On the scalar complexity of Chudnovsky multiplication algorithm in finite fields

Stéphane Ballet, Alexis Bonnecaze, Thanh-Hung Dang

On the scalar complexity of Chudnovsky^2 multiplication algorithm in finite fields

Stéphane Ballet[1], Alexis Bonnecaze[2], and Thanh-Hung Dang[3]

[1] Aix Marseille Univ, CNRS, Centrale Marseille, I2M, Marseille, France.
Institut de Mathématiques de Marseille, UMR 7373, CNRS,
Aix-Marseille Université, case 930, F13288 Marseille cedex 9, France
stephane.ballet@univ-amu.fr
[2] alexis.bonnecaze@univ-amu.fr
[3] thanh-hung.dang@etu.univ-amu.fr

Abstract. We propose a new construction for the multiplication algorithm of D.V. and G.V. Chudnovsky in order to improve scalar algebraic complexity. In particular, we improve the Baum-Shokrollahi construction for multiplication in $\mathbb{F}_{256}/\mathbb{F}_4$ based on the elliptic Fermat curve $x^3 + y^3 = 1$.

Keywords: Finite field, Algebraic function field, Algebraic complexity.

1 Introduction

We are interested by the multiplicative complexity of multiplication in a finite field $\mathbb{F}_{q^n}$, i.e. by the number of multiplications required to multiply in the $\mathbb{F}_q$-vector space $\mathbb{F}_{q^n}$ of dimension n. There exist two types of multiplications in $\mathbb{F}_q$: the scalar multiplication and the bilinear one. The scalar multiplication is the multiplication by a constant (in $\mathbb{F}_q$). The bilinear multiplication is a multiplication that depends on the elements of $\mathbb{F}_{q^n}$ that are multiplied. The bilinear complexity is independent of the chosen representation of the finite field.

Definition 1. *The total number of scalar multiplications in $\mathbb{F}_q$ used in an algorithm $\mathcal{U}$ of multiplication in $\mathbb{F}_{q^n}$ is called scalar complexity and denoted $\mu_s(\mathcal{U})$.*

More precisely, the multiplication of two elements of $\mathbb{F}_{q^n}$ is an $\mathbb{F}_q$-bilinear map from $\mathbb{F}_{q^n} \times \mathbb{F}_{q^n}$ onto $\mathbb{F}_{q^n}$. Then, it can be considered as an $\mathbb{F}_q$-linear map from the tensor product $\mathbb{F}_{q^n} \otimes_{\mathbb{F}_q} \mathbb{F}_{q^n}$ onto $\mathbb{F}_{q^n}$. Therefore, it can also be considered as an element T of $(\mathbb{F}_{q^n})^\star \otimes_{\mathbb{F}_q} (\mathbb{F}_{q^n})^\star \otimes_{\mathbb{F}_q} \mathbb{F}_{q^n}$, where $\mathbb{F}_{q^n}^\star$ denotes the dual of $\mathbb{F}_{q^n}$.

Set $T = \sum_{i=1}^{r} x_i^\star \otimes y_i^\star \otimes c_i$, where $x_i^\star \in \mathbb{F}_{q^n}^\star$, $y_i^\star \in \mathbb{F}_{q^n}^\star$ and $c_i \in \mathbb{F}_{q^n}$. The following holds for any $x, y \in \mathbb{F}_{q^n}$:

$$x \cdot y = T(x \otimes y) = \sum_{i=1}^{r} x_i^\star(x) y_i^\star(y) c_i.$$

Definition 2. *A bilinear multiplication algorithm $\mathcal{U}$ is an expression*

$$x \cdot y = \sum_{i=1}^{r} x_i^\star(x) y_i^\star(y) c_i,$$

where $x_i^\star, y_i^\star \in (\mathbb{F}_{q^n})^\star$, and $c_i \in \mathbb{F}_{q^n}$. Such an algorithm is said symmetric if $x_i^\star = y_i^\star$ for all i. The number r of summands in this expression is called the bilinear (resp. symmetric bilinear) complexity of the algorithm $\mathcal{U}$ and is denoted by $\mu(\mathcal{U})$ (resp. $\mu^{sym}(\mathcal{U})$).

Definition 3. *The minimal number of summands in a decomposition of the tensor T of the multiplication is called the bilinear (resp. symmetric bilinear) complexity of the multiplication and is denoted by $\mu_q(n)$ (resp. $\mu_q^{sym}(n)$):*

$$\mu_q(n)(\textit{resp. } \mu_q^{sym}(n)) = \min_{\mathcal{U}} \mu(\mathcal{U})(\textit{resp. } \mu^{sym}(\mathcal{U}))$$

where $\mathcal{U}$ is running over all bilinear (resp. symmetric bilinear) multiplication algorithms in $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$.

In their seminal papers, Winograd [11] and De Groote [7] have shown that $\mu_q(n) \geq 2n - 1$, with equality holding if and only if $n \leq \frac{1}{2}q + 1$. Winograd has also proved [11] that optimal multiplication algorithms realizing the lower bound belong to the class of interpolation algorithms. Later, generalizing interpolation algorithms on the projective line over $\mathbb{F}_q$ to algebraic curves of higher genus over $\mathbb{F}_q$, D.V. and G.V. Chudnovsky provided a method [6] which enabled to prove the *linearity* [2] of the bilinear complexity of multiplication in finite extensions of a finite field. This is the so-called Chudnovsky^2 algorithm (or CCMA). Note that the original algorithm CCMA is naturally symmetric.

Several studies focused on the qualitative improvement of CCMA but the problem of its scalar complexity was only addressed in 2015 by Atighehchi, Ballet, Bonnecaze and Rolland [1]. They proposed a new construction which slightly improved the scalar complexity eventhough the main objective of this work was not to optimize scalar complexity. Thus, in the absence of a dedicated strategy to scalar optimization, the number of scalar multiplications has not been significantly reduced in finite distance. Therefore, we note that so far, practical implementations of multiplication algorithms of type Chudnovsky over finite fields have failed to simultaneously optimize the number of scalar multiplications and bilinear multiplications.

Our main goal is to seek an optimal construction of Chudnovsky^2 algorithm in order to optimize its multiplicative complexity. We will consider the elliptic case for which it has been proven that the bilinear complexity of the algorithm is optimal [9]. Therefore, we will focus on optimizing the scalar complexity of this algorithm.

The paper is arranged as follows. Section 2, describes CCMA in the general case. Section 3 proposes a new method of construction with an objective to reduce the scalar complexity of Chudnovsky^2 multiplication algorithms. An optimized basis representation of the Riemann-Roch space $\mathcal{L}(2D)$ is sought in order

to minimize the number of scalar multiplications in the algorithm. Considering the multiplication in $\mathbb{F}_{256}/\mathbb{F}_4$, which is the case study of Baum and Shokrollahi in [4], our strategy leads to improve the scalar complexity of their algorithm.

