Quadratic-Time Algorithms for Normal Elements

Mark Giesbrecht
Cheriton School of Computer Science
University of Waterloo
mwg@uwaterloo.ca

Armin Jamshidpey
Cheriton School of Computer Science
University of Waterloo
armin.jamshidpey@uwaterloo.ca

Éric Schost
Cheriton School of Computer Science
University of Waterloo
eschost@uwaterloo.ca

ABSTRACT

For any finite Galois field extension K/F, with Galois group $G = \mathrm{Gal}(\mathsf{K}/\mathsf{F})$, there exists an element $\alpha \in \mathsf{K}$ whose orbit $G \cdot \alpha$ forms an F-basis of K. Such an α is called a *normal element* and $G \cdot \alpha$ is a *normal basis*. We introduce a probabilistic algorithm for finding a normal element when G is either a finite abelian or a metacyclic group. The algorithm is based on the fact that deciding whether a random element $\alpha \in \mathsf{K}$ is normal can be reduced to deciding whether $\sum_{\sigma \in G} \sigma(\alpha)\sigma \in \mathsf{K}[G]$ is invertible. Our algorithm requires a quadratic number of operations in the size of G for metacyclic G, and a slightly subquadratic number of operations for abelian G.

CCS CONCEPTS

• **Computing methodologies → Algebraic algorithms**.

KEYWORDS

Normal bases; Galois groups; metacyclic groups; fast algorithms

ACM Reference Format:
Mark Giesbrecht, Armin Jamshidpey, and Éric Schost. 2019. Quadratic-Time Algorithms for Normal Elements. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '19), July 15–18, 2019, Beijing, China.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3326229.3326260

1 INTRODUCTION

For a finite Galois extension K/F, with Galois group $G = \mathrm{Gal}(\mathsf{K}/\mathsf{F})$, an element $\alpha \in \mathsf{K}$ is called *normal* if its Galois conjugates $G \cdot \alpha = \{\sigma(\alpha) : \sigma \in G\}$ form a basis for K as an F-vector space. Constructive proofs are in most algebra texts, such as [22, §6.13].

While there is a wide range of well-known applications of normal bases in finite fields, such as fast exponentiation [10], there also exist applications of normal elements in characteristic zero. For instance, in multiplicative invariant theory, for a given permutation lattice and related Galois extension, a normal basis is useful in computing the multiplicative invariants explicitly [15].

A number of algorithms are available for finding a normal element in characteristic zero fields and finite fields. Because of their immediate applications in finite fields, algorithms for determining normal elements in this case are most commonly seen. A fast randomized algorithm for determining a normal element in a finite field $\mathbb{F}_{q^n}/\mathbb{F}_q$, where $\mathbb{F}_{q^n}$ is the finite field with q^n elements for any prime power q and integer $n > 1$, is presented by von zur Gathen and Giesbrecht [12], with a cost of $O(n^2 + n \log q)$ operations in $\mathbb{F}_q$. A faster randomized algorithm is introduced by Kaltofen and Shoup [18], with a cost of $O(n^{1.815} \log q)$ operations in $\mathbb{F}_q$. In the bit complexity model, Kedlaya and Umans [21] reduce the exponent of n to $1.5 + \epsilon$ (for any $\epsilon > 0$), using their quasi-linear time algorithm for *modular composition*. Lenstra [25] gives a deterministic algorithm which uses $n^{O(1)}$ operations. Augot and Camion [2] give the fastest known deterministic method, with a cost of $O(n^3 + n^2 \log q)$ operations in $\mathbb{F}_q$.

In characteristic zero, Schlickewei and Stepanov [30] gave an algorithm for finding a normal basis of a number field over $\mathbb{Q}$ with a cyclic Galois group of cardinality n which requires $n^{O(1)}$ operations in $\mathbb{Q}$. Poli [29] gives an algorithm for the more general case of finding a normal basis in an abelian extension K/F which requires $n^{O(1)}$ operations in F. More generally in characteristic zero, for any Galois extension K/F of degree n with Galois group given by a collection of n matrices, Girstmair [14] gives an algorithm which requires $O(n^4)$ operations in F to construct a normal element in K.

In this paper we present a new randomized algorithm for finding a normal element for abelian and metacyclic extensions, with a cost quadratic in the degree n of the extension. The costs of all algorithms are measured by counting *arithmetic operations* in F at unit cost. Our main conventions in this paper are the following.

Assumption 1. *Let K/F be a finite Galois extension presented as $\mathsf{K} = \mathsf{F}[x]/\langle P(x)\rangle$, for an irreducible polynomial $P \in \mathsf{F}[x]$ of degree n, with F of characteristic zero. Then,*

- *elements of K are written on the power basis $1, \xi, \ldots, \xi^{n-1}$, where $\xi := x \bmod P$;*
- *elements of G are represented by their action on ξ.*

In particular, for $g \in G$ given by means of $\gamma := g(\xi) \in \mathsf{K}$, and $\beta = \sum_{0 \le i < n} \beta_i \xi^i \in \mathsf{K}$, $g(\beta)$ is equal to $\beta(\gamma)$, the polynomial composition of β at γ (reduced modulo P).

