A strategy to optimize the complexity of Chudnovsky-type algorithms over the projective line

Stéphane Ballet, Alexis Bonnecaze, Bastien Pacifico

Contemporary Mathematics

A strategy to optimize the complexity of Chudnovsky-type algorithms over the projective line

Stéphane Ballet, Alexis Bonnecaze, and Bastien Pacifico

Abstract. Chudnovsky-type algorithms of multiplication in finite fields are well known for their good bilinear complexity. Recently, two advances have been obtained in the study of these algorithms: a strategy to optimize the scalar complexity of the original algorithm and the development of a generic recursive construction over the projective line. The construction of recursive Chudnodvsky-type algorithms over the projective line makes possible an efficient generic strategy to optimize their complexity (number of scalar and bilinear multiplications and additions in the base field). Then, several examples are given. In particular, considering Baum-Shokrollahi's experiment (1992), this constructive method provides a Chudnovsky-type algorithm of multiplication in $\mathbb{F}_{256}/\mathbb{F}_4$ with the best known complexity, while being much more efficient than existing optimization methods.
Keywords: Multiplicative complexity, Finite fields, Chudnovsky-type algorithms.

1. Introduction

The search for finite field multiplication algorithms with good algebraic complexity (cf. [**BCS97**]) is still a major issue in algorithmics and in cryptography. In this paper, we are interested in the number of arithmetic operations in the base field when multiplying in an extension of finite degree of a finite field. Several more general remarkable methods are known (for example [**Kar63**], [**F09**], and [**SS71**]) and can be used to address this problem. Recently, Harvey and van der Hoeven [**HvdH19**] have proven that such a multiplication can be computed with $\mathcal{O}(n \log n)$ operations (when q is fixed), assuming a widely-believed hypothesis. Similarly to the latter, many works focus on multiplication algorithms with efficient asymptotic complexities, giving estimations of the total number of operations in the base field relatively to the degree of the extension using the $\mathcal{O}$ notations. But these methods may not be optimal at finite distance (i.e. not from the point of view of asymptotic complexity), in particular for moderate-sized parameters (around a few thousands of bits). For example, Schnhage-Strassen's algorithm [**SS71**] has a better asymptotic complexity than Karatsuba's algorithm [**Kar63**], but outperforms the latter method only when the parameters become huge (around millions of bits), exceeding many usage sizes. As for the Frer algorithm [**F09**], it is competitive for even larger numbers. Moreover, it is well known that different operations do not have the same cost in terms of bit operations. In particular, multiplications are

more expensive than addition and bilinear multiplication is itself more expensive than scalar multiplication ([**STV92**], [**BCP$^+$21**] Section 1). We therefore choose to consider the algebraic complexity model while taking into account the different costs of these operations. To do so, we consider the multiplication method of D.V. and G.V. Chudnovsky [**CC88**] which admits the best bilinear complexity, both in finite distance and in asymptotics. We propose a strategy for the construction of this method in order to optimize the number of scalar multiplications and the number of additions while keeping the same bilinear complexity. Our goal is to be able to obtain in practice, i.e. at finite distance, a total complexity competitive with the best algorithms. It should also be noted that determining the asymptotic complexity of our method remains an open problem.

Let q be a prime power, $\mathbb{F}_q$ the finite field with q elements and $\mathbb{F}_{q^n}$ the degree n extension of $\mathbb{F}_q$. Let $\mathcal{B} = \{e_1, ..., e_n\}$ be a basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$ then for $x = \sum_{i=1}^n x_i e_i$ and $y = \sum_{j=1}^n y_j e_j$, the direct calculation of the product is given by

$$z = xy = \sum_{h=1}^{n} z_h e_h = \sum_{h=1}^{n} \Big(\sum_{i,j=1}^{n} t_{ijh} x_i y_j \Big) e_h, \tag{1}$$

where $e_i e_j = \sum_{h=1}^n t_{ijh} e_h$, $t_{ijh} \in \mathbb{F}_q$ being some constants.

One can distinguish two types of multiplications in this product: the bilinear ones, that are depending of the two elements being multiplied (i.e. the $x_i y_j$); and the scalar ones that are multiplications by a constant in $\mathbb{F}_q$. At first glance, the latest computation requires n^2 bilinear multiplications, n^3 scalar multiplications and $n^3 - n$ additions.

Definition 1.1. *Let $\mathcal{U}_{q,n}$ be an algorithm for the multiplication in $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$.*

- *The number of non-trivial scalar multiplications in $\mathbb{F}_q$ (i.e. multiplications by $\alpha \in \mathbb{F}_q$ with $\alpha \neq 0, 1$), used in $\mathcal{U}_{q,n}$ is called its scalar complexity, and is denoted by $\mu_s(\mathcal{U}_{q,n})$.*
- *The number of bilinear multiplications in $\mathbb{F}_q$ used in $\mathcal{U}_{q,n}$ is called its bilinear complexity, denoted by $\mu_b(\mathcal{U}_{q,n})$.*

We also denote by $a(\mathcal{U}_{q,n})$ the number of additions in $\mathbb{F}_q$ in the algorithm. Consequently, the total complexity of $\mathcal{U}_{q,n}$, denoted by $\mu(\mathcal{U}_{q,n})$ is given by $\mu(\mathcal{U}_{q,n}) = \mu_b(\mathcal{U}_{q,n}) + \mu_s(\mathcal{U}_{q,n}) + a(\mathcal{U}_{q,n})$. Note that bilinear multiplications are known to be computationally heavier than the scalar ones. Algorithms with good bilinear complexity are interpolation algorithms. Among them, the method introduced by D.V. and G.V. Chudnovsky [**CC88**] makes it possible to obtain the best known bilinear complexity. The original Chudnovsky-Chudnovsky Multiplication Algorithm (CCMA) is an interpolation algorithm over rational places of a function field. This construction has been generalized in different ways, for instance with the use of places of arbitrary degrees or the use of derivative evaluations. A detailed review on the topic is given in [**BCP$^+$21**].

Nevertheless, the total complexity of these algorithms has not been deeply studied yet. The first step in this direction has been made by Ballet et al. [**BBD19, BBD21**], giving a strategy to optimize the scalar complexity of the original CCMA. This strategy can be summarized as follows. The algorithm involves two matrices. For each coefficient distinct from zero and one of a matrix, a scalar multiplication

is performed. Therefore, in order to reduce the number of scalar multiplications, these matrices must have a maximum number of zeros and ones.

In this paper, we focus on the optimization of the recursive Chudnovsky-type algorithms over the projective line, proposed in [**BBP21**]. First, we rely on the work of [**BBD21**] to extend it to the use of places of arbitrary degrees. Then, a constructive method is developed to optimize the complexity of the algorithms on the projective line. The paper is organized as follows. In Section 2, we introduce Chudnovsky-type algorithms with evaluations at places of arbitrary degrees, and then the recursive construction over the projective line. In Section 3, we deal with the scalar complexity of Chudnovsky-type algorithms when non-rational places are used. Then, we propose a strategy to improve the scalar complexity of recursive Chudnovsky-type algorithms over the projective line. In Section 4, we give several examples. In particular, we illustrate this process on the extension of degree 4 of $\mathbb{F}_4$, and obtain an algorithm of same bilinear complexity and better total complexity than the Baum-Shokrollahi experiment [**BS91**], and its optimizations given in [**BBD21**].

2. Chudnovsky-type Algorithms

Let $F/\mathbb{F}_q$ be a function field of genus g over $\mathbb{F}_q$. For $\mathcal{O}$ a valuation ring, the place P is defined to be $P = \mathcal{O} \setminus \mathcal{O}^\times$. We denote by $F_P = \mathcal{O}_P/P$ the residue class field at the place P, that is isomorphic to $\mathbb{F}_{q^d}$, d being the degree of the place. A rational place is a place of degree 1. A divisor $\mathcal{D}$ is a formal sum $\mathcal{D} = \sum_i n_i P_i$, where P_i are places and n_i are relative integers. The support *supp* $\mathcal{D}$ of $\mathcal{D}$ is the set of the places P_j for which $n_j \neq 0$, and $\mathcal{D}$ is effective if all the n_i are positive. The degree of $\mathcal{D}$ is defined by $\deg \mathcal{D} = \sum_i n_i$. The Riemann-Roch space associated to the divisor $\mathcal{D}$ is denoted by $\mathcal{L}(\mathcal{D})$. A divisor $\mathcal{D}$ is said to be non-special if $\dim \mathcal{L}(\mathcal{D}) = \deg(\mathcal{D}) + 1 - g$. Details about algebraic function fields can be found in [**Sti08**].

2.1. CCMA with evaluation at places of arbitrary degrees. The latest generalization of CCMA is given in [**BCP$^+$21**]. Before introducing the algorithm, let us give a definition of the generalized Hadamard product.

Definition 2.1. *Let q be a prime power and $d_1, \dots, d_N$ be positive integers. The generalized Hadamard product in $\mathbb{F}_{q^{d_1}} \times \cdots \times \mathbb{F}_{q^{d_N}}$, denoted by $\underline{\odot}$, is given for all $(a_1, \dots, a_N), (b_1, \dots, b_N) \in \mathbb{F}_{q^{d_1}} \times \cdots \times \mathbb{F}_{q^{d_N}}$ by*

$$(a_1, \dots, a_N) \underline{\odot} (b_1, \dots, b_N) = (a_1 b_1, \dots, a_N b_N).$$

With this notation, we recall the version of the Chudnovsky-Chudnovsky algorithm useful for our study, namely the one allowing evaluations at places of arbitrary degrees (see [**BCP$^+$21**], Corollary 5.4).