2 The Chudnovsky^2 multiplication algorithm

2.1 Description and construction of CCMA algorithm

Let $F/\mathbb{F}_q$ be an algebraic function field over the finite field $\mathbb{F}_q$ of genus $g(F)$. We denote by $N_k(F/\mathbb{F}_q)$ the number of places of degree k of F over $\mathbb{F}_q$. If D is a divisor, $\mathcal{L}(D)$ denotes the Riemann-Roch space associated to D. We denote by $\mathcal{O}_Q$ the valuation ring of the place Q and by F_Q its residue class field $\mathcal{O}_Q/Q$ which is isomorphic to $\mathbb{F}_{q^{\deg Q}}$ where $\deg Q$ is the degree of the place Q. The order of a divisor $D = \sum_P a_P P$ in the place P is the number a_P, denoted $ord_P(D)$. The support of a divisor D is the set *supp* D of the places P such that $ord_P(D) \neq 0$. The divisor D is called effective if $ord_P(D) \geq 0$ for any P. Let us define the following Hadamard product in $\mathbb{F}_{q^{l_1}} \times \mathbb{F}_{q^{l_2}} \times \cdots \times \mathbb{F}_{q^{l_N}}$ denoted by $\odot$, where the l_i's denote positive integers, by $(u_1, \ldots, u_N) \odot (v_1, \ldots, v_N) = (u_1 v_1, \ldots, u_N v_N)$. The following theorem describes the original multiplication algorithm of D.V. and G.V. Chudnovsky [6].

Theorem 1. *Let*

- *n be a positive integer,*
- *$F/\mathbb{F}_q$ be an algebraic function field,*
- *Q be a degree n place of $F/\mathbb{F}_q$,*
- *D be a divisor of $F/\mathbb{F}_q$,*
- *$\mathcal{P} = \{P_1, \ldots, P_N\}$ be an ordered set of places of degree one of $F/\mathbb{F}_q$.*

We suppose that supp $D \cap \{Q, P_1, ..., P_N\} = \emptyset$ and that

(i) The evaluation map

$$\begin{aligned} Ev_Q : \mathcal{L}(D) &\to F_Q \\ f &\mapsto f(Q) \end{aligned}$$

is surjective

(ii) The evaluation map

$$\begin{aligned} Ev_{\mathcal{P}} : \mathcal{L}(2D) &\to \mathbb{F}_q^N \\ f &\mapsto \left(f(P_1), \ldots, f(P_N)\right) \end{aligned}$$

is injective

Then

(1) For any two elements x, y in $\mathbb{F}_{q^n}$, we have:

$$xy = E_Q \circ Ev_{\mathcal{P}}|_{Im Ev_{\mathcal{P}}}{}^{-1} \left(E_{\mathcal{P}} \circ Ev_Q^{-1}(x) \odot E_{\mathcal{P}} \circ Ev_Q^{-1}(y) \right), \qquad (1)$$

where E_Q denotes the canonical projection from the valuation ring $\mathcal{O}_Q$ of the place Q in its residue class field F_Q, $E_{\mathcal{P}}$ the extension of $Ev_{\mathcal{P}}$ on the valuation ring $\mathcal{O}_Q$ of the place Q, ${Ev_{\mathcal{P}}|_{ImEv_{\mathcal{P}}}}^{-1}$ the restriction of the inverse map of $Ev_{\mathcal{P}}$ on its image, and $\circ$ the standard composition map.

(2)

$$\mu_q^{sym}(n) \leq N.$$

Since Q is a place of degree n, the residue class field F_Q of place Q is an extension of degree n of $\mathbb{F}_q$ and it therefore can be identified to $\mathbb{F}_{q^n}$. Moreover, the evaluation map Ev_Q being onto, one can associate the elements $x, y \in \mathbb{F}_{q^n}$ with elements of $\mathbb{F}_q$-vector space $\mathcal{L}(D)$, denoted respectively f and g. We define $h := fg$ by

$$(h(P_1), ..., h(P_N)) = E_{\mathcal{P}}(f) \odot E_{\mathcal{P}}(g) = (f(P_1)g(P_1), ..., f(P_N)g(P_N)) . \quad (2)$$

We know that such an element h belongs to $\mathcal{L}(2D)$ since the functions f, g lie in $\mathcal{L}(D)$. Moreover, thanks to injectivity of $Ev_{\mathcal{P}}$, the function h is in $\mathcal{L}(2D)$ and is uniquely determined by (2). We have

$$xy = Ev_Q(f)Ev_Q(g) = E_Q(h)$$

where E_Q is the canonical projection from the valuation ring $\mathcal{O}_Q$ of the place Q in its residue class field F_Q, Ev_Q is the restriction of E_Q over the vector space $\mathcal{L}(D)$.

In order to make the study and the construction of this algorithm easier, we proceed in the following way. We choose a place Q of degree n and a divisor D of degree $n+g-1$, such that Ev_Q and $Ev_{\mathcal{P}}$ are isomorphisms. In this aim in [2], S. Ballet introduces simple numerical conditions on algebraic curves of an arbitrary genus g giving a sufficient condition for the application of the algorithm CCMA (existence of places of certain degree, of non-special divisors of degree $g-1$) generalizing the result of A. Shokrollahi [9] for the elliptic curves. Let us recall this result:

Theorem 2. *Let q be a prime power and let n be an integer > 1. If there exists an algebraic function field $F/\mathbb{F}_q$ of genus g satisfying the conditions*

1. *$N_n > 0$ (which is always the case if $2g+1 \leq q^{\frac{n-1}{2}}(q^{\frac{1}{2}}-1)$),*
2. *$N_1 > 2n + 2g - 2$,*

then there exists a divisor D of degree $n+g-1$ and a place Q such that:

(i) The evaluation map

$$\begin{aligned} Ev_Q : \mathcal{L}(D) &\to \frac{\mathcal{O}_Q}{Q} \\ f &\mapsto f(Q) \end{aligned}$$

is an isomorphism of vector spaces over $\mathbb{F}_q$.

(ii) There exist places P_1,...,P_N such that the evaluation map

$$\begin{aligned} Ev_{\mathcal{P}} : \mathcal{L}(2D) &\to \mathbb{F}_q^N \\ f &\mapsto \left(f(P_1), \ldots, f(P_N)\right) \end{aligned}$$

is an isomorphism of vector spaces over $\mathbb{F}_q$ with $N = 2n + g - 1$.

Remark 1. First, note that in the elliptic case, the condition (2) is a large inequality thanks to a result due to Chaumine [5]. Secondly, note also that the divisor D is not necessarily effective.