Our algorithms combine techniques and ideas due to [12, 18]: $\alpha \in \mathsf{K}$ is normal if and only if the element $S_\alpha := \sum_{g \in G} g(\alpha) g \in \mathsf{K}[G]$ is invertible in the group algebra $\mathsf{K}[G]$. The algorithms choose α at random; a generic choice is normal (so we expect $O(1)$ random trials to be sufficient). However, writing down S_α involves $\Theta(n^2)$ elements in F, which precludes a subquadratic running time. Instead, knowing α, the algorithms use a randomized reduction to a similar question in $\mathsf{F}[G]$, that amounts to applying a random projection $\ell : \mathsf{K} \to \mathsf{F}$ to all entries of S_α, giving us an element $s_{\alpha,\ell} \in \mathsf{F}[G]$. For this, we adapt algorithms from [18], that are written for Galois groups of finite fields.

Having $s_{\alpha,\ell}$ in hand, we need to test its invertibility. We present an algorithm for abelian G which relies on the fact that $\mathsf{F}[G]$ is

ISSAC '19, July 15–18, 2019, Beijing, China
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6084-5/19/07…$15.00
https://doi.org/10.1145/3326229.3326260

isomorphic to a multivariate quotient polynomial ring by an ideal $(x_i^{e_i} - 1)_{1\le i\le m}$, where e_i's are positive integers.

For metacyclic groups, two algorithms are introduced to solve the same problem; which one is faster depends on the parameters defining our group. Both algorithms are based on testing the invertibility of an injective homomorphic image of $s_{\alpha,\ell}$ in a matrix algebra over a product of fields. These questions are closely related to Fourier transforms over G, and there is a vast literature on fast algorithms for Fourier transforms (over $\mathbb{C}$). For recent progress [7, 28] and references therein, though it is not clear how to apply these methods here.

Since our main goal is to highlight the exponent (in n) in our runtime analyses, costs are given using the soft-O notation: $S(n)$ is in $\tilde{O}(T(n))$ if it is in $O(T(n)\log(T(n))^c)$, for some constant c.

The main result of this paper is the following theorem. We use a constant $\omega(4/3)$, defined below, where $(3/4)\cdot\omega(4/3) < 1.99$, that describes the cost of certain rectangular matrix products.

Theorem 1.1. *Under Assumption 1, a normal element of* K *can be found using* $\tilde{O}(|G|^{(3/4)\cdot\omega(4/3)})$ *operations in* F *if* G *is abelian. The same problem for metacyclic groups can be solved using* $\tilde{O}(|G|^2)$ *operations in* F*. The algorithms are probabilistic of the Las Vegas type: they can select random elements from* F *at unit cost, the output is always correct, and the run-times are an expected number of operations in* F*.*

Although the cost of our algorithm is quadratic in the size of input for a general metacyclic group, it will be (slightly) subquadratic under specific parameters defining G (see Section 4).

Section 2 of this paper is devoted to definitions and preliminary discussions. In Section 3, two subquadratic-time algorithms are presented for the randomized reduction of our main question to invertibility testing in $\mathsf{F}[G]$, for respectively abelian and metacyclic groups. Finally, in Section 4, we show that the latter problem can be solved in quasi-linear time for an abelian group; for metacyclic groups, we give a quadratic-time algorithm, and discuss cases when this cost can be further improved.

Our algorithms make extensive use of known algorithms for polynomial and matrix arithmetic; in particular, we use the fact that polynomials of degree d in $\mathsf{F}[x]$ can be multiplied in $\tilde{O}(n)$ operations in F [31]. Arithmetic operations $(+,\times,\div)$ in K can thus be accomplished using $\tilde{O}(n)$ operations in F [11].

For matrix arithmetic, we will rely on some non-trivial results on rectangular matrix multiplication initiated by Lotti and Romani [27]. For $k \in \mathbb{R}$, we denote by $\omega(k)$ a constant such that matrices of size $n \times n$ can be multiplied by matrices of size $n \times \lceil n^k \rceil$ with $O(n^{\omega(k)})$ operations. [24] shows $\omega(4/3) < 2.654$; this follows from the upper bounds they give on $\omega(1.3)$ and $\omega(1.4)$, and the fact that $k \mapsto \omega(k)$ is convex [27]. In particular, $3/4\cdot\omega(4/3) < 1.99$. Note also the inequality $\omega(k) \ge 1 + k$ for $k \ge 1$, from the input/output size.

For square matrix multiplication, [23] shows $\omega(1) \le 2.373$, and we denote $\omega = \omega(1)$. Over a field K, we will frequently use the fact that further matrix operations (determinant or inverse) can be done in $O(n^\omega)$ base operations in K.

2 PRELIMINARIES

Assume K/F is a finite Galois extension with Galois group $G = \{g_1,\ldots,g_n\}$. If $\alpha \in \mathsf{K}$ is a normal element, then

$$\sum_{j=1}^{n} c_j g_j(\alpha) = 0, \quad c_j \in \mathsf{F} \tag{2.1}$$

implies $c_1 = \cdots = c_n = 0$. For $i \le n$, applying g_i to (2.1) yields

$$\sum_{j=1}^{n} c_j g_i g_j(\alpha) = 0. \tag{2.2}$$

Using (2.1) and (2.2), we form the linear system $\mathbf{M}_G(\alpha)\boldsymbol{c} = \mathbf{0}$, with $\boldsymbol{c} = [c_1 \cdots c_n]^T$ and where, for $\alpha \in \mathsf{K}$, $\mathbf{M}_G(\alpha) = \left[g_i g_j(\alpha)\right]_{1\le i,j\le n}$. Classical proofs then show there exists $\alpha \in \mathsf{K}$ with $\det(\mathbf{M}_G(\alpha)) \ne 0$.