Theorem 2.2 (CCMA at places of arbitrary degrees). *Let*

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

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

(i) *the evaluation map*

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

is surjective,

(ii) *the evaluation map*

$$\begin{array}{rccc} Ev_{\mathcal{P}}: & \mathcal{L}(2\mathcal{D}) & \to & \mathbb{F}_{q^{\deg P_1}} \times \cdots \times \mathbb{F}_{q^{\deg P_N}} \\ & f & \mapsto & \big(f(P_1), \ldots, f(P_N)\big) \end{array}$$

is injective.

Then,

(1) *we have a multiplication algorithm* $\mathcal{U}_{q,n}^{F,\mathcal{P}}(\mathcal{D}, Q)$ *such that for any two elements* x, y *in* $\mathbb{F}_{q^n}$*:*

$$xy = E_Q \circ {Ev_{\mathcal{P}}|_{ImEv_{\mathcal{P}}}}^{-1} \left(E_{\mathcal{P}} \circ Ev_Q^{-1}(x) \underline{\odot} E_{\mathcal{P}} \circ Ev_Q^{-1}(y) \right), \tag{2}$$

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,* $\underline{\odot}$ *the generalized Hadamard product and* $\circ$ *the standard composition map;*

(2) *the algorithm* $\mathcal{U}_{q,n}^{F,\mathcal{P}}(\mathcal{D}, Q)$ *defined by* (2) *has bilinear complexity*

$$\mu_b(\mathcal{U}_{q,n}^{F,\mathcal{P}}(\mathcal{D}, Q)) = \sum_{i=1}^{N} \mu_b(\mathcal{U}_{q,\deg P_i}(P_i)),$$

where $\mathcal{U}_{q,\deg P_i}(P_i)$ *is the algorithm used to multiply the evaluations at* P_i*, in* $\mathbb{F}_{q^{\deg P_i}}$*.*

Sufficient application conditions are given in the following.

Proposition 2.3 (Criteria for CCMA at places of arbitrary degrees). *Let* q *be a prime power and let* $n > 1$ *be an integer. If there exists an algebraic function field* $F/\mathbb{F}_q$ *of genus* g *with a set of places* $\mathcal{P} = \{P_1, \ldots, P_N\}$ *and an effective divisor* $\mathcal{D}$ *of degree* $n + g - 1$ *such that*

1) *there exists a place* Q *of degree* n *(which is always the case if* $2g + 1 \leq q^{\frac{n-1}{2}}(q^{\frac{1}{2}} - 1)$*),*
2) *Supp* $\mathcal{D} \cap (\mathcal{P} \cup Q) = \emptyset$*, and* $D - Q$ *is non-special,*
3) $\sum_{i=1}^{N} \deg P_i = 2n + g - 1$ *and* $2\mathcal{D} - \sum P_i$ *is non-special,*

then,

(i) *the evaluation map*

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

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

(ii) *and the evaluation map*

$$\begin{array}{rccc} Ev_{\mathcal{P}}: & \mathcal{L}(2\mathcal{D}) & \to & F_{q^{\deg P_1}} \times \cdots \times \mathbb{F}_{q^{\deg P_N}} \\ & f & \mapsto & \big(f(P_1), \ldots, f(P_N)\big) \end{array}$$

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

2.2. Recursive Chudnovsky-type algorithm over the projective line. In this paper, we focus on the optimization of recursive Chudnovsky-type algorithms over the projective line, introduced in [**BBP21**]. These algorithms are a specialization of the algorithm from Theorem 2.2 to the rational function field.

DEFINITION 2.4. *Let q be a prime power and n be a positive integer. A recursive Chudnovsky-type algorithm $\mathcal{U}_{q,n}^{\mathcal{P}_n}(Q)$ over the projective line is an algorithm $\mathcal{U}_{q,n}^{F,\mathcal{P}}(\mathcal{D},Q)$ satisfying the assumptions of Theorem 2.2 such that:*

- *$F/\mathbb{F}_q$ is the rational function field $\mathbb{F}_q(x)$,*
- *Q is a place of degree n of $\mathbb{F}_q(x)$,*
- *$\mathcal{D} = (n-1)P_\infty$, where P_∞ is the place at infinity of $\mathbb{F}_q(x)$,*
- *$\mathcal{P}_n$ is a set of places of degrees lower than n such that*

$$\sum_{P\in\mathcal{P}_n} \deg P = 2n-1,$$

- *the multiplication in $F_P \simeq \mathbb{F}_{q^d}$, where $d = \deg P$, is computed by $\mathcal{U}_{q,d}^{\mathcal{P}_d}(P)$, where $P \in \mathcal{P}_n$.*

Note that if $P \in \mathcal{P}_n$ is a rational place, the algorithm $\mathcal{U}_{q,1}^{\mathcal{P}_1}(P)$ consists in only a bilinear multiplication in $\mathbb{F}_q$. Such an algorithm verifies the criteria of Proposition 2.3. The bilinear complexity of these algorithms is given by the following.

PROPOSITION 2.5. *Let $\mathcal{U}_{q,n}^{\mathcal{P}_n}(Q)$ be a recursive Chudnovsky-type algorithm over the projective line. Its bilinear complexity is given by*

$$\mu_b(\mathcal{U}_{q,n}^{\mathcal{P}_n}(Q)) = \sum_{P\in\mathcal{P}_n} \mu_b(\mathcal{U}_{q,d}^{\mathcal{P}_d}(P)),$$

where $d = \deg P$.

Note that the evaluation at P_∞ is defined specifically in this context, since P_∞ is in the support of $\mathcal{D}$.

DEFINITION 2.6. *Let k be a positive integer and P_∞ be the place at infinity of $\mathbb{F}_q(x)$. Let $\mathcal{D} = kP_\infty$, and let $f = \sum_{i=0}^{k} f_i x^i \in \mathcal{L}(\mathcal{D})$. We define the evaluation at P_∞ to be for all $f \in \mathcal{L}(\mathcal{D})$,*

$$f(P_\infty) := f_k.$$

EXAMPLE 2.7. *Consider the multiplication in $\mathbb{F}_{4^4}$ over $\mathbb{F}_4$. Let $P_0, P_1, P_\omega, P_{\omega^2}$ and P_∞ be the rational places of $\mathbb{F}_4[x]$. Let P^2 be a place of degree 2, and Q be a place of degree 4. Then, we can construct a recursive Chudnovsky-type algorithm over the projective line with $\mathcal{P}_4 = \{P_0, P_1, P_\omega, P_{\omega^2}, P_\infty, P^2\}$. This algorithm uses the algorithm $\mathcal{U}_{4,2}^{\mathcal{P}_2}(P^2)$, defined with $\mathcal{P}_2 = \{P_0, P_1, P_\infty\}$. The diagram of its construction is given in Table 1. As well as the Baum-Shokrollahi experiment* [**BS91**]*, this algorithm has an optimal bilinear complexity $\mu_b(\mathcal{U}_{4,4}^{\mathcal{P}_4}(Q)) = 8$.*

3. Optimization of scalar complexity

3.1. Some general results. First, we discuss how to adapt the work of [**BBD21**] to Chudnovsky-type algorithms using places of arbitrary degrees.

Let $\mathcal{U}_{q,n} = \mathcal{U}_{q,n}^{F,\mathcal{P}}(\mathcal{D},Q)$ be an algorithm as defined in Theorem 2.2, and verifying the criteria of Proposition 2.3, for the multiplication in $\mathbb{F}_{q^n}$. Following [**BBD21**], we consider that the basis of $\mathbb{F}_{q^{\deg P_1}} \times \cdots \times \mathbb{F}_{q^{\deg P_N}}$ is always the canonical basis.

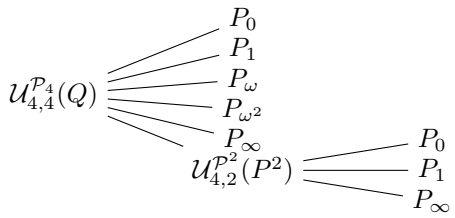

TABLE 1. Diagram of the construction of $\mathcal{U}_{4,4}^{\mathcal{P}}(Q)$.

The basis $\mathcal{B}_Q$ of $F_Q = \mathbb{F}_{q^n}$ is defined by $\mathcal{B}_Q = Ev_Q(\mathcal{B}_\mathcal{D})$, where $\mathcal{B}_\mathcal{D}$ is the basis of $\mathcal{L}(\mathcal{D})$. We also denote by $\mathcal{B}_{2\mathcal{D}}$ the basis of the Riemann-Roch space $\mathcal{L}(2\mathcal{D})$. Since we take $\mathcal{D}$ as an effective divisor, we have that $\mathcal{L}(\mathcal{D}) \subset \mathcal{L}(2\mathcal{D})$, and we take $\mathcal{B}_{2\mathcal{D}} = \mathcal{B}_\mathcal{D} \cup \mathcal{B}_\mathcal{D}^c$, where $\mathcal{B}_\mathcal{D}^c$ is a basis of the supplementary space of $\mathcal{L}(\mathcal{D})$ in $\mathcal{L}(2\mathcal{D})$. Let T_D (resp. $T_{2\mathcal{D}}$) be the matrix of $E_\mathcal{P} : \mathcal{L}(\mathcal{D}) \longrightarrow \mathbb{F}_{q^{\deg P_1}} \times \cdots \times \mathbb{F}_{q^{\deg P_N}}$ in the basis $\mathcal{B}_\mathcal{D}$ (resp. $Ev_\mathcal{P} : \mathcal{L}(2\mathcal{D}) \longrightarrow \mathbb{F}_{q^{\deg P_1}} \times \cdots \times \mathbb{F}_{q^{\deg P_N}}$, in the basis $\mathcal{B}_{2\mathcal{D}}$). Let C be the matrix of the map E_Q from the Riemann-Roch Space $\mathcal{L}(2\mathcal{D})$, in the basis $\mathcal{B}_{2\mathcal{D}}$, to the finite field $\mathbb{F}_{q^n}$, in the basis $\mathcal{B}_Q$ over $\mathbb{F}_q$. Using these matrices, Algorithm (2) is written

$$XY = CT_{2\mathcal{D}}^{-1}(T_\mathcal{D}(X) \underline{\odot} T_\mathcal{D}(Y)), \tag{3}$$

where X and Y are the two elements of $\mathbb{F}_{4^4}$ in the basis $\mathcal{B}_Q$ being multiplied, and $\underline{\odot}$ is the generalised Hadamard product. In the following, we consider the product of C and $T_{2\mathcal{D}}^{-1}$ as one matrix $CT_{2\mathcal{D}}^{-1}$.