By this last remark, it is important to add the property of effectivity for the divisor D in a perspective of implemention. Indeed, it is easier to construct the algorithm CCMA with this assumption because in this case $\mathcal{L}(D) \subseteq \mathcal{L}(2D)$ and we can directly apply the evaluation map $Ev_{\mathcal{P}}$ instead of $E_{\mathcal{P}}$ in the algorithm (1), by means of a suitable representation of $\mathcal{L}(2D)$. Moreover, in this case we need to consider simultaneously the assumption that the support of the divisor D does not contain the rational places and the place Q of degree n and the assumption of effectivity of the divisor D. Indeed, it is known that the support moving technic (cf. [8, Lemma 1.1.4.11]), which is a direct consequence of Strong Approximation Theorem (cf. [10, Proof of Theorem I.6.4]), applied on an effective divisor generates the loss of effectivity of the initial divisor (cf. also [1, Remark 2.2]). So, let us suppose these two last assumptions.

Remark 2. As in [3], in practice, we take as a divisor D one place of degree $n + g - 1$. It has the advantage to solve the problem of the support of divisor D (cf. also [1, Remark 2.2]) as well as the problem of the effectivity of the divisor D. However, it is not required to be considered in the theoretical study, but, as we will see, it will have some importance in the strategy of optimization.

We can therefore consider the basis $\mathcal{B}_Q$ of the residue class field F_Q over $\mathbb{F}_q$ as the image of a basis of $\mathcal{L}(D)$ by Ev_Q or equivalently (which is sometimes useful following the considered situation) the basis of $\mathcal{L}(D)$ as the reciprocal image of a basis of the residue class field F_Q over $\mathbb{F}_q$ by Ev_Q^{-1}. Let

$$\mathcal{B}_D := (f_1, ..., f_n) \tag{3}$$

be a basis of $\mathcal{L}(D)$ and let us denote the basis of the supplementary space $\mathcal{M}$ of $\mathcal{L}(D)$ in $\mathcal{L}(2D)$ by

$$\mathcal{B}_D^c := (f_{n+1}, ..., f_N) \tag{4}$$

where $N := dim\mathcal{L}(2D) = 2n + g - 1$. Then, we choose

$$\mathcal{B}_{2D} := \mathcal{B}_D \cup \mathcal{B}_D^c \tag{5}$$

as the basis of $\mathcal{L}(2D)$.

We denote by T_{2D} the matrix of the isomorphism $Ev_{\mathcal{P}} : \mathcal{L}(2D) \to \mathbb{F}_q^N$ in the basis $\mathcal{B}_{2D}$ of $\mathcal{L}(2D)$ (the basis of $\mathbb{F}_q^N$ will always be the canonical basis).

Then, we denote by T_D the matrix of the first n columns of the matrix T_{2D}. Therefore, T_D is the matrix of the restriction of the evaluation map $Ev_{\mathcal{P}}$ on the Riemann-Roch vector space $\mathcal{L}(D)$, which is an injective morphism.

Note that the canonical surjection E_Q is the extension of the isomorphism Ev_Q since, as $Q \notin supp(D)$, we have $\mathcal{L}(D) \subseteq \mathcal{O}_Q$. Moreover, as $supp(2D) = supp(D)$, we also have $\mathcal{L}(2D) \subseteq \mathcal{O}_Q$. We can therefore consider the images of elements of the basis $\mathcal{B}_{2D}$ by E_Q and obtain a system of N linear equations as follows:

$$E_Q(f_r) = \sum_{m=1}^{n} c_r^m Ev_Q(f_i), \quad r = 1, ..., N$$

where E_Q denotes the canonical projection from the valuation ring $\mathcal{O}_Q$ of the place Q in its residue class field F_Q, Ev_Q is the restriction of E_Q over the vector space $\mathcal{L}(D)$ and $c_r^m \in \mathbb{F}_q$ for $r = 1, ..., N$. Let C be the matrix of the restriction of the map E_Q on the Riemann-Roch vector space $\mathcal{L}(2D)$, from the basis $\mathcal{B}_{2D}$ in the basis $\mathcal{B}_Q$. We obtain the product $z := xy$ of two elements $x, y \in \mathbb{F}_{q^n}$ by the algorithm (1) in Theorem 1, where M^t denotes the transposed matrix of the matrix M:

Algorithm 1 Multiplication algorithm in $\mathbb{F}_{q^n}$

INPUT: $x = \sum_{i=1}^{n} x_i Ev_Q(f_i)$, and $y = \sum_{i=1}^{n} y_i Ev_Q(f_i)$ $//x_i, y_i \in \mathbb{F}_q$

1. $X := (X_1, ..., X_N) \leftarrow (x_1, ..., x_n)T_D^t$
   $Y := (Y_1, ..., Y_N) \leftarrow (y_1, ..., y_n)T_D^t$
2. $Z := X \odot Y = (Z_1, ..., Z_N) \leftarrow (X_1Y_1, \ldots, X_NY_N)$
3. $(z_1, \ldots, z_n) \leftarrow (Z_1, ..., Z_N)(T_{2D}^t)^{-1}C^t$.

OUTPUT: $z = xy = \sum_{i=1}^{n} z_i Ev_Q(f_i)$ $//z := xy$

Now, we present an initial setup algorithm which is only done once.

Algorithm 2 Setup algorithm

INPUT: $F/\mathbb{F}_q$, Q, D, $\mathcal{P} = \{P_1, \ldots, P_{2n+g+1}\}$.
OUTPUT: $\mathcal{B}_{2D}$, T_{2D} and CT_{2D}^{-1}.

(i) Check the function field $F/\mathbb{F}_q$, the place Q, the divisors D are such that Conditions (i) and (ii) in Theorem 2 can be satisfied.
(ii) Represent $\mathbb{F}_{q^n}$ as the residue class field of the place Q.
(iii) Construct a basis $\mathcal{B}_{2D} := (f_1, \ldots, f_n, f_{n+1}, \ldots, f_{2n+g-1})$ of $\mathcal{L}(2D)$, where $\mathcal{B}_D := (f_1, \ldots, f_n)$ is a basis of $\mathcal{L}(D)$, and $\mathcal{B}_D^c := (f_{n+1}, ..., f_{2n+g-1})$ a basis of the supplementary space $\mathcal{M}$ of $\mathcal{L}(D)$ in $\mathcal{L}(2D)$.
(iv) Compute the matrices T_{2D}, C and CT_{2D}^{-1}.