This approach can be used as the basis of a randomized algorithm for finding a normal element: choose a random element α in K until we find one such that $\mathbf{M}_G(\alpha)$ is invertible. A direct implementation computes all the entries of the matrix and then uses linear algebra to compute its determinant; using fast matrix arithmetic this requires $O(n^\omega)$ operations in K, that is $\tilde{O}(n^{\omega+1})$ operations in F. This is at least cubic in n, and only a minor improvement over the previously best-known approach of Girstmair [14]. The main contribution of this paper is to show how to speed up this verification.

If we write $\alpha = a_0 + \cdots + a_{n-1}\xi^{n-1}$, the determinant of $\mathbf{M}_G(\alpha)$ is a (not identically zero) homogeneous polynomial of degree n in $(a_0,\ldots,a_{n-1})$. If the a_i's are chosen uniformly at random in a finite set $X \subset \mathsf{F}$, the Lipton-DeMillo-Schwartz-Zippel implies that the probability that α be normal is at least $1 - n/|X|$.

If G is cyclic, [12] compute the GCD of $S_\alpha := \sum_{1\le i\le n} g_i(\alpha)x^{i-1}$ and $x^n - 1$ instead of computing a determinant. This amounts to testing whether S_α is invertible in the group ring $\mathsf{K}[G] \simeq \mathsf{K}[x]/\langle x^n - 1\rangle$. This is a general fact: for any G, $\mathbf{M}_G(\alpha)$ is the matrix of (left) multiplication by the orbit sum

$$S_\alpha := \sum_{g\in G} g(\alpha)g \in \mathsf{K}[G],$$

and α being normal is equivalent to S_α being a unit in $\mathsf{K}[G]$. This point of view may make it possible to avoid linear algebra of size n over K, but writing S_α itself still involves $\Theta(n^2)$ elements in F. The following lemma gives a randomized reduction to testing whether a suitable projection of S_α in $\mathsf{F}[G]$ is a unit.

Lemma 2.1. *For* $\alpha \in \mathsf{K}$, $\mathbf{M}_G(\alpha)$ *is invertible if and only if*

$$\ell(\mathbf{M}_G(\alpha)) := [\ell(g_i g_j(\alpha))]_{ij} \in M_n(\mathsf{F})$$

is invertible, for a generic F*-linear projection* $\ell : \mathsf{K} \to \mathsf{F}$.

Proof. ($\Rightarrow$) For $\alpha \in \mathsf{K}$, any entry of $\mathbf{M}_G(\alpha)$ can be written as

$$\sum_{k=0,\ldots,n-1} a_{ijk}\xi^k, \tag{2.3}$$

and for $\ell : \mathsf{K} \to \mathsf{F}$, the corresponding entry in $\ell(\mathbf{M}_G(\alpha))$ can be written $\sum_{k=0}^{n-1} a_{ijk}\ell_k$, with $\ell_k = \ell(\xi^k)$. Replacing these ℓ_k's by indeterminates L_k's, the determinant becomes a polynomial in $P \in \mathsf{F}[L_1,\ldots,L_n]$. Viewing P over K, we have $P(1,\xi,\ldots,\xi^{n-1})$ $= \det(\mathbf{M}_G(\alpha))$, which is non-zero by assumption; so, P is not zero.

($\Leftarrow$) Assume $\mathbf{M}_G(\alpha)$ is not invertible. Following the proof of [15, Lemma 4], we first show that there exists a non-zero $\boldsymbol{u} \in \mathsf{F}^n$ in the kernel of $\mathbf{M}_G(\alpha)$.

The elements of G act on rows of $\mathbf{M}_G(\alpha)$ entrywise and the action permutes the rows the matrix. Assume $\varphi : G \to \mathfrak{S}_n$ is the group homomorphism such that $g(\mathbf{M}_i) = \mathbf{M}_{\varphi(g)(i)}$ for all i, where $\mathbf{M}_i$ is the i-th row of $\mathbf{M}_G(\alpha)$.