Recall that the scalar complexity of the algorithm $\mathcal{U}_{q,n}$ is defined as its number of multiplications by a non-trivial constant (distinct from 0 or 1) in $\mathbb{F}_q$. In $\mathcal{U}_{q,n}$, the matrices $T_\mathcal{D}$ and $CT_{2\mathcal{D}}^{-1}$ provide some scalar multiplications of the algorithm. We therefore wish to obtain matrices with as many coefficients equal to zero or one as possible, to have a maximum number of trivial multiplications that do not count in the scalar complexity. Consequently, we focus on the number of zeros and ones in these matrices. We denote by $N_z(T_D)$ (resp. $N_z(CT_{2\mathcal{D}}^{-1})$) the number of zeros in $T_\mathcal{D}$ (resp. $CT_{2\mathcal{D}}^{-1}$)) and also denote by $N_1(T_\mathcal{D})$ (resp. $N_1(CT_{2D}^{-1})$) the number of ones in the matrices $T_\mathcal{D}$ (resp. $CT_{2\mathcal{D}}^{-1}$). Note that it is useful to distinguish between zeros and ones, since the coefficients equal to one must be counted in the number of additions used by the algorithm. Since the matrix $T_\mathcal{D}$ is used twice in the algorithm, and $CT_{2\mathcal{D}}^{-1}$ only once, we denote the total number of zeros by $N_z = 2N_z(T_D) + N_z(CT_{2\mathcal{D}}^{-1})$ and the total number of ones by $N_1 = 2N_1(T_D) + N_1(CT_{2\mathcal{D}}^{-1})$ If the algorithm $\mathcal{U}_{q,n}$ is an original CCMA, the evaluations are only at rational places and all the scalar multiplications are given by the matrices $T_\mathcal{D}$ and $CT_{2\mathcal{D}}^{-1}$. Hence, the scalar complexity is given by [**BBD19, BBD21**]:

$$\mu_s(\mathcal{U}_{q,n}) = 3n(n+g-1) - N_z - N_1. \tag{4}$$

In our study, we allow the evaluations to be at places of arbitrary degrees. Consequently, we have to count the scalar multiplications involved in an algorithm

$\mathcal{U}_{q,d}(P)$ (not necessarily of type Chudnovsky), where $d = \deg P$, for the multiplication in the residue class field $F_P \simeq \mathbb{F}_{q^d}$, required to multiply the evaluations at P.

PROPOSITION 3.1. *Let $\mathcal{U}_{q,n}$ be a Chudnovsky-type algorithm from Theorem 2.2, with evaluations at places of arbitrary degrees. Then, its scalar complexity is such that*

$$\mu_s(\mathcal{U}_{q,n}) = 3n(n+g-1) - N_z - N_1 + \sum_{P\in\mathcal{P}} \mu_s(\mathcal{U}_{q,d}(P)), \tag{5}$$

where $d = \deg P$ and $\mathcal{U}_{q,d}(P)$ is the algorithm used to multiply the evaluations at P.

PROOF. Follows from (4), where we add the scalar complexity of the multiplications in $\mathbb{F}_{q^{\deg P}}$ for all places P used by the algorithm. □

Since we have to add the scalar complexity of the algorithms $\mathcal{U}_{q,d}(P)$, for all $P \in \mathcal{P}$ and $d = \deg P$, the use of non rational places looks heavier for the scalar complexity. However, this is not necessarily the case. First, because a function field of lower genus can be used, which implies the use of smaller matrices, and also because the matrix $T_{\mathcal{D}}$ might contain more zeros.

PROPOSITION 3.2. *Let $\mathcal{U}_{q,n}$ be a Chudnovsky-type algorithm as defined in Theorem 2.2, satisfying Proposition 2.3. Consider $\mathcal{P}' \subset \mathcal{P}$ constructed by taking places that are in $\mathcal{P}$ by growing degrees as long as the sum of their degrees remains lower than or equal to $n+g-1$. Then, the number of zeros of $T_{\mathcal{D}}$ verifies*

$$N_z(T_{\mathcal{D}}) \leq n\big(n+g-1+\sum_{P\in\mathcal{P}\setminus\mathcal{P}'}(\deg P-1)\big).$$

PROOF. By Proposition 2.3, the divisor $\mathcal{D}$ is taken effective and of degree $n+g-1$. Thus, a function f in $\mathcal{L}(\mathcal{D})$ has at most $\deg \mathcal{D} = n+g-1$ zeros. Consider $f \in \mathcal{B}_{\mathcal{D}}$, then a column of $T_{\mathcal{D}}$ is given by the evaluations of f at the places in $\mathcal{P}$. Moreover, the evaluation at a place of degree d gives d coefficients in $\mathbb{F}_q$, and such an evaluation can give $d-1$ zeros without vanishing. As the ratio $(d-1)/d$ is increasing, the column defined by the evaluations of f would have the largest number of zeros if f has (at most) $n+g-1$ zeros at the places of smallest possible degree, i.e. at the places in $\mathcal{P}'$, and $\deg P - 1$ coefficients equal to zero for each other place. Then, a column of $T_{\mathcal{D}}$ is given by the evaluations of a function in $\mathcal{L}(\mathcal{D})$ has at most $n+g-1+\sum_{P\in\mathcal{P}\setminus\mathcal{P}'}(\deg P-1)$ coefficients equal to zeros. The bound is obtained by counting this maximal number of zeros for all the n columns of $T_{\mathcal{D}}$. □

Another important result for the strategy of optimization is that for given $\mathcal{P}$, $\mathcal{D}$ and Q, since the basis $\mathcal{B}_{\mathcal{D}}$ of $\mathcal{L}(\mathcal{D})$ is fixed, and that the basis $\mathcal{B}_{2\mathcal{D}} = \mathcal{B}_{\mathcal{D}} \cup \mathcal{B}^c_{\mathcal{D}}$ of $\mathcal{L}(2\mathcal{D})$ is an extension of the basis of $\mathcal{L}(D)$, the algorithm does not depend on the choice of $\mathcal{B}^c_{\mathcal{D}}$. This result is established in [**BBD19**, Proposition 1], and is also true when places of higher degrees are used (same proof holds).

3.2. Optimization of the complexity of a recursive Chudnovsky-type algorithm over the projective line. Consider a recursive Chudnovsky-type algorithm over the projective line $\mathcal{U}^{\mathcal{P}_n}_{q,n}(Q)$. The bilinear complexity of such an algorithm is known by Proposition 2.5. In this section, we introduce our strategy to optimize its total complexity.

PROPOSITION 3.3. *Let $\mathcal{U}_{q,n}^{\mathcal{P}_n}(Q)$ be a recursive Chudnovsky-type algorithm over the projective line. Then, the scalar complexity of $\mathcal{U}_{q,n}^{\mathcal{P}_n}(Q)$ is given by*

$$\mu_s(\mathcal{U}_{q,n}^{\mathcal{P}_n}(Q)) = 3n(2n-1) - N_z - N_1 + \sum_{P\in\mathcal{P}_n} \mu_s(\mathcal{U}_{q,d}^{\mathcal{P}_d}(P)),$$

where $d = \deg P$.

PROOF. Follows immediately from Proposition 3.1, where the Chudnovsky-type algorithm $\mathcal{U}_{q,d}^{\mathcal{P}_d}(P)$ over the projective line is used to multiply in $F_P \simeq \mathbb{F}_{q^d}$. □

By the results obtained in [**BBD19, BBD21**] and the previous section, the optimization of the scalar complexity of $\mathcal{U}_{q,n}^{\mathcal{P}_n}(Q)$ only depends on the basis of $\mathcal{L}(D)$, when $\mathcal{P}_n$, $\mathcal{D}$ and the place Q of degree n are fixed. Hence, the main focus is to find a basis $\mathcal{B}_\mathcal{D}$ of $\mathcal{L}(\mathcal{D})$ such that the matrices $T_\mathcal{D}$ and $CT_{2\mathcal{D}}^{-1}$ have the most possible zeros and ones. The basis of $\mathbb{F}_{q^n}$ will be defined in accordance with this basis.

The strategy proposed in [**BBD19**] and significantly completed in [**BBD21**] is to construct a first basis $\mathcal{B}_\mathcal{D}$, and apply the linear group $GL_n(\mathbb{F}_q)$ to look for the best possible bases. It is effective, but expensive. With a recursive algorithm over the projective line, one can construct directly some bases of $\mathcal{L}(\mathcal{D})$ that improve the total complexity of the algorithm without using the action of the linear group.