2.2 Complexity analysis

The total complexity, in terms of number of multiplications in $\mathbb{F}_q$, is equal to $(3n+1)(2n+g-1)$, including $3n(2n+g-1)$ scalar multiplications. Recall that the bilinear complexity of Chudnovsky^2 algorithms of type (1) in Theorem 1 satisfying assumptions of Theorem 2 is optimized. Therefore, we only focus on optimizing the scalar complexity of the algorithm. From Algorithm (1), we observe that the number of the scalar multiplications, denoted by N_s, depends directly on the number of zeros in the matrices T_D and $C.T_{2D}^{-1}$, respectively denoted by $N_{zero}(T_D)$ and $N_{zero}(C.T_{2D}^{-1})$. Indeed, all the involved matrices being constructed once, the multiplication by a coefficient zero in a matrix has not to be taken into account. Thus, we get the formula to compute the number of scalar multiplications of this algorithm with respect to the number of zeros of the involved matrices as follows:

$$\begin{aligned} N_s &= 2\Big(n(2n+g-1) - N_{zero}(T_D)\Big) + \Big(n(2n+g-1) - N_{zero}(C.T_{2D}^{-1})\Big) \\ &= 3n(2n+g-1) - N_{zero}, \end{aligned} \tag{6}$$

where

$$N_{zero} = 2N_{zero}(T_D) + N_{zero}(C.T_{2D}^{-1}). \tag{7}$$

3 Optimization of the scalar complexity

By Section 2.2, reducing the number of operations means finding an algebraic function field $F/\mathbb{F}_q$ having a genus g as small as possible and a suitable set of divisors and place $(D, Q, \mathcal{P})$ with a good representation of the associated Riemann-Roch spaces, namely such that the matrices T_D, T_{2D} and $C.T_{2D}^{-1}$ are as hollow as possible. Therefore, for a place Q and a suitable divisor D, we seek the best possible representations of Riemann-Roch spaces $\mathcal{L}(D)$ and $\mathcal{L}(2D)$ to maximize both parameters $N_{zero}(T_D)$ and $N_{zero}(C.T_{2D}^{-1})$.

3.1 Different types of strategy

With fixed divisor and places In this section, we consider the optimization for a fixed suitable set of divisor and places $(D, Q, \mathcal{P})$ for a given algebraic function field $F/\mathbb{F}_q$ of genus g. So, let us give the following definition:

Definition 4. *We call $\mathcal{U}_{D,Q,\mathcal{P}}^{F,n} := (\mathcal{U}_{D,Q,\mathcal{P}}^{A}, \mathcal{U}_{D,Q,\mathcal{P}}^{R})$ a Chudnovsky^2 multiplication algorithm of type (1) where $\mathcal{U}_{D,Q,\mathcal{P}}^{A} := E_{\mathcal{P}} \circ Ev_Q^{-1}$ and $\mathcal{U}_{D,Q,\mathcal{P}}^{R} := E_Q \circ {Ev_{\mathcal{P}}|_{ImEv_{\mathcal{P}}}}^{-1}$, satisfying the assumptions of Theorem 1. We will say that two algorithms are equal, and we will note: $\mathcal{U}_{D,Q,\mathcal{P}}^{F,n} = \mathcal{U}_{D',Q',\mathcal{P}'}^{F,n}$, if $\mathcal{U}_{D,Q,\mathcal{P}}^{A} = \mathcal{U}_{D',Q',\mathcal{P}'}^{A}$ and $\mathcal{U}_{D,Q,\mathcal{P}}^{R} = \mathcal{U}_{D',Q',\mathcal{P}'}^{R}$.*

Note that in this case, this definition makes sense only if the bases of implied vector-spaces are fixed. So, we denote respectively by $\mathcal{B}_Q$, $\mathcal{B}_D$, and $\mathcal{B}_{2D}$ the

basis of the residue class field F_Q, and of Riemann-Roch vector-spaces $\mathcal{L}(D)$, and $\mathcal{L}(2D)$ associated to $\mathcal{U}^{F,n}_{D,Q,\mathcal{P}}$. Note that the basis of the $\mathbb{F}_q$-vector space $\mathbb{F}_q^N$ is always the canonical basis. Then, we obtain the following result:

Proposition 1. *Let us consider an algorithm $\mathcal{U}^{F,n}_{D,Q,\mathcal{P}}$ such that the divisor D is an effective divisor, $D-Q$ a non-special divisor of degree $g-1$, and such that the cardinal of the set $\mathcal{P}$ is equal to the dimension of the Riemann-Roch space $\mathcal{L}(2D)$. Then we can choose the basis $\mathcal{B}_{2D}$ as (5) and for any σ in $GL_{\mathbb{F}_q}(2n+g-1)$, where $GL_{\mathbb{F}_q}(2n+g-1)$ denotes the linear group, we have*

$$\mathcal{U}^{F,n}_{\sigma(D),Q,\mathcal{P}} = \mathcal{U}^{F,n}_{D,Q,\mathcal{P}},$$

where $\sigma(D)$ denotes the action of σ on the basis $\mathcal{B}_{2D}$ of $\mathcal{L}(2D)$ in $\mathcal{U}^{F,n}_{D,Q,\mathcal{P}}$, with a fixed basis $\mathcal{B}_Q$ of the residue class field of the place Q and $\mathcal{B}_c$ the canonical basis of $\mathbb{F}_q^{2n+g-1}$. In particular, the quantity $N_{zero}(C.T_{2D}^{-1})$ is constant under this action.

Proof 1 *Let E, F and H be three vector spaces of finite dimension on a field K respectively equipped with the basis $\mathcal{B}_E$, $\mathcal{B}_F$ and $\mathcal{B}_H$. Consider two morphisms f and h respectively defined from E into F and from F into H and consider respectively their associated matrix $M_f(\mathcal{B}_E,\mathcal{B}_F)$ and $M_h(\mathcal{B}_F,\mathcal{B}_H)$. Then it is obvious that the matrix $M_{h\circ f}(\mathcal{B}_E,\mathcal{B}_H)$ of the morphism $h\circ f$ is independant from the choice of the basis $\mathcal{B}_F$ of F. As the divisor D is effective, we have $\mathcal{L}(D)\subset\mathcal{L}(2D)$ and then $\mathcal{U}^A_{D,Q,\mathcal{P}} := Ev_{\mathcal{P}}\circ Ev_Q^{-1} = Ev_{\mathcal{P}}\circ Ev_Q^{-1}$ and as $D-Q$ a non-special divisor of degree $g-1$, Ev_Q is an isomorphism from $\mathcal{L}(D)$ into F_Q and we have $\mathcal{U}^A_{D,Q,\mathcal{P}} = Ev_{\mathcal{P}}|_{\mathcal{L}(D)}\circ Ev_Q^{-1}$. Moreover, as the cardinal of the set $\mathcal{P}$ is equal to the dimension of the Riemann-Roch space $\mathcal{L}(2D)$, $Ev_{\mathcal{P}}$ is an isomorphism from $\mathcal{L}(2D)$ into $\mathbb{F}_q^{2n+g-1}$ equipped with the canonical basis $\mathcal{B}_c$. Thus, $\mathcal{U}^R_{D,Q,\mathcal{P}} := E_Q\circ Ev_{\mathcal{P}}^{-1}|_{ImEv_{\mathcal{P}}} = E_Q|_{\mathcal{L}(2D)}\circ Ev_{\mathcal{P}}^{-1}$. Then, the matrix of $\mathcal{U}^A_{D,Q,\mathcal{P}}$ (resp. $\mathcal{U}^R_{D,Q,\mathcal{P}}$) is invariant under the action of σ in $GL_{\mathbb{F}_q}(n)$ (resp. in $GL_{\mathbb{F}_q}(2n+g-1)$) on the basis $\mathcal{B}_D$ (resp. $\mathcal{B}_{2D}$) since the set (E,F,H) is equal to $(F_Q,\mathcal{L}(D),\mathcal{B}_c)$ (resp. $(\mathbb{F}_q^{2n+g-1},\mathcal{L}(2D),\mathcal{B}_Q)$) for $h\circ f := Ev_{\mathcal{P}}|_{\mathcal{L}(D)}\circ Ev_Q^{-1}$ (resp. $E_Q|_{\mathcal{L}(2D)}\circ Ev_{\mathcal{P}}^{-1}$).* □