Since $\mathbf{M}_G(\alpha)$ is singular, there exists a non-zero $\boldsymbol{v} \in \mathsf{K}^n$ such that $\mathbf{M}_G(\alpha)\boldsymbol{v} = 0$; we choose $\boldsymbol{v}$ having the minimum number of non-zero entries. Let $i \in \{1, \dots, n\}$ such that $v_i \neq 0$. Define $\boldsymbol{u} = 1/v_i\boldsymbol{v}$. Then, $\mathbf{M}_G(\alpha)\boldsymbol{u} = 0$, which means $\mathbf{M}_j\boldsymbol{u} = 0$ for $j \in \{1, \dots, n\}$. For $g \in G$, we have $g(\mathbf{M}_j\boldsymbol{u}) = \mathbf{M}_{\varphi(g)(j)}g(\boldsymbol{u}) = 0$. Since this holds for any j, we conclude that $\mathbf{M}_G(\alpha)g(\boldsymbol{u}) = 0$, hence $g(\boldsymbol{u}) - \boldsymbol{u}$ is in the kernel of $\mathbf{M}_G(\alpha)$. On the other hand since the i-th entry of $\boldsymbol{u}$ is one, the i-th entry of $g(\boldsymbol{u}) - \boldsymbol{u}$ is zero. Thus the minimality assumption on $\mathbf{v}$ shows that $g(\boldsymbol{u}) - \boldsymbol{u} = 0$, equivalently $g(\boldsymbol{u}) = \boldsymbol{u}$, so $\boldsymbol{u} \in \mathsf{F}^n$.

Now we show that $\ell(\mathbf{M}_G(\alpha))$ is singular for all ℓ. By (2.3),

$$\mathbf{M}_G(\alpha) = \sum_{j=0,\dots,n-1} \mathbf{M}^{(j)}\xi^j, \quad \mathbf{M}^{(j)} \in M_n(\mathsf{F}) \text{ for all } j.$$

Since $\boldsymbol{u}$ is in F^n, $\mathbf{M}_G(\alpha)\boldsymbol{u} = 0$ yields $\mathbf{M}^{(j)}\boldsymbol{u} = 0$ for $j \leq n$. Hence,

$$\sum_{j=0,\dots,n-1} \mathbf{M}^{(j)}\ell_j\boldsymbol{u} = 0$$

for any ℓ_j's in F, and $\ell(\mathbf{M}_G(\alpha))$ is not invertible for any ℓ. □

Our algorithm chooses random α in K and $\ell : \mathsf{K} \to \mathsf{F}$, and let

$$s_{\alpha,\ell} := \sum_{g\in G} \ell(g(\alpha))g \in \mathsf{F}[G]. \tag{2.4}$$

The matrix $\ell(\mathbf{M}_G(\alpha))$ is the multiplication matrix by $s_{\alpha,\ell}$ in $\mathsf{F}[G]$, so once $s_{\alpha,\ell}$ is known, we are left with testing whether it is a unit in $\mathsf{F}[G]$. In the next two sections, we address the respective questions of computing $s_{\alpha,\ell}$, and testing its invertibility in $\mathsf{F}[G]$.

3 COMPUTING ORBIT SUM PROJECTIONS

In this section we present algorithms to compute $s_{\alpha,\ell}$, when G is either abelian or metacyclic. We start by sketching our ideas in simplest case, cyclic groups, with $G = \langle g \rangle$.

Given $\alpha \in \mathsf{K}$ and $\ell : \mathsf{K} \to \mathsf{F}$, we want to compute

$$\ell(g^i(\alpha)), \text{ for } 0 \leq i \leq n-1. \tag{3.1}$$

Kaltofen and Shoup [18] call this the *automorphism projection problem* and gave an algorithm to solve it in subquadratic time, when g is the q-power Frobenius $\mathbb{F}_{q^n} \to \mathbb{F}_{q^n}$. The key idea in their algorithm is to use the baby-steps/giant-steps technique: for a suitable parameter t, the values in (3.1) can be rewritten as

$$(\ell \circ g^{tj})(g^i(\alpha)), \text{ for } 0 \leq j < m := \lceil n/t \rceil \text{ and } 0 \leq i < t.$$

First, we compute all $G_i := g^i(\alpha)$ for $0 \leq i < t$. Then we compute all $L_j := \ell \circ g^{tj}$ for $0 \leq j < m$, where the L_j's are themselves linear mappings $\mathsf{K} \to \mathsf{F}$. Finally, a matrix product yields all values $L_j(G_i)$.

The algorithm of Kaltofen and Shoup [18] uses properties of the Frobenius mapping. In our case, we cannot apply these results directly; instead, we have to revisit proofs from Kaltofen and Shoup [18] using rectangular matrix multiplication.

3.1 Multiple automorphism evaluation

The remark following Assumption 1 reduces automorphism evaluation to modular composition of polynomials (this idea goes back to von zur Gathen and Shoup [13], where it was credited to Kaltofen). For instance, given $g \in G$ (by means of $\gamma := g(\xi)$), we can deduce $g^2 \in G$ (again, by means of its image at ξ) as $\gamma(\gamma)$; this can be done with $\tilde{O}(n^{(\omega+1)/2})$ operations in F using the modular composition algorithm of [5]. The algorithms below describe similar operations along these lines.

Lemma 3.1. *Given $\alpha_1, \dots, \alpha_s$ in K and g in G, with $s = O(\sqrt{n})$, we can compute $g(\alpha_1), \dots, g(\alpha_s)$ in $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in F.*

Proof (Compare [18, Lemma 3]) As noted above, for $i \leq s$, $g(\alpha_i) = \alpha_i(\gamma)$, with $\gamma := g(\xi) \in \mathsf{K}$. Let $t := \lceil n^{3/4} \rceil$, $m := \lceil n/t \rceil$, and rewrite $\alpha_1, \dots, \alpha_s$ as

$$\alpha_i = \sum_{0\leq j<m} a_{i,j}\xi^{tj},$$

where the $a_{i,j}$'s are polynomials of degree less than t. The next step is to compute $\gamma_i := \gamma^i$, for $i = 0, \dots, t$. There are t products in K to perform, so this amounts to $\tilde{O}(n^{7/4})$ operations in F.