3.2.1. *Optimization strategy.* Now, let us introduce the heart of the strategy of optimization of Chudnovsky-type algorithms over the projective line. We want to sculpt $\mathcal{B}_\mathcal{D}$ to obtain the minimum scalar multiplications and additions. For this purpose, we want to get as many zeros as possible before processing the number of ones. More precisely, we will focus on obtaining the most possible zeros and then ones in the matrix $T_\mathcal{D}$. Two reasons for that: we do not have information on how the choice of the basis of $\mathcal{L}(\mathcal{D})$ affects the matrix $CT_{2\mathcal{D}}^{-1}$, and moreover the matrix $T_\mathcal{D}$ counts twice. For all these reasons, our goal is finally to sculpt $\mathcal{B}_\mathcal{D}$ such that $T_\mathcal{D}$ has a maximal number of zeros, and then a maximal number of ones.

Recall that $\mathcal{D} = (n-1)P_\infty$, and hence $\mathcal{L}(\mathcal{D})$ is the space of polynomials over $\mathbb{F}_q$ of degrees at most $n-1$. Moreover, the places of $\mathbb{F}_q(x)$ are given by the irreducible polynomials over $\mathbb{F}_q[x]$ and the place at infinity. The idea is to take the vectors of the basis $\mathcal{B}_\mathcal{D}$ as products of irreducible polynomials associated to places in $\mathcal{P}_n$. Therefore, the evaluation of such a vector will vanish at the places used to define it. This translates into zeros in the matrix $T_\mathcal{D}$. We define such bases as $\mathcal{P}_n-$bases of $\mathcal{L}(\mathcal{D})$.

DEFINITION 3.4 ($\mathcal{P}_n-$basis of $\mathcal{L}(\mathcal{D})$). *Let q be a prime power and $n > 1$ be an integer. Let $\mathcal{P}_n = \{P_1, \ldots, P_N\}$ be a set of distinct places of $\mathbb{F}_q(x)$ such that $\sum_{P\in\mathcal{P}_n} \deg P = 2n-1$. If P_j is not the place at infinity, let $P_j(x)$ be the monic irreducible polynomial associated to the place P_j. In the case of $P_j = P_\infty$, let $P_j(x) = 1$. We say that $\mathcal{B} = \{V_1, \ldots, V_n\}$ is a $\mathcal{P}_n-$basis of $\mathcal{L}(\mathcal{D})$ if every vector V_i of the basis $\mathcal{B}$ is defined as $V_i(x) = \prod_{P_j\in\mathcal{V}_i} P_j(x)$, where $\mathcal{V}_i$ is a subset of $\mathcal{P}_n$. The set $\mathcal{V}_i$ is called the support of V_i.*

We can notice that since $\mathcal{L}(\mathcal{D})$ is the space of polynomials of degree at most $n-1$, the vectors defining the basis shall be of degree at most $n-1$. The following proposition gives a lower bound for the number of zeros in the matrices $T_\mathcal{D}$ constructed using a $\mathcal{P}_n-$basis of $\mathcal{L}(\mathcal{D})$.

PROPOSITION 3.5. *Let $\mathcal{B} = \{V_1, \ldots, V_n\}$ be a $\mathcal{P}_n-$basis of $\mathcal{L}(\mathcal{D})$, and let $T_\mathcal{D}$ be a matrix obtained using $\mathcal{B}$. Then,*

$$N_z(T_\mathcal{D}) \geq \sum_{i=1}^{n} \deg V_i.$$

PROOF. Every column of $T_\mathcal{D}$ is given by the evaluation of a vector V_i at the places in $\mathcal{P}_n$. Every such vector is the product of some $P_{i,j}(x)$, where $P_{i,j} \in \mathcal{V}_i \subset \mathcal{P}_n$ and V_i can be written as the product $V_i(x) = P_{i,1}(x) \ldots P_{i,N_i}(x)$, where $\sum_{j=1}^{N_i} \deg P_{i,j} = \deg V_i$. In particular, $V_i(P_{i,j}) = 0$, and it gives $\deg P_{i,j}$ zeros in the $i-$th column of $T_\mathcal{D}$, for all $j = 1, \ldots, N_i$. Then, the $i - th$ column of $T_\mathcal{D}$ contains at least $\deg V_i$ zeros. This is the case for all the V_i in $\mathcal{B}_\mathcal{D}$, and then $N_z(T_\mathcal{D}) \geq \sum_{i=1}^{n} \deg V_i$. □

The preferred configuration to maximize the number of zeros is when the vectors of the $\mathcal{P}_n-$basis $\mathcal{B}$ are of degree $n-1$, or as close to $n-1$ as possible.

COROLLARY 3.6. *If $P_\infty \in \mathcal{P}_n$ and for all i, $\deg V_i = n-1$ or $n-2$. Then,*

$$N_z(T_\mathcal{D}) \geq n(n-1).$$

PROOF. If $\deg V_i = n-1$, then V_i has $n-1$ zeros. If $\deg V_i = n-2$, then $V_i(P_\infty) = 0$ since this evaluation is the coefficient in x^{n-1} of V_i (by Definition 2.6), and $n-2$ zeros at the places (distinct from P_∞) defining V_i. Finally, the n vectors of the basis have at least $n-1$ zeros. □

3.2.2. *Generic optimization.* Thus far, we have seen that using $\mathcal{P}_n-$bases gives an information on the number of zeros in the matrix $T_\mathcal{D}$, and then can be used to improve the complexity of Chudnovsky-type algorithms over the projective line. It remains to be proven that such a basis always exists. An efficient way to obtain such a basis is to construct $\mathcal{B} = \{V_1, \ldots, V_n\}$ such that for all i, $\deg V_i = i-1$. Secondly, we can construct the matrix $T_\mathcal{D}$ and maximize its number of ones by multiplying the vectors of the basis by a constant in $\mathbb{F}_q$. For each column in $T_\mathcal{D}$, suppose that $a \in \mathbb{F}_q$ is the non-zero scalar that occurs the most in the column. Then, we multiply the corresponding vector of the basis by a^{-1}. The generic construction of a $\mathcal{P}_n-$basis of $\mathcal{L}(\mathcal{D})$ is given in the following algorithm.

Algorithm 1 Construction of a generic $\mathcal{P}_n-$basis of $\mathcal{L}(\mathcal{D})$ and the associated matrix $T_\mathcal{D}$.

INPUT: q, n, $\mathcal{P}_n = \{P_1, \ldots, P_N\}$ be a set of places such that $\sum_{P \in \mathcal{P}_n} \deg P = 2n-1$.

OUTPUT: $B_\mathcal{D}$, T_D.

(1) For $i = 1, \ldots, n$, construct $V_i(x) = \prod_{P_j \in \mathcal{V}_i} P_j(x)$, such that $\deg V_i = i-1$, and $\mathcal{V}_i$ is a subset of $\mathcal{P}_n$.
(2) Construct $T_\mathcal{D}$. For each column of $T_\mathcal{D}$, if $a \in \mathbb{F}_q$ is the scalar that occurs the most, multiply both the corresponding vector of $\mathcal{B}_\mathcal{D}$ and the associated columns of $T_\mathcal{D}$ by a^{-1}.

The natural strategy to construct Chudnovsky-type algorithms over the projective line is to include in $\mathcal{P}_n$ all places by increasing degrees, until the sum of their degrees is equal to $2n-1$ (if the sum gets bigger than $2n-1$, remove a place of the appropriate degree, see [**BBP21**] Section 4.2).

PROPOSITION 3.7. *If $\mathcal{P}_n$ is constructed including places by increasing degrees, then Algorithm 1 is correct.*

PROOF. If $\mathcal{P}_n$ is constructed by taking places of increasing degrees, then it contains places of every degrees until some integer k, except in the case of $q = 2$ and the only degree 2 place of $\mathbb{F}_2(x)$ has been removed from $\mathcal{P}_n$. We can assume that P_∞ is always in $\mathcal{P}_n$. Note that the polynomial associated to P_∞ is defined by $P_\infty(x) = 1$ (see Definition 3.4). Suppose that $\mathcal{P}_n$ contains places of every degree until some integer k. Then, there exists $P_j(x)$ of degree j for $j = 1, \ldots, k$. Set $V_1(x) = 1$. Then, for $i = 2$ to $n-1$, we construct the polynomials $V_i(x)$ by taking the product of all monic polynomials by increasing degrees until the degree is equal to $i-1$ (if the degree gets greater than $i-1$, divide this product by an appropriate monic irreducible polynomial, that is in the support of V_i at this moment of the construction). If $q = 2$ and $\mathcal{P}_n$ does not contain the place of degree 2, we can divide V_i by the two irreducible polynomials of degree 1. Moreover, the sum of all $P_i(x)$ is of degree $2n - 2 \geq n$, and then we can construct $V_i(x)$ of degree d for all $d = 0, \ldots, n-1$. Finally, since there exists some products of the $P_i(x)$ for any degree d, one can obtain n vectors $V_1(x), \ldots, V_n(x)$ of degrees $0, \ldots, n-1$ respectively, such that for all j the function $V_j(x)$ is the product of some distinct $P_i(x)$. Then, $\mathcal{B} = \{V_1(x), \ldots, V_n(x)\}$ is a $\mathcal{P}_n-$basis of $\mathcal{L}(\mathcal{D})$. □

Moreover, Algorithm 1 is ending in polynomial time.

PROPOSITION 3.8. *Algorithm 1 is running in time $\mathcal{O}(n^3 \log n \log\log n)$.*

PROOF. Step 1. For the n vectors, we take at most n products of $P_i(x)$. We roughly consider that we have at most n^2 products of polynomials whose degrees are bounded by n. Each product can be computed with $\mathcal{O}(n \log n \log\log n)$ operations by Schnhage-Strassen ([**SS71**], [**vzGG03**, Theorem 8.23.]), thus this step can be completed in time $\mathcal{O}(n^3 \log n \log\log n)$.