Remark 3. Note that a priori for any permutation τ of the set $\mathcal{P}$, we have $\mathcal{U}^{F,n}_{\sigma(D),Q,\tau(\mathcal{P})}$ different from $\mathcal{U}^{F,n}_{D,Q,\mathcal{P}}$, where $\sigma(D)$ denotes the action of σ on the basis $\mathcal{B}_{2D}$ of $\mathcal{L}(2D)$ in $\mathcal{U}^{F,n}_{D,Q,\mathcal{P}}$, with a fixed basis $\mathcal{B}_Q$ of the residue class field of the place Q. Indeed, the action of τ corresponds to a permutation of the canonical basis $\mathcal{B}_c$ of $\mathbb{F}_q^{2n+g-1}$. It corresponds to a permutation of the lines of the matrix T_{2D}. In this case, $N_{zero}(T_{2D})$ is obviously constant under the action of τ but nothing enables us to claim that $N_{zero}(C.T_{2D}^{-1})$ is constant.

Proposition 2. *Let $\mathcal{U}^{F,n}_{D,Q,\mathcal{P}}$ be a Chudnovsky^2 multiplication algorithm in a finite field $\mathbb{F}_{q^n}$, satisfying the assumptions of Proposition 1. The optimal scalar*

complexity $\mu_{s,o}(\mathcal{U}^{F,n}_{D,Q,\mathcal{P}})$ *of* $\mathcal{U}^{F,n}_{D,Q,\mathcal{P}}$ *is reached for the set* $\{\mathcal{B}_{D,max}, \mathcal{B}_Q\}$ *such that* $\mathcal{B}_{D,max}$ *is the basis of* $\mathcal{L}(D)$ *satisfying*

$$N_{zero}(T_{D,max}) = \max_{\sigma \in GL_{\mathbb{F}_q}(n)} N_{zero}(T_{\sigma(D)}),$$

where $\sigma(D)$ *denotes the action of* σ *on the basis* $\mathcal{B}_D$ *of* $\mathcal{L}(D)$ *in* $\mathcal{U}^{F,n}_{D,Q,\mathcal{P}}$, $T_{D,max}$ *the matrix of the restriction of the evaluation map* $Ev_{\mathcal{P}}$ *on the Riemann-Roch vector space* $\mathcal{L}(D)$ *equipped with the bases* $\mathcal{B}_{D,max}$ *and* $\mathcal{B}_Q = Ev_Q(\mathcal{B}_{D,max})$. *In particular,*

$$\mu_{s,o}(\mathcal{U}^{F,n}_{D,Q,\mathcal{P}}) = \min_{\sigma \in GL_{\mathbb{F}_q}(n)} \{\mu_s(\mathcal{U}^{F,n}_{\sigma(D),Q,\mathcal{P}} \mid \sigma(\mathcal{B}_D) \text{ is the basis of } \mathcal{L}(D) \text{ and } \mathcal{B}_Q = Ev_Q(\mathcal{B}_D)\})$$

$$= 3n(2n+g-1) - (2N_{zero}(T_{D,max}) + N_{zero}(T^{-1}_{2D,n})),$$

where matrices C *and* T_{2D} *are defined with respect to the basis* $\mathcal{B}_Q = Ev_Q(\mathcal{B}_{D,max})$, *and* $\mathcal{B}_{2D} = \mathcal{B}_{D,max} \cup \mathcal{B}^c_D$ *with* $\mathcal{B}^c_D$ *a basis of the kernel of* $E_Q|_{\mathcal{L}(2D)}$, *and* $T^{-1}_{2D,n}$ *denotes the matrix constituted of the* n *first lines of the matrix* T^{-1}_{2D}.

Proof 2 *It follows directly from Proposition 1 and formulae (6) and (7). Note that since the quantity* $N_{zero}(C.T^{-1}_{2D})$ *is constant for any basis* $\mathcal{B}_{2D}$ *of* $\mathcal{L}(2D)$, *we can take the matrix* $C.T^{-1}_{2D} = T^{-1}_{2D,n}$ *if* $\mathcal{B}^c_D$ *is a basis of the kernel of* $E_Q|_{\mathcal{L}(2D)}$. □

Other strategies of optimization In the view of a complete optimization (with respect to scalar complexity i.e. with fixed bilinear complexity) of the multiplication in a finite field $\mathbb{F}_{q^n}$ by a Chudnovsky^2 type multiplication algorithm, we have to vary the eligible sets $(F, D, Q, \mathcal{P})$. As an example, for a fixed integer n, a given algebraic function field $F/\mathbb{F}_q$, and a couple divisor and place (D, Q) satisfying the conditions of Proposition 1, we must apply the optimization strategy studied in Section 3.1 on each suitable ordered subset $\mathcal{P}$ (of cardinal $2n+g-1$) of the set of rational places (i.e. each suitable subset $\mathcal{P}$ and all their associated permutations $\tau(\mathcal{P})$). Then we have to vary the couples (D, Q) and apply the previous step: for example, we can start by fixing the place Q and then vary the suitable divisors D. We can then look for a fixed suitable algebraic function field of genus g, up to isomorphism, and repeat all the previous steps. Finally, it is still possible to look at the trade-off between scalar complexity and bilinear complexity by increasing the genus and then re-conducting all the previous optimizations.