Having γ_i's in hand, one can form the matrix $\mathbf{\Gamma} := [\Gamma_0 \ \cdots \ \Gamma_{t-1}]^T$, where each column Γ_i is the coefficient vector of γ_i (with entries in F); this matrix has $t \in O(n^{3/4})$ rows and n columns. We also form

$$\mathbf{A} := \left[A_{1,0} \cdots A_{1,m-1} \cdots A_{s,0} \cdots A_{s,m-1}\right]^T,$$

where $A_{i,j}$ is the coefficient vector of $a_{i,j}$. This matrix has $sm \in O(n^{3/4})$ rows and $t \in O(n^{3/4})$ columns.

Compute $\mathbf{B} := \mathbf{A}\,\mathbf{\Gamma}$; by definition of exponents $\omega(\cdot)$, this can be done in $O(n^{(3/4)\cdot\omega(4/3)})$ operations in F, and the rows of this matrix give all $a_{i,j}(\gamma)$. The last step is to write $\alpha_i(\gamma) = \sum_{0\leq j<m} a_{i,j}(\gamma)\gamma_t^j$. Using Horner's scheme, this takes $O(sm)$ operations in K, which is $\tilde{O}(n^{7/4})$ operations in F. Since $\omega(3/4) \geq 7/4$, the leading exponent in all costs so far is $(3/4)\cdot\omega(4/3)$. □

Lemma 3.2. *Given α in K, $g_1, \dots, g_r$ in G and positive integers $(s_1, \dots s_r)$ such that $\prod_{i=1}^r s_i = O(\sqrt{n})$ and $r \in O(\log(n))$, all*

$$g_1^{i_1} \cdots g_r^{i_r}(\alpha), \quad \textit{for } 0 \leq i_j \leq s_j,\ 1 \leq j \leq r$$

can be computed in $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in F.

Proof (Compare [18, Lemma 4].) For a given $m \in \{1, \dots, r\}$, suppose we have computed

$$G_{i_1,\dots,i_m} := g_m^{i_m} \cdots g_1^{i_1}(\alpha)$$

for $0 \leq i_j \leq s_j$ if $1 \leq j < m$, and $0 \leq i_m < k_m$, as well as the automorphism $\eta := g_m{}^{k_m}$ (by means of its value at ξ).

Then, we can obtain $G_{i_1,\dots,i_m}$ for $0 \leq i_j \leq s_j$ if $1 \leq j < m$, and $0 \leq i_m < 2k_m$, by computing $\eta(G_{i_1,\dots,i_m})$, for all indices $i_1, \dots, i_m$ available to us, that is, $0 \leq i_j \leq s_j$ if $1 \leq j < m$, and $0 \leq i_m < k_m$. This can be carried out using $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in F by applying Lemma 3.1. Prior to entering the next iteration, we compute η^2 by means of a modular composition, at negligible cost.

Using the above doubling method for g_m, we have to do $O(\log s_m)$ steps, for a total cost of $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in F. We repeat this procedure for $m = 1, \dots, r$; since r is in $O(\log(n))$, the cost remains $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$. □

We now present dual versions of the previous two lemmas. Viewed as an F-linear map, $g : \alpha \mapsto g(\alpha)$ admits a transpose, mapping an F-linear form $\ell : \mathsf{K} \to \mathsf{F}$ to the F-linear form $\ell \circ g : \alpha \mapsto \ell(g(\alpha))$. The *transposition principle* [6, 20] implies that if a linear map $\mathsf{F}^N \to \mathsf{F}^M$ can be computed in time T, its transpose can be computed in time $T + O(N + M)$. Given s linear forms $\ell_1, \dots, \ell_s$ and g in G, transposing Lemma 3.1 shows that we can compute $\ell_1 \circ g, \dots, \ell_s \circ g$ in time $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$.

Lemma 3.3. *Given* F*-linear forms* $\ell_1, \dots, \ell_s : \mathsf{K} \to \mathsf{F}$ *and* g *in* $G = \mathrm{Gal}(\mathsf{K}/\mathsf{F})$*, with* $s = O(\sqrt{n})$*, we can compute* $\ell_1 \circ g, \dots, \ell_s \circ g$ *in time* $\tilde{O}(n^{3/4\omega(4/3)})$.

Proof. Given ℓ_i by its values on the power basis $1, \xi, \dots, \xi^{n-1}$, $\ell_i \circ g$ is represented by its values at $1, \gamma, \dots, \gamma^{n-1}$, with $\gamma := g(\xi)$.

Let t, m and $\gamma_0, \dots, \gamma_t$ be as in Lemma 3.1. Compute the giant steps $\gamma_t^j = \gamma^{tj}$, $j = 0, \dots, m-1$ and for $i = 1, \dots, s$ and $j = 0, \dots, m-1$, deduce $L_{i,j}$ defined by $L_{i,j}(\alpha) := \ell_i(\gamma^{tj}\alpha)$ for α in K. Each of them is obtained by a *transposed multiplication* in time $\tilde{O}(n)$ [32, §4.1], so that the total cost thus far is $\tilde{O}(n^{7/4})$.