Step 2. The matrix $T_\mathcal{D}$ is constructible using $\mathcal{O}(n^3)$ operations in the base field. Indeed, the coefficients of the matrix are obtained by computing the evaluations of the polynomials in the basis of $\mathcal{L}(\mathcal{D})$ at the places in $\mathcal{P}_n$. The evaluation of such a polynomial $V(x)$, that is of degree at most $n-1$, at a place P of degree $d < n$ is obtained by the modular reduction of $V(x)$ modulo $P(x)$. More precisely, let $v(x) = V(x) \pmod{P(x)} = \sum_{i=0}^{d-1} v_i x^i$. The coefficients of $\{v_0, \ldots, v_{d-1}\}$ are exactly the evaluation of $V(x)$ at P in the basis $\{1, \alpha, \ldots, \alpha^{d-1}\}$, for α a root of $P(x)$. Such a computation gives d coefficients of the matrix, and can be computed using the Euclidean Algorithm for polynomials, that uses $\mathcal{O}(nd)$ operations by [**vzGG03**, Theorem 3.11.]. The matrix $T_\mathcal{D}$ having $\mathcal{O}(n^2)$ coefficients, it means that all of them can be computed using $\mathcal{O}(n^3)$ operations. Then, for the n columns, we count each occurrence of non-zero scalars in the $2n-1$ coefficients. Then, we have to multiply the n vectors of the basis, and the n columns of $T_\mathcal{D}$ by a scalar. This last part is computed in $\mathcal{O}(n^2)$. Finally, this step uses $\mathcal{O}(n^3)$ operations in $\mathbb{F}_q$. □

REMARK 3.9. *In* [**HvdH19**]*, it is proven that if there exists a Linnik constant with $L < 1+2^{-1162}$, the product of two degree n polynomials over $\mathbb{F}_q$ can be computed in $\mathcal{O}(n \log q \log(n \log q))$, uniformly in q. Considering that q is fixed, the running time of Algorithm 1 becomes $\mathcal{O}(n^3 \log n)$.*

EXAMPLE 3.10. *Table 2 gives an illustration of the improvement of the complexity of Chudnovsky-type algorithms over the projective line for small extensions*

TABLE 2. Total complexity of Chudnovsky-type algorithms over $\mathbb{F}_2(x)$

n	2	3	4	5	6	$\cdots$	54
Non optimized	*7*	*23*	*54*	*91*	*129*		*10152*
Generic optimization	*7*	*20*	*49*	*77*	*99*		*8703*

of $\mathbb{F}_2$, and for the extension of degree 54. The non optimized algorithm uses the canonical basis $\{1, x, \ldots, x^{n-1}\}$ of $\mathcal{L}(\mathcal{D})$, while the generic optimization uses the basis provided by Algorithm 1. Details are given in Section 4.3.

3.2.3. *Non-generic optimization.* Even though we obtained a first improvement of the total complexity, this generic process does not provide the $\mathcal{P}_n-$basis of $\mathcal{L}(\mathcal{D})$ giving the best total complexity. For instance, we should take the vectors in the basis of the highest possible degree, since it ensures more zeros in the matrix. If the degree of the extension is low, we can check all possible $\mathcal{P}_n-$bases of $\mathcal{L}(\mathcal{D})$. Nevertheless, there are less than $2^{\#\mathcal{P}_n}$ possible vectors for the basis, and hence less than $\binom{2^{\#\mathcal{P}_n}}{n}$ possible $\mathcal{P}_n-$bases to try.

REMARK 3.11. *Even if this complete optimization is too heavy to be used generically for large extension degree, it is still way more efficient than the optimization of* [**BBD19, BBD21**] *involving the action of the linear group. Considering the optimization in the extension of degree* 13 *of $\mathbb{F}_{16}$, the linear group is of cardinality $\simeq 10^{203}$, while our strategy would consist in looking through $\lessapprox 10^{72}$ possible $\mathcal{P}_n-$bases.*

Depending on the time and resources that can be used, we can still improve the complexity of the algorithm. Instead of looking through all possible $\mathcal{P}_n-$bases, we can focus on the bases including only vectors of degree $n-1$ or $n-2$. Between these vectors, one can only look through the ones whose evaluations give a maximum number of zeros. An example of such optimization is given in Section 4.2.

This ends our strategy to optimize a Chudnovsky-type algorithm $\mathcal{U}_{q,n}^{\mathcal{P}_n}(Q)$ over the projective line, when the parameters $\mathcal{P}_n$ and Q are fixed. Finally, one can look for the best parameters, i.e. include in $\mathcal{P}_n$ in priority the places P such that the multiplication in F_P with $\mathcal{U}_{q,\deg P}^{\mathcal{P}_{\deg P}}(P)$ has the lowest complexity, or similarly for the place Q of degree n. A full optimization process is given in Section 4.1.

4. Examples

In this section, we provide several examples of optimizations of Chudnovsky-type algorithms over the projective line. All the computations were done using Magma Computational Algebra System [**BCP97**].

4.1. Multiplication in $\mathbb{F}_{256}$ over $\mathbb{F}_4$. We now illustrate the strategy introduced in the previous section to the multiplication in the extension of degree 4 of $\mathbb{F}_4$. The construction of a recursive Chudnovsky-type algorithm over the projective line to multiply in this extension has already been given in Example 2.7. More precisely, consider $\mathbb{F}_4 = \frac{\mathbb{F}_2[x]}{(x^2+x+1)}$. Hence, the elements of $\mathbb{F}_4$ are $\{0, 1, \omega, \omega^2\}$, where ω is a root of $x^2 + x + 1$. Let $\mathbb{F}_4(x)$, be the rational function field over $\mathbb{F}_4$. This function field has 5 rational places, that we denote by $P_0, P_1, P_\omega, P_{\omega^2}$ and

P_∞. These places are given by the irreducible polynomials $x, x+1, x+\omega, x+\omega^2$ and the place at infinity respectively. There exist 6 places of degree 2, which we denote by $P_1^2, \ldots, P_6^2$, and 60 places of degree 4. As in Example 2.7, we take $\mathcal{P}_4 = \{P_0, P_1, P_\omega, P_{\omega^2}, P_\infty, P^2\}$, where P^2 is one of the six places of degree 2, to obtain an algorithm of optimal bilinear complexity. Consequently, our algorithm requires to use the algorithm $\mathcal{U}_{4,2}^{\mathcal{P}_2}(P^2)$. We first focus on optimizing this algorithm, in order to take P^2 such that the complexity of $\mathcal{U}_{4,2}^{\mathcal{P}_2}(P^2)$ is minimal.

4.1.1. *Optimization of* $\mathcal{U}_{4,2}^{\mathcal{P}_2}(P^2)$. As seen in Example 2.7, the Chudnovsky-type algorithm over the projective line for the multiplication in the quadratic extension of $\mathbb{F}_4$ is defined using the ordered set $\mathcal{P}_2 = \{P_0, P_1, P_\infty\}$. Actually, the canonical basis $\{f_1, f_2\} = \{1, x\}$ of $\mathcal{L}(\mathcal{D})$ is already optimal. In fact, the matrix $T_\mathcal{D}$ is then given by

$$T_\mathcal{D} = \begin{pmatrix} f_1(P_0) & f_2(P_0) \\ f_1(P_1) & f_2(P_1) \\ f_1(P_\infty) & f_2(P_\infty) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

It has a maximal number of zero with respect to Proposition 3.2, and all its non-zero coefficients are equal to one. Hence, this matrix is optimal in terms of scalar complexity. Thus we do not need to search for a better basis of $\mathcal{L}(\mathcal{D})$. It remains to find for which places of degree 2 we obtain the more competitive algorithms. Hence, we compute $CT_{2\mathcal{D}}^{-1}$ for all the 6 possible places of degree 2 of $\mathbb{F}_4(x)$. We obtain

- $N_z(CT_{2\mathcal{D}}^{-1}) = 2$ with $P_1^2 = (x^2 + x + \omega)$ and $P_2^2 = (x^2 + x + \omega^2)$,
- $N_z(CT_{2\mathcal{D}}^{-1}) = 1$ with all other places.

Moreover, using P_1^2 or P_2^1, we have $N_1(CT_{2\mathcal{D}}^{-1}) = 3$. Hence, we can pick either P_1^2 or P_2^2 as the place of degree 2 in $\mathcal{P}_4$. By Proposition 3.3, we obtain

$$\mu_s(\mathcal{U}_{4,2}^{\mathcal{P}_2}(P_1^2)) = 1,$$

and the number of additions is given by

$$a(\mathcal{U}_{4,2}^{\mathcal{P}_2}(P_1^2)) = 4.$$

4.1.2. *Optimization of* $\mathcal{U}_{4,4}^{\mathcal{P}_4}(Q)$. Recall that $\mathcal{P}_4 = \{P_0, P_1, P_\omega, P_{\omega^2}, P_\infty, P^2\}$. By the previous section, one shall pick $P^2 = P_1^2 = (x^2+x+\omega)$ or $P_2^2 = (x^2+x+\omega^2)$. In the following, we choose $P^2 = P_1^2$.