3.2 Optimization of scalar complexity in the elliptic case

Now, we study a specialisation of the Chudnovsky^2 multiplication algorithm of type (1) in the case of the elliptic curves. In particular, we improve the effective algorithm constructed in the article of U. Baum and M.A. Shokrollahi [4] which presented an optimal algorithm from the point of view of the bilinear complexity in the case of the multiplication in $\mathbb{F}_{256}/\mathbb{F}_4$ based on Chudnovsky^2 multiplication algorithm applied on the Fermat curve $x^3+y^3=1$ defined over $\mathbb{F}_4$. Our method of construction leads to a multiplication algorithm in $\mathbb{F}_{256}/\mathbb{F}_4$ having a lower scalar complexity with an optimal bilinear complexity.

Experiment of Baum-Shokrollahi The article [4] presents Chudnovsky^2 multiplication in $\mathbb{F}_{4^4}$, for the case $q = 4$ and $n = 4$. The elements of $\mathbb{F}_4$ are denoted by $0, 1, \omega$ and ω^2. The algorithm construction requires the use of an elliptic curve over $\mathbb{F}_4$ with at least 9 $\mathbb{F}_4$-rational points (which is the maximum possible number by Hasse-Weil Bound). Note that in this case, Conditions 1) and 2) of Theorem 2 are well satisfied. It is well known that the Fermat curve $u^3 + v^3 = 1$ satisfies this condition. By the substitutions $x = 1/(u + v)$ and $y = u/(u + v)$, we get the isomorphic curve $y^2 + y = x^3 + 1$. From now on, $F/\mathbb{F}_q$ denotes the algebraic function field associated to the elliptic curve $\mathcal{C}$ with plane model $y^2+y = x^3+1$, of genus one. The projective coordinates $(x : y : z)$ of $\mathbb{F}_4$-rational points of this elliptic curve are:

$$P_\infty = (0 : 1 : 0), P_1 = (0 : \omega : 1), P_2 = (0 : \omega^2 : 1), P_3 = (1 : 0 : 1),$$
$$P_4 = (1 : 1 : 1), P_5 = (\omega : 0 : 1), P_6 = (\omega : 1 : 1), P_7 = (\omega^2 : 0 : 1), P_8 = (\omega^2 : 1 : 1).$$

Now, we represent $\mathbb{F}_{256}$ as $\mathbb{F}_4[x]/\mathcal{Q}(x)$ with primitive root α, where $\mathcal{Q}(x) = x^4 + x^3 + \omega x^2 + \omega x + \omega$.

- For the place Q of degree 4, the authors considered $Q = \sum_{i=1}^{4} \mathfrak{p}_i$ where $\mathfrak{p}_1$ corresponds to the $\mathbb{F}_{4^4}$-rational point with projective coordinates $(\alpha^{16} : \alpha^{174} : 1)$ and $\mathfrak{p}_2, \mathfrak{p}_3, \mathfrak{p}_4$ are its conjugates under the Frobenius map. We see that α^{16} is a root of the irreducible polynomial $\mathcal{Q}(x) = x^4+x^3+\omega x^2+\omega x+\omega$. Thus, the place Q is a place lying over the place $(\mathcal{Q}(x))$ of $\mathbb{F}_4(x)/\mathbb{F}_4$. Note also that the place $((\mathcal{Q}(x))$ of $\mathbb{F}_4(x)/\mathbb{F}_4$ is totally splitted in the algebraic function field $F/\mathbb{F}_4$, which means that there exist two places of degree n in $F/\mathbb{F}_4$ lying over the place $(\mathcal{Q}(x))$ of $\mathbb{F}_4(x)/\mathbb{F}_4$, since the function field $F/\mathbb{F}_q$ is an extension of degree 2 of the rational function field $\mathbb{F}_4(x)/\mathbb{F}_q$. The place Q is one of the two places in $F/\mathbb{F}_4$ lying over the place $(\mathcal{Q}(x))$. Notice that the second place is given by the orbit of the conjugated point $(\alpha^{16} : \alpha^{174}+1 : 1)$. Therefore, we can represent $\mathbb{F}_{256} = \mathbb{F}_{4^4} = \mathbb{F}_4[x]/\mathcal{Q}(x)$ as the residue class field F_Q of the place Q in $F/\mathbb{F}_4$.
- For the divisor D, we choose the place described as $\sum_{i=1}^{4} \mathfrak{d}_i$ where $\mathfrak{d}_1$ corresponds to the $\mathbb{F}_{4^4}$-rational point $(\alpha^{17} : \alpha^{14} : 1)$ and $\mathfrak{d}_2, \mathfrak{d}_3, \mathfrak{d}_4$ are its conjugates under the Frobenius map. By computation we see that α^{17} is a root of irreducible polynomial $\mathcal{D}(x) = x^2 + x + \omega$ and $\deg D = 4$ because $\mathfrak{d}_1$, $\mathfrak{d}_2, \mathfrak{d}_3, \mathfrak{d}_4$ are all distinct. Therefore, D is the only place in $F/\mathbb{F}_4$ lying over the place $(\mathcal{D}(x))$ of $\mathbb{F}_4(x)$ since the residue class field F_D of the place D is a quadratic extension of the residue class field $F_{\mathcal{D}}$ of the place $\mathcal{D}$, which is an inert place of $\mathbb{F}_4(x)$ in $F/\mathbb{F}_4$.

The matrix T_{2D} obtained in the basis of Riemann-Roch space $L(2D)$:
$\mathcal{B}_{2D} = \{f_1 = 1/f, f_2 = x/f, f_3 = y/f, f_4 = x^2/f, f_5 = 1/f^2, f_6 = xy/f, f_7 = y/f^2, f_8 = x/f^2\}$, with $f = x^2 + x + \omega$ is the following:

$$T_{2D} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \omega^2 & 0 & 1 & 0 & \omega & 0 & \omega^2 & 0 \\ \omega^2 & 0 & \omega & 0 & \omega & 0 & 1 & 0 \\ \omega^2 & \omega^2 & 0 & \omega^2 & \omega & 0 & 0 & \omega \\ \omega^2 & \omega^2 & \omega^2 & \omega^2 & \omega & \omega & \omega & \omega \\ \omega & \omega^2 & 0 & 1 & \omega^2 & 0 & 0 & 1 \\ \omega & \omega^2 & \omega & 1 & \omega^2 & 1 & \omega^2 & 1 \\ \omega & 1 & 0 & \omega^2 & \omega^2 & 0 & 0 & \omega \end{pmatrix}.$$

Then, computation gives:

