin{bmatrix} 2 & 2 \end{bmatrix} \Leftarrow$
```
t:=i0+i1;
o0:=t*2;
```
3. Linear dependencies: $\begin{cases} o_0 := (i_0 + i_1)/2; \\ o_1 := (i_0 - i_1)/2; \end{cases} \Rightarrow \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix}$

Generic Straight-Line Programs optimizations: PLinOpt

1. Common sub-expression elimination: $\begin{cases} o_0 := i_0 + i_1; \\ o_1 := i_0 + i_1; \end{cases} \Rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \Leftarrow$
```
t:=i0+i1;
o0:=t;
o1:=t;
```
2. Common scalar elimination: $\{o_0 := i_0 * 2 + i_1 * 2 \Rightarrow \begin{bmatrix} 2 & 2 \end{bmatrix} \Leftarrow$
```
t:=i0+i1;
o0:=t*2;
```
3. Linear dependencies: $\begin{cases} o_0 := (i_0 + i_1)/2; \\ o_1 := (i_0 - i_1)/2; \end{cases} \Rightarrow \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix} \Leftarrow$
```
o0:=(i0+i1)/2;
o1:=o0-i1;
```

Generic Straight-Line Programs optimizations: PLinOpt

1. Common sub-expression elimination: $\begin{cases} o_0 := i_0 + i_1; \\ o_1 := i_0 + i_1; \end{cases} \Rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \Leftarrow$
```
t:=i0+i1;
o0:=t;
o1:=t;
```
2. Common scalar elimination: $\{o_0 := i_0 * 2 + i_1 * 2 \Rightarrow \begin{bmatrix} 2 & 2 \end{bmatrix} \Leftarrow$
```
t:=i0+i1;
o0:=t*2;
```
3. Linear dependencies: $\begin{cases} o_0 := (i_0 + i_1)/2; \\ o_1 := (i_0 - i_1)/2; \end{cases} \Rightarrow \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix} \Leftarrow$
```
o0:=(i0+i1)/2;
o1:=o0-i1;
```

[Boyar-Peralta-Matthews 2008]: optimal SLP is NP-hard

Generic Straight-Line Programs optimizations: PLinOpt

1. Common sub-expression elimination: $\begin{cases} o_0 := i_0 + i_1; \\ o_1 := i_0 + i_1; \end{cases} \Rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \Leftarrow$
```
t:=i0+i1;
o0:=t;
o1:=t;
```
2. Common scalar elimination: $\{o_0 := i_0 * 2 + i_1 * 2 \Rightarrow \begin{bmatrix} 2 & 2 \end{bmatrix} \Leftarrow$
```
t:=i0+i1;
o0:=t*2;
```
3. Linear dependencies: $\begin{cases} o_0 := (i_0 + i_1)/2; \\ o_1 := (i_0 - i_1)/2; \end{cases} \Rightarrow \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix} \Leftarrow$
```
o0:=(i0+i1)/2;
o1:=o0-i1;
```

- *[Boyar-Peralta-Matthews 2008]*: optimal SLP is NP-hard
- *[Morgenstern 1975]*: have to explore all minors of L, R, P ...
  - ⇨ 1: 2×2 minors
  - ⇨ 3: maximal minors (*[D-Pernet-Sedoglavic 2024]* & "Randomized Kernel Decomposition")

(Aparté) examples of SLP additive complexity optimization via PLinOpt

Degree 4 polynomial multiplication (over $\mathbb{Q}$)

	MUL	ADD	SCA	Total here	*[El Mrabet-Guillevic-Ionica 2011]*
Standard	25	0+0+16	-	25M+16A	25M+16A
[Montg.]	13	11+11+27	0+0+1	13M+50A	13M+62A
[Toom-5]	9	19+19+31	8+8+18	9M+103A	9M+124A

(Aparté) examples of SLP additive complexity optimization via PLinOpt

Degree 4 polynomial multiplication (over $\mathbb{Q}$)

	MUL	ADD	SCA	Total here	*[El Mrabet-Guillevic-Ionica 2011]*
Standard	25	0+0+16	-	25M+16A	25M+16A
[Montg.]	13	11+11+27	0+0+1	13M+50A	13M+62A
[Toom-5]	9	19+19+31	8+8+18	9M+103A	9M+124A