We want to construct a good basis $\mathcal{B}_\mathcal{D}$ of $\mathcal{L}(\mathcal{D})$, with $\mathcal{D} = 3P_\infty$. Hence the Riemann-Roch space $\mathcal{L}(\mathcal{D})$ is the space of polynomials of degrees at most 3 over $\mathbb{F}_4$. Note that this time P_∞ is in $\mathcal{P}_4$. By Definition 2.6, a function f in $\mathcal{L}(\mathcal{D})$ has a zero at P_∞ if and only if f is a polynomial of degree at most 2. For all other places P in $\mathcal{P}_4$, let $P(x)$ be the corresponding polynomial. Then, a function f in $\mathcal{L}(\mathcal{D})$ has a zero at P if and only if $P(x) \mid f$. By Corollary 3.6, we shall construct the vectors of the basis as polynomials of degrees $n-1$ or $n-2$, which are the product of the polynomials defining the places in $\mathcal{P}_4$. Moreover, we want this vectors to vanish on rational places. Hence, we construct possible vectors for the basis of $\mathcal{L}(\mathcal{D})$ as

- the product of two irreducible polynomials of degree one $P_i(x)P_j(x)$, for $i, j \in \{0, 1, \omega, \omega^2\}$, then this vector has zeros at the places P_i, P_j and P_∞,
- the product of three irreducible polynomials of degree one $P_i(x)P_j(x)P_k(x)$, for $i, j, k \in \{0, 1, \omega, \omega^2\}$ then this vector has zeros at the places P_i, P_j and P_k.

Consequently, we can take the vectors of the basis $\mathcal{B}_{\mathcal{D}}$ as the product of three distinct elements of $\{1, x, x+1, x+\omega, x+\omega^2\}$ until a basis is found. This gives $\binom{5}{3} = 10$ possible vectors for the basis of $\mathcal{L}(\mathcal{D})$. Then, there are $\binom{10}{4} = 210$ possible combinations of these vectors to build the basis. Moreover, we want the vectors in each combination to be relatively prime in terms of polynomials, so that they do not all vanish at the same place. By computation, There are 150 possibilities left. We now consider that for given parameters, we only have to look through 150 possibilities to construct a $\mathcal{P}_4-$basis of $\mathcal{L}(\mathcal{D})$. In the strategy of [**BBD19**] and [**BBD21**], it was required to look through $\mid GL_4(\mathbb{F}_4) \mid = 2961100800$ possibilities. By Corollary 3.6, there are at least 12 zeros in the matrices $T_{\mathcal{D}}$ obtained using these bases. Nevertheless there can be at most 16 zeros by Proposition 3.2. We obtained exactly one basis of $\mathcal{L}(\mathcal{D})$ such that $N_z(T_{\mathcal{D}}) = 16$. This basis is given by

$$\mathcal{B}_{\mathcal{D}} = \{(x+\omega)(x+1), x(x+1), (x+\omega^2)(x+\omega), x(x+\omega^2)(x+\omega)\}.$$

The corresponding evaluation matrix $T_{\mathcal{D}}$ is then

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

where the rows are given by the evaluations at the places in $\mathcal{P}_4$, with the following order: $P_0, P_\omega, P_{\omega^2}, P_1, P_1^2$ and P_∞. Notice that the evaluation at P_1^2 takes two rows, in the basis $\{1, \alpha\}$, where α is a root of $P_1^2(x)$. Following Step 2 of Algorithm 1, one shall try to increase the number of ones in this matrix. In particular, the first column only contains 0 and ω. Hence, we modify the basis by multiplying the first vector by $\omega^{-1} = \omega^2$, we obtain

$$\mathcal{B}_{\mathcal{D}} = \{\omega^2(x+\omega)(x+1), x(x+1), (x+\omega^2)(x+\omega), x(x+\omega^2)(x+\omega)\}$$

and

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

The last step is now to find a place Q that gives the best scalar complexity. Finally, we compute the matrices $CT_{2\mathcal{D}}^{-1}$ using the basis of $\mathcal{L}(2\mathcal{D})$ given by $\mathcal{B}_{2\mathcal{D}} = \mathcal{B}_{\mathcal{D}} \cup \{x^4, x^5, x^6\}$ for all the 60 places Q of degree 4 in $\mathbb{F}_4(x)$. There are 3 places such that $CT_{2\mathcal{D}}^{-1}$ has a maximum number of zeros $N_z(CT_{2\mathcal{D}}^{-1}) = 12$. Between those matrices, two have 4 coefficients equal to one, and the last one, that is defined using $Q = (x^4 + \omega x^2 + \omega x + \omega^2)$, has 6 coefficients equal to one. The matrix is in this latest case given by

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

Finally, the algorithm $\mathcal{U}_{4,4}^{\mathcal{P}_4}(Q)$ is obtained with these parameters. The finite field $\mathbb{F}_{4^4}$ is represented as $\mathbb{F}_4[x]/(Q(x)) = \mathbb{F}_4[\beta]$, for β a root of $Q(x)$. Its basis over $\mathbb{F}_4$ is given by $\mathcal{B}_Q = Ev_Q(\mathcal{B}_\mathcal{D})$ and hence by

$$\mathcal{B}_Q = \{\beta^{43}, \beta^{198}, \beta^{108}, \beta^{109}\}.$$

4.1.3. *Comparison with other algorithms.* The matrices obtained contain $N_z(T_\mathcal{D}) = 16$ and $N_z(CT_{2\mathcal{D}}^{-1}) = 12$ zeros, and $N_1(T_\mathcal{D}) = 9$ and $N_1(CT_{2\mathcal{D}}^{-1}) = 6$ ones. Finally, we can compute the scalar complexity of $\mathcal{U}_{4,4}^{\mathcal{P}_4}(Q)$, including the scalar complexity of $\mathcal{U}_{4,2}^{\mathcal{P}_2}(P_1^2)$. By Proposition 3.3, we obtain,

$$\mu_s(\mathcal{U}_{4,4}^{\mathcal{P}_4}(Q)) = 17,$$

and

$$a(\mathcal{U}_{4,4}^{\mathcal{P}_4}(Q)) = 4 + 2 \times 5 + 12 = 26.$$

Originally, the Baum-Shokrollahi experiment [**BS91**] introduced an algorithm for the extension of degree 4 of $\mathbb{F}_4$ with optimal bilinear complexity. This algorithm is an original CCMA over the function field defined by the Fermat curve $u^3+v^3 = 1$. It also uses 51 scalar multiplications and 52 additions. In [**BBD19, BBD21**], the same algorithm is optimized (BS Optimized) with a good choice of the basis of $\mathbb{F}_{4^4}$ to obtain only 33 scalar multiplications and 33 additions. In this paper, the proposed algorithm is constructed over the rational function field, and only requires 17 scalar multiplications and 26 additions, for the same bilinear complexity.

At last, we want to compare our algorithm to well-known methods of polynomial interpolation. The generalized Karatsuba algorithm computes the product of two 4-terms polynomials using 9 (bilinear) multiplications and 24 additions (see [**WP06**], Appendix). Once that this product is computed, the modulo $Q(x)$ reduction still needs to be performed. For the comparison, we define $\mathbb{F}_{4^4}$ as in our construction, using $Q(x) = x^4 + \omega x^2 + \omega x + \omega^2$. The reduction then uses 9 additions and 8 scalar multiplications. The comparison between these methods is given in Table 3. We can see that the total complexity of our algorithm is equal to Karatsuba's to the nearest 1.

Remark 4.1. *Other experiments have similar performances, for example for the degree 3 extension of $\mathbb{F}_2$, regardless of the polynomial used to define the extension (see Table 6 and Table 7).*

Table 3. Comparison of algorithms for the multiplication in $\mathbb{F}_{4^4}$

Algorithm	$\mu_b(\mathcal{U})$	$\mu_s(\mathcal{U})$	$a(\mathcal{U})$	$\mu(\mathcal{U})$
Baum-Shokrollahi [**BS91**]	8	51	52	111
BS Optimized [**BBD21**]	8	19	43	70
Our construction	8	17	26	51
Karatsuba [**WP06**] + Reduction	9	8	33	50

4.2. The degree 13 extension of $\mathbb{F}_{16}$. Let the finite field $\mathbb{F}_{16}$ be defined as $\mathbb{F}_2(\omega)$, where ω is a root of x^4+x+1. In [**Bal02**], Ballet constructed a Chudnovsky-Chudnovsky Multiplication Algorithm with quasi-optimal bilinear complexity for the multiplication in the extension of degree $n = 13$ of $\mathbb{F}_{16}$. This algorithm is defined using the hyperelliptic curve given by the plane equation $y^2 + y = x^5$ of genus 2 that has 33 rational points. The algorithm uses 27 bilinear multiplications, that is still the best known bilinear complexity for the multiplication in this extension. The calculation of the number of operations of such an algorithm in Magma gives 833 scalar multiplications and 840 additions.

We can define a Chudnovsky-type algorithm over the projective line for the multiplication in $\mathbb{F}_{16^{13}}$ over $\mathbb{F}_{16}$. The rational function field over $\mathbb{F}_{16}$ has 17 rational places and 120 places of degree 2. We construct the set $\mathcal{P}_{16}$ with the 17 rational places and 4 places of degree 2. Then, the sum of the degrees of the places in $\mathcal{P}_{16}$ is equal to $17+2\times 4 = 25 = 2n-1$. As in the previous example, we start by including in $\mathcal{P}_{16}$ the places P^2 of degree 2 such that the algorithm $\mathcal{U}_{16,2}^{\mathcal{P}_2}(P^2)$ has the best complexity. There are 8 places P^2 of degree 2 such that $\mu_s(\mathcal{U}_{16,2}^{\mathcal{P}_2}(P^2)) = 1$ and $a(\mathcal{U}_{16,2}^{\mathcal{P}_2}(P^2)) = 4$. We include 4 of them in $\mathcal{P}_{16}$. In the following, we consider that the places of degree 2 in $\mathcal{P}_{16}$ are given by $(x^2+x+\omega^7)$, $(x^2+x+\omega^{14})$, $(x^2+x+\omega^{13})$ and $(x^2 + x + \omega^{11})$. Consider the place $Q = (x^{13} + x^4 + x^3 + x + 1)$ of $\mathbb{F}_{16}(x)$ of degree 13, and $\mathcal{D} = 12P_\infty$. We can now construct the algorithm $\mathcal{U}_{16,13}^{\mathcal{P}_{16}}(Q)$. Without any optimization, we use the canonical basis of $\mathcal{L}(\mathcal{D})$ given by $\{1, x, \ldots, x^{12}\}$. The algorithm then uses 29 bilinear multiplications, 686 scalar multiplications and 815 additions.