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & \omega & 0 & \omega^2 & \omega \\ 0 & 1 & 0 & 0 & 0 & \omega^2 & \omega & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & \omega & 0 & \omega \end{pmatrix} \text{ and } CT_{2D}^{-1} = \begin{pmatrix} 1 & \omega & 1 & \omega & 1 & 1 & \omega & 0 \\ 1 & 0 & \omega^2 & \omega & 1 & \omega^2 & 1 & \omega \\ 1 & \omega & \omega & \omega^2 & 1 & \omega^2 & \omega & \omega \\ 0 & \omega & \omega^2 & \omega & 1 & \omega^2 & 0 & 0 \end{pmatrix}.$$

Consequently, we obtain:

$$N_{zero}(T_D) = 10, \quad N_{zero}(CT_{2D}^{-1}) = 5.$$

Thus, the total number N_s of scalar multiplications in the algorithm constructed by Baum and Shokrollahi in [4] is $N_s = 71$ by the formula (6). In the next section, we follow the approach described in Section 3, and we improve the Chudnovsky^2 multiplication algorithm in $\mathbb{F}_{4^4}$ constructed by Baum and Shokrollahi in [4]. By using the same elliptic curve and the same set $\{D, Q, \mathcal{P}\}$, we obtain an algorithm with the same bilinear complexity and lower scalar complexity.

New design of the Baum-Shokrollahi construction The new construction of Chudnovsky^2 algorithm for the multiplication in $\mathbb{F}_{256}/\mathbb{F}_4$ using strategy given in Proposition 2 of Section 3.1 gives the following matrices T_{2D} with a better basis $\mathcal{B}_{2D} = (f_1, f_2, ..., f_8)$ of $\mathcal{L}(2D)$ space, where

$$\begin{aligned}
f_1 &= (\omega x^2 + x)/(x^2 + x + \omega), \\
f_2 &= (\omega^2 x^2 + \omega^2 x + \omega^2)/(x^2 + x + \omega), \\
f_3 &= \omega^2/(x^2 + x + \omega)c + (\omega^2 x + 1)/(x^2 + x + \omega), \\
f_4 &= \omega^2/(x^2 + x + \omega)c + (\omega^2 x + \omega)/(x^2 + x + \omega), \\
f_5 &= (x^2 + x)/(x^4 + x^2 + \omega^2)c + (x^4 + \omega x^3 + \omega x^2 + \omega x)/(x^4 + x^2 + \omega^2), \\
f_6 &= \omega^2 x/(x^4 + x^2 + \omega^2)c + (\omega x^4 + x^2 + \omega x + 1)/(x^4 + x^2 + \omega^2), \\
f_7 &= (\omega^2 x + 1)/(x^4 + x^2 + \omega^2)c + (\omega^2 x^4 + \omega^2 x^3 + \omega x^2 + \omega)/(x^4 + x^2 + \omega^2), \\
f_8 &= (x^2 + \omega x + 1)/(x^4 + x^2 + \omega^2)c + (x^4 + \omega x^3 + x^2 + \omega^2 x + \omega^2)/(x^4 + x^2 + \omega^2).
\end{aligned}$$

$$T_{2D} = \begin{pmatrix} \omega & \omega^2 & 0 & 0 & 1 & \omega & \omega^2 & 1 \\ 0 & \omega & 0 & \omega & 0 & \omega & 0 & \omega \\ 0 & \omega & \omega & 0 & 0 & \omega & \omega & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & \omega^2 & \omega^2 \\ 1 & 0 & 1 & 0 & \omega & \omega & \omega^2 & 0 \\ 0 & 0 & 1 & 0 & \omega & \omega & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & \omega^2 & \omega & 0 \\ \omega & \omega & 1 & \omega^2 & 1 & 0 & 0 & \omega^2 \end{pmatrix} \text{ and } T_{2D,4}^{-1} = \begin{pmatrix} 0 & \omega & 1 & 0 & 0 & 1 & 1 & \omega^2 \\ 0 & 0 & 0 & 0 & 1 & \omega & \omega & \omega^2 \\ \omega^2 & \omega & \omega^2 & \omega^2 & \omega & \omega & 0 & 0 \\ 1 & \omega^2 & \omega & \omega^2 & 0 & 0 & 1 & \omega^2 \end{pmatrix}$$

Therefore, $N_{zero}(T_D) = 16$ and $N_{zero}(T_{2D,4}^{-1}) = 11$. By the formula (6), we obtain $N_s = 53$, a gain of 25% over Baum and Shokrollahi's method.

References

1. Kevin Atighehchi, Stéphane Ballet, Alexis Bonnecaze, and Robert Rolland. Arithmetic in Finite Fields based on Chudnovsky's multiplication algorithm. *Mathematics of Computation*, 86(308):297–3000, 2017.
2. Stéphane Ballet. Curves with Many Points and Multiplication Complexity in Any Extension of $\mathbb{F}_q$. *Finite Fields and Their Applications*, 5:364–377, 1999.
3. Stéphane Ballet. Quasi-optimal Algorithms for Multiplication in the Extensions of $\mathbb{F}_{16}$ of degree 13, 14, and 15. *Journal of Pure and Applied Algebra*, 171:149–164, 2002.
4. Ulrich Baum and Amin Shokrollahi. An optimal algorithm for multiplication in $\mathbb{F}_{256}/\mathbb{F}_4$. *Applicable Algebra in Engineering, Communication and Computing*, 2(1):15–20, 1991.
5. Jean Chaumine. On the bilinear complexity of multiplication in small finite fields. *Comptes Rendus de l'Académie des Sciences, Série I*, 343:265–266, 2006.
6. David Chudnovsky and Gregory Chudnovsky. Algebraic complexities and algebraic curves over finite fields. *Journal of Complexity*, 4:285–316, 1988.
7. Hans De Groote. Characterization of division algebras of minimal rank and the structure of their algorithm varieties. *SIAM Journal on Computing*, 12(1):101–117, 1983.
8. Julia Pieltant. *Tours de corps de fonctions algébriques et rang de tenseur de la multiplication dans les corps finis*. PhD thesis, Université d'Aix-Marseille, Institut de Mathématiques de Luminy, 2012.
9. Amin Shokhrollahi. Optimal algorithms for multiplication in certain finite fields using algebraic curves. *SIAM Journal on Computing*, 21(6):1193–1198, 1992.
10. Henning Stichtenoth. *Algebraic Function Fields and Codes*. Number 314 in Lectures Notes in Mathematics. Springer-Verlag, 1993.
11. Shmuel Winograd. On multiplication in algebraic extension fields. *Theor. Comput. Sci.*, 8:359–377, 1979.

A New set up algorithm

A new setup algorithm can be obtained directly from the strategy developed in Section 3.1. More precisely, the following setup corresponds to the optimization described by Proposition 2.