Additive upper bounds for Toom-Cook interpolation (Vandermonde) P matrix (over $\mathbb{Q}$)

P matrix	PLinOpt	*[Bodrato-Zanoni 2007]*
[Toom-3]	8A+4S	8A+4S
[Toom-3.5]	11A+6S	12A+6S
[Toom-4]	17A+11S	18A+11S
[Toom-4.5]	22A+12S	22A+14S
[Toom-5]	31A+18S	32A+21S

Folding for multiplication in extensions

 Do not: [multiply; then reduce] 

Folding for multiplication in extensions

⚠ Do not: [multiply; then reduce] ⚠

👍 Instead: pre-compute modular multiplications LRP matrices, with the given irreducible ("Folding of the modular reduction")

Folding for multiplication in extensions

⚠ Do not: [multiply; then reduce] ⚠

👍 Instead: pre-compute modular multiplications LRP matrices, with the given irreducible ("Folding of the modular reduction")

Example

⚠ $$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}}_{P} \begin{bmatrix} a_0 b_0 \\ (a_0 - a_1)(b_1 - b_0) \\ a_1 b_1 \end{bmatrix}$$, then $(c_0 + c_1 X + c_2 X^2 \mod f(X))$ ...

Folding for multiplication in extensions

⚠ Do not: [multiply; then reduce] ⚠

👍 Instead: pre-compute modular multiplications LRP matrices, with the given irreducible ("Folding of the modular reduction")

Example

⚠ $$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}}_{P} \begin{bmatrix} a_0 b_0 \\ (a_0 - a_1)(b_1 - b_0) \\ a_1 b_1 \end{bmatrix}, \text{ then } (c_0 + c_1 X + c_2 X^2 \mod f(X)) \ldots$$

👍 mod $f(X) = X^2 + \alpha X + \beta$: $P_{fold} = \begin{bmatrix} 1 & 0 & -\beta \\ 1 & 1 & 1-\alpha \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} - \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$

Folding for multiplication in extensions

⚠ Do not: [multiply; then reduce] ⚠

👍 Instead: pre-compute modular multiplications LRP matrices, with the given irreducible ("Folding of the modular reduction")

Example

👍 mod $f(X) = X^2 + \alpha X + \beta$: $P_{fold} = \begin{bmatrix} 1 & 0 & -\beta \\ 1 & 1 & 1-\alpha \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} - \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$

☺ Select the irreducible with the smallest folded SLP (additive complexity)

Fields and semifields of order 32

Folding for multiplication in degree-5 extension in characteristic 2

		MUL	ADD	Total
deg. 4 poly. mul. Standard mod 2		25	0+0+16	25M+16A
deg. 4 poly. mul. *[Montgomery]* mod 2		13	9+9+19	13M+37A
$\mathbb{F}_{2^5}$ Standard mod X^5+X^2+1	here	25	0+0+24	25M+24A
$\mathbb{F}_{2^5}$ via *[Montgomery]* mod $X^5+X^4+X^2+X+1$	here	13	9+9+18	13M+36A
$\mathbb{S}_{2^5}$ #2	here	13	12+9+18	13M+39A

Fields and semifields of order 32

Folding for multiplication in degree-5 extension in characteristic 2

		MUL	ADD	Total
deg. 4 poly. mul. Standard mod 2		25	0+0+16	25M+16A
deg. 4 poly. mul. *[Montgomery]* mod 2		13	9+9+19	13M+37A
$\mathbb{F}_{2^5}$ Standard mod X^5+X^2+1	here	25	0+0+24	25M+24A
$\mathbb{F}_{2^5}$ via *[Montgomery]* mod $X^5+X^4+X^2+X+1$	here	13	9+9+18	13M+36A
$\mathbb{S}_{2^5}$ #2	here	13	12+9+18	13M+39A

Optimal Normal Basis: $15M + 40A$ *[Reyhani-Hasan 2003]*

Fields and semifields of order 32

Folding for multiplication in degree-5 extension in characteristic 2

		MUL	ADD	Total
deg. 4 poly. mul. Standard mod 2		25	0+0+16	25M+16A
deg. 4 poly. mul. *[Montgomery]* mod 2		13	9+9+19	13M+37A
$\mathbb{F}_{2^5}$ Standard mod X^5+X^2+1	here	25	0+0+24	25M+24A
$\mathbb{F}_{2^5}$ via *[Montgomery]* mod $X^5+X^4+X^2+X+1$	here	13	9+9+18	13M+36A
$\mathbb{S}_{2^5}$ #2	here	13	12+9+18	13M+39A