4.2.1. *Generic optimization.* With Algorithm 1, we can construct a $\mathcal{P}_{16}-$basis of $\mathcal{L}(\mathcal{D})$. This basis is given by

$$
\begin{aligned}
V_1 &= 1,\\
V_2 &= x,\\
V_3 &= \omega^{11}x^2 + \omega^{12}x,\\
V_4 &= \omega^{13}x^3 + \omega^3x^2 + \omega x,\\
V_5 &= \omega^{13}x^4 + \omega^9x^3 + \omega^{11}x^2 + \omega^4x,\\
V_6 &= \omega^{12}x^5 + \omega^{10}x^4 + \omega^3x^3 + x^2 + \omega^7x,\\
V_7 &= \omega^{13}x^6 + \omega^5x^5 + x^4 + \omega^3x^3 + \omega^{14}x^2 + \omega^{13}x,\\
V_8 &= \omega^{12}x^7 + \omega^7x^6 + \omega^{11}x^5 + \omega x^4 + \omega^3x^3 + \omega^6x^2 + \omega^3x,\\
V_9 &= \omega^{11}x^8 + \omega^2x^7 + \omega^9x^6 + \omega^8x^5 + \omega^{12}x^4 + \omega^6x^3 + \omega^7x^2 + \omega^9x,\\
V_{10} &= \omega^{14}x^9 + \omega^{13}x^8 + \omega x^7 + \omega^3x^6 + \omega x^5 + \omega^{12}x^4 + \omega^4x^3 + \omega^{10}x^2 + \omega^5x,\\
V_{11} &= \omega^7x^{10} + \omega^{11}x^9 + \omega^7x^8 + \omega^5x^7 + \omega^6x^6 + \omega^{11}x^5 + \omega^5x^4\\
&\quad +\omega^2x^3 + \omega x^2 + \omega^7x,\\
V_{12} &= \omega^5x^{11} + \omega^7x^{10} + \omega^8x^9 + \omega^{14}x^8 + \omega^{11}x^7 + \omega^4x^6 + \omega^7x^5\\
&\quad +\omega^6x^4 + \omega^{11}x^3 + \omega^6x^2 + x,\\
V_{13} &= x^{12} + \omega^9x^{11} + \omega^8x^{10} + \omega^4x^9 + \omega^9x^8 + \omega^{13}xx^7 + \omega^4x^6 + \omega^{12}x^5\\
&\quad +\omega^4x^4 + \omega^5x^3 + \omega^3x^2 + \omega^6x.
\end{aligned}
$$

Using this basis, the complexity of the algorithm is reduced to 614 scalar multiplications and 705 additions.

4.2.2. *Non-generic optimization.* In Remark 3.11, we saw that there are too many possible $\mathcal{P}_{16}-$bases for an exhaustive search. Nevertheless, we can still improve our algorithm. Using the proof of Proposition 3.2, a column of the matrix

$T_{\mathcal{D}}$ contains at most $n-1+4=16$ zeros, since $\mathcal{P}_{16}$ contains 4 places of degree 2. Moreover, this equality is possible if and only if the corresponding vector of the basis of $\mathcal{L}(\mathcal{D})$ vanishes only on rational places. Thus, we consider the set $\mathcal{S}=\{1, x, x+\omega, x+\omega^2, \ldots, x+\omega^{15}\}$. By corollary 3.6, we want to construct products of these elements of degree 11 or 12, such that such a function vanishes on 12 rational places of $\mathbb{F}_{16}(x)$. Such a polynomial is the product of 12 elements of $\mathcal{S}$. Hence, there are $\binom{17}{12}=6188$ possibilities. Moreover, the evaluation of each of these vectors gives at most 16 zeros. For all these functions f, we compute $Ev_{\mathcal{P}}(f)$, the vector of the evaluations of f at the places in $\mathcal{P}_{16}$. There are 49 of them containing 16 zeros. Finally, it remains to find a basis using 13 of these vectors. There are still $\binom{49}{13}=262596783764$ possibilities. This is very few compared to an exhaustive search of a $\mathcal{P}_{16}-$basis ($\lessapprox 10^{72}$ possibilities), but still too much. We finally randomly search a basis using these vectors, and apply Algorithm 1, Step 2 to reduce the number of scalar multiplications. We obtain the following.

$$
\begin{aligned}
V_1 &= \omega^4x^{12}+\omega^5x^{10}+\omega^8x^9+\omega^6x^8+\omega^2x^6+\omega^{10}x^5+\omega^3x^4+\omega x^3+\omega^9x^2\\
&\quad +\omega^{12}x+1,\\
V_2 &= \omega^4x^{12}+\omega^5x^{10}+\omega^8x^9+\omega^3x^8+\omega^2x^6+\omega^{10}x^5+\omega^9x^4+\omega x^3\\
&\quad +\omega^{12}x^2+\omega^6x,\\
V_3 &= \omega x^{12}+\omega^5x^{10}+\omega^2x^9+\omega^6x^8+\omega^8x^6+\omega^{10}x^5+\omega^3x^4+\omega^4x^3\\
&\quad +\omega^9x^2+\omega^{12}x,\\
V_4 &= \omega^9x^{11}+\omega^9x^{10}+\omega^{10}x^9+\omega^7x^7+\omega^7x^6+\omega^{14}x^5+x^4+\omega^{12}x^2+\omega^6x,\\
V_5 &= \omega^2x^{12}+\omega^{10}x^{10}+\omega^4x^9+\omega^9x^8+\omega x^6+\omega^5x^5+\omega^{12}x^4+\omega^8x^3\\
&\quad +\omega^6x^2+\omega^3x,\\
V_6 &= \omega^{13}x^{11}+\omega^{13}x^{10}+\omega^3x^9+\omega^{13}x^7+\omega^{13}x^6+\omega^8x^5+\omega^3x^3+\omega^8x^2+\omega^3x,\\
V_7 &= x^{12}+x^9+\omega^{10}x^8+x^6+\omega^5x^4+x^3+\omega^{10}x^2+\omega^5x,\\
V_8 &= \omega^8x^{12}+\omega^{10}x^{10}+\omega x^9+\omega^{12}x^8+\omega^4x^6+\omega^5x^5+\omega^6x^4+\omega^2x^3+\omega^3x^2\\
&\quad +\omega^9x+1,\\
V_9 &= \omega^{10}x^{12}+x^{10}+\omega^5x^9+\omega^8x^8+\omega^5x^6+x^5+\omega^4x^4+\omega^{10}x^3+\omega^2x^2+\omega x,\\
V_{10} &= x^{11}+x^{10}+x^9+\omega^{10}x^7+\omega^{10}x^6+\omega^5x^5+\omega^5x^4+x^2+\omega^5x,\\
V_{11} &= \omega^{12}x^{11}+\omega^{12}x^{10}+\omega^5x^9+\omega^{11}x^7+\omega^{11}x^6+\omega^7x^5+x^4+\omega^6x^2+\omega^3x,\\
V_{12} &= \omega^2x^{11}+\omega^2x^{10}+\omega^{10}x^9+x^7+x^6+\omega^6x^5+\omega^8x^4+\omega^9x^3+\omega^2x^2+\omega^6x,\\
V_{13} &= \omega^{14}x^{11}+\omega^{14}x^{10}+\omega^9x^9+\omega^{14}x^7+\omega^{14}x^6+\omega^4x^5+\omega^9x^3+\omega^4x^2+\omega^9x.
\end{aligned}
$$

Using this basis, the Chudnovsky-type algorithm $\mathcal{U}^{\mathcal{P}_{16}}_{16,13}(Q)$ now uses 423 scalar multiplications and 487 additions.

Karatsuba algorithm is more expensive in terms of bilinear complexity, using 66 bilinear multiplications instead of 29 with our method. It also uses 277 additions ([**WP06**], Appendix). As in the previous section, we compute the reduction modulo $Q(x)$ with 66 additions. We notice that there is no scalar multiplication. This is due to the choice of $Q(x)$ which has all its coefficients in $\mathbb{F}_2$. This kind of situation is more favorable to Karatsuba's technique than to our method for the scalar complexity. On the other hand, we can see that our method is clearly more efficient than the CCMA method using a curve of genus 2. The complexities of all these algorithms are summarized in Table 4.

4.3. Generic optimization over $\mathbb{F}_2$. For this last example, we fix the base field to be $\mathbb{F}_2$. We want to construct and optimize generically Chudnovsky-type algorithms over the projective line to reach large extensions. In the following,

TABLE 4. Comparison of algorithms for the multiplication in $\mathbb{F}_{16^{13}}$

Algorithm	$\mu_b(\mathcal{U})$	$\mu_s(\mathcal{U})$	$a(\mathcal{U})$	$\mu(\mathcal{U})$
CCMA [**Bal02**]	27	833	840	1700
Our construction				
Non optimized	29	686	815	1530
Generic optimization	29	614	705	1348
Non-generic optimization	29	423	487	939
Karatsuba [**WP06**] + Reduction	66	0	338	404

each set of places $\mathcal{P}_n$ is constructed by taking all places of growing degrees until the sum is equal to $2n-1$. Recall that if at some point the sum is bigger than $2n-1$ we can remove from $\mathcal{P}_n$ a place to obtain exactly $2n-1$. Note that since we consider extensions of $\mathbb{F}_2$, there are no scalar multiplication. Moreover, the number of additions depends on the place of degree n used to define the extension. For this reason, we return a list of values for the number of additions, following the order of the places given by Magma. We give the results for the extensions of degrees until 6 for a recursive Chudnovsky-type over the projective line first non-optimized (Table 5), then generically optimized (Table 6), and compared to the Karatsuba Algorithm ([**WP06**], Appendix) with the polynomial reduction (Table 7).