Finally, multiply the $(sm \times n)$ matrix with entries the coefficients of all $L_{i,j}$ (as rows) by the $(n \times t)$ matrix with entries the coefficients of $\gamma_0, \dots, \gamma_{t-1}$ (as columns) to get all $\ell_i(\gamma^j)$, for $i = 1, \dots, s$ an $j = 0, \dots, n-1$. This is done in time $O(n^{(3/4)\cdot\omega(4/3)})$. □

From this, we deduce the transposed version of Lemma 3.2, whose proof follows the same pattern.

Lemma 3.4. *Given* $\ell : \mathsf{K} \to F, g_1, \dots, g_r$ *in* G *and positive integers* $(s_1, \dots s_r)$ *such that* $\prod_{i=1}^r s_i = O(\sqrt{n})$ *and* $r \in O(\log(n))$*, all*

$$\ell \circ g_1^{i_1} \cdots g_r^{i_r}, \quad \textit{for}\ 0 \le i_j \le s_j,\ 1 \le j \le r,$$

can be computed in $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ *operations in* F.

Proof (Compare [18, Lemma 8].) For $m = 1, \dots, r$, assume we know $L_{i_1,\dots,i_m} := \ell \circ (g_1^{i_1} \cdots g_m^{i_m})$, for $0 \le i_j \le s_j$ if $1 \le j < m$, and $0 \le i_m < k_m$. Using Lemma 3.3, we compute all $L_{i_1,\dots,i_m} \circ {g_m}^{k_m}$, which gives us $L_{i_1,\dots,i_m}$ for indices $0 \le i_m < 2k_m$. The analysis is as in Lemma 3.2. □

3.2 Abelian Groups

The first main result in this section is the following proposition. Assume G is an abelian group presented as

$$\langle g_1, \dots, g_r : g_1^{e_1} = \cdots = g_r^{e_r} = 1 \rangle,$$

where $e_i \in \mathbb{N}$ is the order of g_i and $n = e_1 \cdots e_r$. Without loss of generality, we assume $e_i \ge 2$ for all i, so that r is in $O(\log n)$. Elements of $\mathsf{F}[G]$ are written as polynomials $\sum_{i_1,\dots,i_r} c_{i_1,\dots,i_r} {g_1}^{e_1} \cdots {g_r}^{e_r}$, with $0 \le i_j < e_j$ for all j.

Proposition 3.5. *Suppose that* G *is abelian, with notation as above. For* α *in* K *and* $\ell : \mathsf{K} \to \mathsf{F}$*,* $s_{\alpha,\ell} \in \mathsf{F}[G]$*, as defined in* (2.4)*, is computable using* $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ *operations in* F.

Proof. Our goal is to compute

$$\ell(g_1^{i_1}, \dots, g_r^{i_r}(\alpha)),\ 1 \le j \le r, 0 \le i_j \le e_j, \tag{3.2}$$

where ℓ is an F-linear projection $\mathsf{K} \to \mathsf{F}$. For $1 \le i \le r$, define $s_i := \lceil \sqrt{e_i} \rceil$. As we sketched in the cyclic case, the elements in (3.2) can be expressed as $L_{j_1,\dots,j_r}(G_{i_1,\dots,i_r})$, for $1 \le m \le r, 0 \le i_m < s_m, 0 \le j_m < s_m$. Here, $L_{j_1,\dots,j_r} := \ell \circ (g_1^{s_1 j_1} \cdots g_r^{s_r j_s})$ are linear projections presented as row vectors and $G_{i_1,\dots,i_r} := g_1^{i_1} \cdots g_r^{i_r}(\alpha)$ are field elements presented as column vectors. Then, all elements in (3.2) can be computed with the following steps, the sum of whose costs proves the proposition.

Step 1. Apply Lemma 3.2 to get

$$G_{i_1,\dots,i_r} = g_1^{i_1} \cdots g_r^{i_r}(\alpha),\ 1 \le m \le r, 0 \le i_m < s_m,$$

with cost $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$.

Step 2. Compute all $g_i^{s_i}$, $i = 1, \dots, r$; using $O(\log(n))$ modular compositions. The cost is negligible compared to that of Step 1.

Step 3. Use Lemma 3.4 to compute

$$L_{j_1,\dots,j_r} = \ell \circ (g_1^{s_1 j_1} \cdots g_r^{s_r j_s}),\ 1 \le m \le r, 0 \le j_m < s_m,$$

with cost $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$

Step 4. Multiply the matrix with rows the coefficients of all $L_{j_1,\dots,j_r}$ by the matrix with columns the coefficients of all $G_{i_1,\dots,i_r}$; this yields all required values. We compute this product in $O(n^{(1/2)\cdot\omega(2)})$ operations in F, which is in $O(n^{(3/4)\cdot\omega(4/3)})$. □