Lower bound:

✗ $13M + (3\mathrm{SSLP}_2[13,5,5] + 13 - 5)A = 13M + (3*8 + 13 - 5)A = 13M + 32A$

Fields and semifields of order 81

Degree 4 folding in characteristic 3

		MUL	ADD	Total
$\mathbb{F}_{3^4}$ standard mod X^4+X-1		16	0+0+15	16M+15A
$\mathbb{F}_{3^4}$ Karatsuba$^{\otimes^2}$ mod $1+X+X^2+X^3+X^4$	here	9	5+5+11	9M+21A
$\mathbb{S}_{3^4}$	here	8	6+6+10	8M+22A

Fields and semifields of order 81

Degree 4 folding in characteristic 3

		MUL	ADD	Total
$\mathbb{F}_{3^4}$ standard mod X^4+X-1		16	0+0+15	16M+15A
$\mathbb{F}_{3^4}$ Karatsuba$^{\otimes^2}$ mod $1+X+X^2+X^3+X^4$	here	9	5+5+11	9M+21A
$\mathbb{S}_{3^4}$	here	8	6+6+10	8M+22A

Lower bounds:

- ✗ $9M + (3\mathrm{SSLP}_3[9,4,4] + 9 - 4)A = 9M + (3*5+9-4)A = 9M + 20A$
- ✓ $8M + (3\mathrm{SSLP}_3[8,4,4] + 8 - 4)A = 8M + (3*6+8-4)A = 8M + 22A$

Fields and semifields of order 243

Degree 5 folding in characteristic 3

		MUL	ADD	Total
$\mathbb{F}_{3^5}$ standard mod $X^5 - X + 1$		25	0+0+24	25M+24A
$\mathbb{F}_{3^5}$ *[Montgomery]* mod $X^5 + X^4 - X^3 - X^2 - 1$		13	11+11+20	13M+42A
$\mathbb{F}_{3^5}$ mod $X^5 - X^4 + X^3 + X^2 + X + 1$	here	12	14+14+19	12M+47A
$\mathbb{F}_{3^5}$ mod $X^5 - X + 1$	here	11	12+12+20	11M+44A
$\mathbb{S}_{3^5}$	here	10	13+13+17	10M+43A

Fields and semifields of order 243

Degree 5 folding in characteristic 3

		MUL	ADD	Total
$\mathbb{F}_{3^5}$ standard mod $X^5 - X + 1$		25	0+0+24	25M+24A
$\mathbb{F}_{3^5}$ *[Montgomery]* mod $X^5 + X^4 - X^3 - X^2 - 1$		13	11+11+20	13M+42A
$\mathbb{F}_{3^5}$ mod $X^5 - X^4 + X^3 + X^2 + X + 1$	here	12	14+14+19	12M+47A
$\mathbb{F}_{3^5}$ mod $X^5 - X + 1$	here	11	12+12+20	11M+44A
$\mathbb{S}_{3^5}$	here	10	13+13+17	10M+43A

Lower bounds:

- ✗ $11M + (3\mathrm{SSLP}_3[11,5,5] + 11 - 5)A \geq 11M + (3 * 12 + 11 - 5) = 11M + 42A$
- ✗ $10M + (3\mathrm{SSLP}_3[10,5,5] + 10 - 5)A \geq 10M + (3 * 10 + 10 - 5) = 10M + 35A$

Open problems

- Tensor rank in extensions of degrees 6, 7, 8, ...
- General lower bounds for $\mathrm{SSLP}_q(r, k, \geq k)$?

Open problems

- Tensor rank in extensions of degrees 6, 7, 8, ...
- General lower bounds for $\mathrm{SSLP}_q(r, k, \geq k)$?
- Improve SLP optimization?
- Specific improvements?
  - Multiplication in finite (semi)fields of order $3^4 = 81$ with only $9M + 20A$ operations?
  - Multiplication in finite (semi)fields of order $3^5 = 243$ with:
    - $10M$ and fewer than $43A$ operations?
    - $11M$ and fewer than $44A$ operations?