Using all places of degrees lower than or equal to 6 of $\mathbb{F}_2(x)$, one can define a Chudnovsky-type algorithm over the projective line for the multiplication in the extension of degree 54 of $\mathbb{F}_2$. The set $\mathcal{P}_{54}$ then contains all of these places. Considering $Q(x) = x^{54} + x^{34} + x^{32} + x^{31} + x^{30} + x^{29} + x^{27} + x^{25} + x^{21} + x^{18} + x^{17} + x^{16} + x^{15} + x^{13} + x^7 + x^4 + x^2 + x + 1$, we obtain the results of Table 8.

REMARK 4.2. *In this paper, we focused on constructing the matrices that gives the less possible operations when applied canonically. We did not focus on how to compute the multiplication by those matrices. For instance, if a non trivial α appears more than once in a column of a matrix, we can compute the multiplication by α once and for all, thus reducing the number of scalar multiplications.*

TABLE 5. Non optimized generic Chudnovsky-type algorithms over the projective line

Extension degree	$\mu_b(\mathcal{U})$	$a(\mathcal{U})$	$\min\{\mu(\mathcal{U})\}$
2	3	[4]	7
3	6	[18, 17]	23
4	11	[44, 44, 43]	54
5	15	[81, 85, 76, 78, 78, 79]	91
6	18	[118, 125, 115, 112, 112, 126, 112, 115, 111]	129

REMARK 4.3. *This strategy of optimization is specialized to Chudnovsky-type algorithms over the rational function field $\mathbb{F}_q(x)$, where the places are fully defined*

TABLE 6. Generically optimized Chudnovsky-type algorithms over the projective line

Extension degree	$\mu_b(\mathcal{U})$	$a(\mathcal{U})$	$\min\{\mu(\mathcal{U})\}$
2	3	$[4]$	7
3	6	$[14, 14]$	20
4	11	$[41, 41, 38]$	49
5	15	$[65, 68, 68, 62, 63, 66]$	77
6	18	$[88, 93, 95, 95, 89, 93, 81, 89, 85]$	99

TABLE 7. Karatsuba [**WP06**] + Reduction

Extension degree	$\mu_b(\mathcal{U})$	$a(\mathcal{U})$	$\min\{\mu(\mathcal{U})\}$
2	3	$[6]$	9
3	6	$[19, 19]$	25
4	9	$[32, 30, 35]$	49
5	15	$[57, 60, 58, 60, 63, 59]$	72
6	18	$[82, 78, 71, 67, 78, 79, 79, 83, 75]$	85

TABLE 8. Comparison of algorithms for the multiplication in $\mathbb{F}_{2^{54}}$

Algorithm	$\mu_b(\mathcal{U})$	$a(\mathcal{U})$	$\mu(\mathcal{U})$
Our Construction			
Non optimized	303	9849	10152
Generic optimization	303	8400	8703
Karatsuba [**WP06**]+ Reduction	630	4512	5142

by polynomials over $\mathbb{F}_q$. Nevertheless, one can consider the generalization of this strategy to optimize algorithms over a function field $F/\mathbb{F}_q$ of genus $g > 0$, by using local uniformizers of the places instead of monic irreducible polynomials.

REMARK 4.4. *This work, together with* [**BBD19, BBD21**]*, are the very first works on the scalar optimization of Chudnovsky-type algorithms, and it reduces significantly the number of algebraic operations used by these algorithms. Concerning practical efficiency, this is a first step before being able to explore and realize efficient implementations of the formulae given by the method. It would then be relevant to realize timings of implementations of our algorithms, and possibly to compare it for instance with the specific algorithms over $\mathbb{F}_4$ presented by Harvey, Lecerf and van der Hoeven in* [**HvdHL16**]*. But this work of comparison with these algorithms is sufficiently important to require a further work of its own. More precisely, it requires to translate the algorithms we obtained in terms of computer instructions, for example using multiplications, additions, but also shifts. Furthermore, it would also be interesting to compare our results with other algorithms of evaluation and interpolation over rational points (other than Karatsuba's), that are closer to our*

method, like the Toom-Cook methods optimized by Bodrato [**Bod07**] *and Bodrato and Zanoni in* [**BZ07**]*. But even this comparison requires a non-trivial translation of our method, which can only be done later.*

Acknowledgement

The authors are deeply grateful to the anonymous referees for their comments, that helped to improve and complete this article.

References

[Bal02] 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** (2002), 149–164.

[BBD19] Stéphane Ballet, Alexis Bonnecaze, and Thanh-Hung Dang, *On the scalar complexity of Chudnovsky*^{2} *multiplication algorithm in finite fields*, Algebraic Informatics (Cham) (Miroslav Ćirić, Manfred Droste, and Jean-Éric Pin, eds.), Springer International Publishing, 2019, pp. 64–75.

[BBD21] Stéphane Ballet, Alexis Bonnecaze, and Thanh-Hung Dang, *Optimization of the scalar complexity of Chudnovsky*^{2} *multiplication algorithms in finite fields*, Cryptography and Communications, accepted (2021).

[BBP21] Stphane Ballet, Alexis Bonnecaze, and Bastien Pacifico, *Multiplication in finite fields with Chudnovsky-type algorithms on the projective line*, 2021.

[BCP97] Wieb Bosma, John Cannon, and Catherine Playoust, *The Magma algebra system. I. The user language*, Journal of Symbolic Computation **24** (1997), no. 3-4, 235–265, Computational algebra and number theory (London, 1993).

[BCP$^+$21] Stphane Ballet, Jean Chaumine, Julia Pieltant, Matthieu Rambaud, Hugues Randriambololona, and Robert Rolland, *On the tensor rank of multiplication in finite extensions of finite fields and related issues in algebraic geometry*, Uspekhi Mathematichskikh Nauk, 76:1(457), 3194 (2021).

[BCS97] Peter Bürgisser, Michael Clausen, and Amin Shokrollahi, *Algebraic Complexity Theory*, Springer, 1997.

[Bod07] Marco Bodrato, *Towards optimal Toom-Cook multiplication for univariate and multivariate polynomials in characteristic 2 and 0*, Arithmetic of Finite Fields (Berlin, Heidelberg) (Claude Carlet and Berk Sunar, eds.), Springer Berlin Heidelberg, 2007, pp. 116–133.

[BS91] 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** (1991), no. 1, 15–20.

[BZ07] Marco Bodrato and Alberto Zanoni, *Integer and polynomial multiplication: Towards optimal Toom-Cook matrices*, Proceedings of the 2007 International Symposium on Symbolic and Algebraic Computation (New York, NY, USA), ISSAC '07, Association for Computing Machinery, 2007, p. 1724.

[CC88] David Chudnovsky and Gregory Chudnovsky, *Algebraic complexities and algebraic curves over finite fields*, Journal of Complexity **4** (1988), 285–316.

[F09] Martin Frer, *Faster integer multiplication*, SIAM Journal on Computing **39** (2009), no. 3, 979–1005.

[HvdH19] David Harvey and Joris van der Hoeven, *Polynomial multiplication over finite fields in time* $O(n \log n)$.

[HvdHL16] David Harvey, Joris van der Hoeven, and Grégoire Lecerf, *Fast polynomial multiplication over* $\mathbb{F}_{2^{60}}$, Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation (New York, NY, USA), ISSAC '16, Association for Computing Machinery, 2016, p. 255262.

[Kar63] Anatolii Karatsuba, *Multiplication of multidigit number on automata*, Soviet Physics Doklady **7** (1963), 595–596.

[SS71] Arnold Schnhage and Volker Strassen, *Schnelle Multiplikation groer Zahlen [fast multiplication of large numbers]*, Computing **7** (1971), no. 3-4, 281292.

[Sti08] Henning Stichtenoth, *Algebraic function fields and codes*, second ed., Graduate Texts in Mathematics, no. 254, Springer-Verlag, 2008.

[STV92] Igor Shparlinski, Michael Tsfasman, and Serguei Vlăduţ, *Curves with many points and multiplication in finite fields*, Coding Theory and Algebraic Geometry (Berlin) (H. Stichtenoth and M.A. Tsfasman, eds.), Lectures Notes in Mathematics, no. 1518, Springer-Verlag, 1992, Proceedings of AGCT-3 conference, June 17-21, 1991, Luminy, pp. 145–169.

[vzGG03] Joachim von zur Gathen and Jugen Gerhard, *Modern computer algebra*, Cambridge University Press, 2003.

[WP06] Andre Weimerskirch and Christof Paar, *Generalizations of the Karatsuba Algorithm for efficient implementations.*, IACR Cryptology ePrint Archive. (2006).

Stéphane Ballet, Aix Marseille Univ, CNRS, I2M, Marseille, France
Email address: Stephane.Ballet@univ-amu.fr

Alexis Bonnecaze, Aix Marseille Univ, CNRS, I2M, Marseille, France
Email address: Alexis.Bonnecaze@univ-amu.fr

Bastien Pacifico, Aix Marseille Univ, CNRS, I2M, Marseille, France
Email address: Bastien.Pacifico@univ-amu.fr