3.3 Metacyclic Groups

A group G is metacyclic if it has a normal cyclic subgroup H such that G/H is cyclic; for instance, any group with a squarefree order is metacyclic. See [16, p. 88] or [8, p. 334] for more background. A metacyclic group can always be presented as

$$\langle \sigma, \tau : \sigma^m = 1, \tau^s = \sigma^t, \tau^{-1}\sigma\tau = \sigma^r \rangle, \tag{3.3}$$

for some integers m, t, r, s, with $r, t \le m$ and $r^s = 1 \bmod t, rt = t \bmod m$. For example, the dihedral group

$$D_{2m} = \langle \sigma, \tau : \sigma^m = 1, \tau^2 = 1, \tau^{-1}\sigma\tau = \sigma^{m-1} \rangle,$$

is metacyclic, with $s = 2$. Generalized quaternion groups, which can be presented as

$$Q_m = \langle \sigma, \tau : \sigma^{2m} = 1, \tau^2 = \sigma^m, \tau^{-1}\sigma\tau = \sigma^{2m-1} \rangle,$$

are metacyclic, with $s = 2$ as well.

Using the notation of (3.3), $n = |G|$ is equal to ms, and all elements in a metacyclic group can be presented uniquely as either

$$\{\sigma^i\tau^j,\ 0 \le i \le m-1,\ 0 \le j \le s-1\}, \text{ or} \tag{3.4}$$

$$\{\tau^j\sigma^i,\ 0 \le i \le m-1,\ 0 \le j \le s-1\}. \tag{3.5}$$

Accordingly, elements in the group algebra $\mathsf{F}[G]$ can be written as

$$\sum_{\substack{i<m \\ j<s}} c_{i,j}\sigma^i\tau^j \quad \text{or} \quad \sum_{\substack{i<m \\ j<s}} c'_{i,j}\tau^j\sigma^i.$$

Conversion between the two representations involves no operation in F, using the commutation relation $\sigma^k\tau^c = \tau^c\sigma^{kr^c}$ for $k, c \ge 0$.

Proposition 3.6. *Suppose that* G *is metacyclic. For* α *in* K *and* $\ell : \mathsf{K} \to \mathsf{F}$*,* $s_{\alpha,\ell} \in \mathsf{F}[G]$ *is computable in time* $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$.

Proof. Suppose first that $s \le m$; then, we use the presentation (3.4) of the elements of G. Take α in K and $\ell : \mathsf{K} \to \mathsf{F}$; the goal is to compute $\ell(\sigma^i\tau^j(\alpha))$, for all $0 \le i < m$ and $0 \le j < s$. This is accomplished with the following steps.

Step 1. Apply Lemma 3.2 to compute

$$G_{i,j} := \sigma^i\tau^j(\alpha),\ 0 \le i < \lceil \sqrt{m/s} \rceil,\ 0 \le j < s.$$

Note that $\lceil \sqrt{m/s} \rceil s \le \lceil \sqrt{sm} \rceil \in O(\sqrt{n})$, so we can apply the lemma. This takes $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in F.

Step 2. Compute $\sigma^{\lceil \sqrt{m/s} \rceil}$, in $O(\log(n))$ modular compositions in degree n. The cost is no more than that of Step 1.

Step 3. Compute

$$L_k := \ell \circ \sigma^{k\lceil\sqrt{m/s}\rceil},\ \ 0 \le k < \lceil\sqrt{sm}\rceil,$$

using Lemma 3.4. This takes $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in F.

Step 4. At this point, we compute all

$$L_k(s_{i,j}) = \ell(\sigma^{k\lceil\sqrt{m/s}\rceil+i}\tau^j(\alpha)),$$

for $0 \le k < \lceil\sqrt{sm}\rceil$, $0 \le i < \lceil\sqrt{m/s}\rceil$ and $0 \le j < s$; these are precisely the values we needed.

This can be carried out by multiplying the matrix with rows the coefficients of all L_k by the matrix with columns the coefficients of all $G_{i,j}$; this yields all required values, as pointed out above. There are $O(\sqrt{sm}) = O(\sqrt{n})$ linear forms L_k's, and $O(\sqrt{n})$ field elements $G_{i,j}$'s, so we can compute this product in $O(n^{(1/2)\cdot\omega(2)})$ operations in F, which is $O(n^{(3/4)\cdot\omega(4/3)})$.

This concludes the proof in the case $s \le m$. When $m \le s$, use the presentation (3.5) of the elements of G and proceed as above. □

4 TESTING INVERTIBILITY

In this section we consider the problem of invertibility testing in $\mathsf{F}[G]$, specifically for abelian and metacyclic groups G: given an element β in $\mathsf{F}[G]$, for a field F and a group G, determine whether β is a unit in $\mathsf{F}[G]$. Since we are in characteristic zero, Wedderburn's theorem implies the existence of an F-algebra isomorphism (which we will refer to as a Fourier Transform)

$$\mathsf{F}[G] \to M_{d_1}(D_1) \times \cdots \times M_{d_r}(D_r),$$

where all D_i's are division algebras over F. If we were working over $\mathsf{F} = \mathbb{C}$, all D_i's would simply be $\mathbb{C}$ itself. A natural solution to test the invertibility of $\beta \in \mathsf{F}[G]$ would then be to compute its Fourier transform and test whether all its components $\beta_1 \in M_{d_1}(\mathbb{C}), \dots, \beta_r \in M_{d_r}(\mathbb{C})$ are invertible. This boils down to linear algebra over $\mathbb{C}$, and takes $O(d_1^\omega + \cdots + d_r^\omega)$ operations. Since $d_1^2 + \cdots + d_r^2 = |G|$, this is $O(|G|^{\omega/2})$ operations in $\mathbb{C}$.