Algorithm 3 New setup algorithm

INPUT: $F/\mathbb{F}_q,\ Q, D, \mathcal{P} = \{P_1, \ldots, P_{2n+g+1}\}$.
OUTPUT: $\mathcal{B}_{2D} = \mathcal{B}_D \cup \mathcal{B}_D^c$, T_{2D} and $T_{2D,n}^{-1}$.

(i) Check the function field $F/\mathbb{F}_q$, the place Q, the divisors D are such that Conditions (i) and (ii) in Theorem 2 can be satisfied.
(ii) Go through the set (or subset) of bases $\mathcal{B}_D$ of $\mathcal{L}(D)$.
(iii) Choose a basis $\mathcal{B}_D := (f_1, \ldots, f_n)$ such that the matrix T_D owns the largest number of zeros.
(iv) Set $\mathcal{B}_Q := Ev_Q(\mathcal{B}_D)$.
(v) Construct a basis $\mathcal{B}_D^c := (f_{n+1}, ..., f_{2n+g-1})$ of the supplementary space $\mathcal{M} := KerE_Q|_{\mathcal{L}(2D)}$ of $\mathcal{L}(D)$ in $\mathcal{L}(2D)$.
(iv) Compute the matrices T_{2D} and $T_{2D,n}^{-1}$ in the basis $\mathcal{B}_{2D} = \mathcal{B}_D \cup \mathcal{B}_D^c$.

B Magma implementation of the optimized multiplication algorithm in the finite field $\mathbb{F}_{4^4}$ over the finite field $\mathbb{F}_4$

```
//%%%%%%%%%PRE-COMPUTING FUNCTIONS %%%%%%%%%%
//Count the number of zeros in k rows et l column of a matrix A
function CountZeros(A, k, l)
CounterZ:=0;
for i:=1 to k do
    for j:=1 to l do
        if A[i,j] eq 0 then CounterZ:=CounterZ + 1;
        end if;
    end for;
end for;
return(CounterZ);
end function;
//Create matrix of the evaluation map of basis matrix B of m vectors at n points in P
function MatrixSecondEval(B,P,m,n)
ST:=[];
for j:=1 to n do
for i:=1 to m do
ST:=Append(ST, Evaluate(B[i,1], P[j]));
end for;
end for;
T:=Matrix(n,m, ST);
return(T);
end function;
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n:=4;g:=1;q:=4;
F4<a>:= GF(4);
G:=SL(4,F4); //cardinality= 987.033.600 matrices, 84 conjugacy classes.
Kx<x>:= FunctionField(F4);
Kxy<y>:= PolynomialRing(Kx);

f:=y^2 + y - x^3 -1;
F<c> := FunctionField(f);
LP:=Places(F,1);

QQ := x^4 + x^3 + a*x^2 +a*x+a;
Q := Decomposition(F,Zeros(Kx!QQ)[1])[1];
K<b> := ResidueClassField(Q);
"degree of Q is ", Degree(Q);

DD:= x^2+x+a;
D:= Decomposition(F,Zeros(Kx!DD)[1])[1];
D:=1*D;
"D-Q is special? ",IsSpecial(D-Q);
```

```
"dim L(D) is ", Dimension(D);

LD, h1 :=RiemannRochSpace(D);
L2D, h2 :=RiemannRochSpace(2*D);

B:=Basis(LD);
MLD:=[L2D!h1(v): v in B];
BL2D := h2(ExtendBasis(MLD,L2D));
BasisL2D:=Matrix(2*n+g-1,1, [BL2D[i] : i in [1..2*n+g-1]]);
BasisLD:=Matrix(n,1, [BL2D[i] : i in [1..n]]);
M:=Matrix(n+g-1,1, [BL2D[i] : i in [n+1..2*n+g-1]]);

T:=MatrixSecondEval(BasisL2D,LP, 2*n+g-1,2*n+g-1);
TI:=T^-1;

A:=Matrix(2*n+g-1,n, [T[j,i]: i in [1..n], j in [1..2*n+g-1]]);
//A is the matrix of the n first columns of T1
CounterA:=CountZeros(A,2*n+g-1,n);

EL2D:=Matrix(2*n+g-1,1, [Evaluate(BasisL2D[i,1],Q) : i in [1..2*n+g-1]]);
BELD:=Matrix(F4,n,n, [ElementToSequence(EL2D[i][1]) : i in [1..n]]);

//%%%%%%%%% the optimization %%%%%%%%%%
opBasisLD:=BasisLD;
opA:=A;
mCounterA:=CounterA;

//NOTE!! replace j by numbers from 1 to 84 before executing next commands.
rep:=Classes(G)[j][3];
Orbit:=Class(G,rep);

BasisLD1:=BasisLD;
k:= 1;
for k in [1.. #Orbit] do
        BasisLD1:= BELD^-1*Matrix(F,Orbit[k])*BasisLD;
    A1:= MatrixSecondEval(BasisLD1,LP,n,2*n+g-1);
    CounterA1:=CountZeros(A1,2*n+g-1,n);
    if mCounterA lt CounterA1 then
          mCounterA:=CounterA1; print(mCounterA);
          opBasisLD:=BasisLD1;
            end if;
end for;
//%%% Kernel of the restrictions of map E_Q on the L(2D) %%%%
nBasisL2D:=Matrix(2*n+g-1,1,[opBasisLD[i,1] : i in [1..n]]
                     cat [M[i,1] : i in [1..n+g-1]]);
```

```
nEL2D:=Matrix(2*n+g-1,1, [Evaluate(nBasisL2D[i,1],Q) : i in [1..2*n+g-1]]);
nBEL2D:=Matrix(F4,2*n+g-1,n, [ElementToSequence(nEL2D[i][1]) : i in [1..2*n+g-1]]);
Ker:=Parent(ZeroMatrix(F,n+g-1,2*n+g-1))! Matrix(Basis(NullSpace(nBEL2D)))*nBasisL2D;
//%%%%%%%%%% Space L(2D)= L(D) + Kernel %%%%%%%%%%
BasisL2D1:=Matrix(2*n+g-1,1, [opBasisLD[i,1] : i in [1..n]]
                    cat [Ker[i,1] : i in [1..n+g-1]]);
T1:=MatrixSecondEval(BasisL2D1,LP,2*n+g-1,2*n+g-1);
CounterT1I:=CountZeros(T1^-1,n,2*n+g-1);
Ns:=6*n^2-(2*mCounterA+CounterT1I);

print "The optimized basis of space L(2D):" ;  BasisL2D1;
print "The matrix T_2D of the algorithm:";  T1;
print "The matrix T_(2D,4) of the algorithm:" ;
Matrix(n,2*n+g-1,[T1^-1[i,j] : i in [1..n], j in [1..2*n+g-1]]);
print "The number of scalar multiplications of the algorithm: "; Ns;
%\end{lstlisting}
```