However, we do not wish to make such a strong assumption as $\mathsf{F} = \mathbb{C}$. Since we measure the cost of our algorithms in F-operations, the direct approach that embeds $\mathsf{F}[G]$ into $\mathbb{C}[G]$ does not make it possible to obtain a subquadratic cost in general. If, for instance, $\mathsf{F} = \mathbb{Q}$ and G is cyclic of order $n = 2^k$, computing the Fourier Transform of β requires we work in a degree $n/2$ extension of $\mathbb{Q}$, implying a quadratic runtime.

We give algorithms for the problem of invertibility testing for the families of groups seen so far, abelian and metacyclic. For the former, we prove a stronger result: starting from a suitable presentation of G, we give a softly linear-time algorithm to find an isomorphic image of $\beta \in \mathsf{F}[G]$ in a product of F-algebras of the form $\mathsf{F}[z]/\langle P_i(z)\rangle$, for certain polynomials $P_i \in \mathsf{F}[z]$ (recovering β from its image is softly-linear time as well). Not only does this allow us to test whether β is invertible, this would also make it possible to find its inverse in $\mathsf{F}[G]$ in softly-linear time. For metacyclic groups, we describe an injective F-algebra homomorphism from $\mathsf{F}[G]$ to a matrix algebras over a cyclotomic ring. The codomain is in general of dimension higher than $|G|$, so the algorithm we deduce from this is not linear-time.

4.1 Abelian groups

Because an abelian group is a product of cyclic groups, its group algebra $\mathsf{F}[G]$ is the tensor product of cyclic algebras, so it admits a description of the form $\mathsf{F}[x_1, \dots, x_t]/\langle x_1^{n_1}-1, \dots, x_t^{n_t}-1\rangle$, for some integers $n_1, \dots, n_t$. The complexity of arithmetic operations in an F-algebra such as $\mathbb{A} := \mathsf{F}[x_1, \dots, x_t]/\langle P_1(x_1), \dots, P_t(x_t)\rangle$ is difficult to pin down precisely. For general P_i's, the cost of multiplication in $\mathbb{A}$ is known to be $O(\dim(\mathbb{A})^{1+\varepsilon})$, for any $\varepsilon > 0$ [26, Theorem 2]. From this it may be possible to deduce similar upper bounds on the complexity of invertibility tests, following [9], but this seems non-trivial.

Instead, we give an algorithm with softly linear runtime, that uses the factorization properties of cyclotomic polynomials and Chinese remaindering techniques to transform our problem into that of invertibility testing in algebras of the form $\mathsf{F}[z]/\langle P_i(z)\rangle$, for various polynomials P_i. The reference [29] also discusses the factors of algebras such as $\mathsf{F}[x_1, \dots, x_t]/\langle x_1^{n_1}-1, \dots, x_t^{n_t}-1\rangle$, but the resulting algorithms are different (and the cost of the Poli's 1994 algorithm is only known to be polynomial in $|G|$).

Tensor product of two cyclotomic rings: coprime orders. The following proposition will be the key to foregoing multivariate polynomials, and replacing them by univariate ones. Let m, m' be two coprime integers and define

$$\mathbf{h} := \mathsf{F}[x, x']/\langle \Phi_m(x), \Phi_{m'}(x')\rangle,$$

where for $i \ge 0$, Φ_i is the cyclotomic polynomial of order i. In what follows, φ is Euler's totient function, so that $\varphi(i) = \deg(\Phi_i)$ for all i.

Lemma 4.1. *There exists an F-algebra isomorphism $\gamma : \mathbf{h} \to \mathsf{F}[z]/\langle\Phi_{mm'}(z)\rangle$ given by $xx' \mapsto z$. Given Φ_m and $\Phi_{m'}$, $\Phi_{mm'}$ can be computed in time $\tilde{O}(\varphi(mm'))$; given these polynomials, one can apply γ and its inverse to any input using $\tilde{O}(\varphi(mm'))$ operations in F.*

Proof. Without loss of generality, we prove the first claim over $\mathbb{Q}$; the result over F follows by scalar extension. In the field $\mathbb{Q}[x, x']/\langle\Phi_m(x), \Phi_{m'}(x')\rangle$, xx' is cancelled by $\Phi_{mm'}$. Since this polynomial is irreducible, it is the minimal polynomial of xx', which is thus a primitive element for $\mathbb{Q}[x, x']/\langle\Phi_m(x), \Phi_{m'}(x')\rangle$. This proves the first claim.

For the second claim, we first determine the images of x and x' by γ. Start from a Bézout relation $am + a'm' = 1$, for some a, a' in $\mathbb{Z}$. Since $x^m = x'^{m'} = 1$ in $\mathbf{h}$, we deduce that $\gamma(x) = z^u$ and $\gamma(x') = z^v$, with $u := am \bmod mm'$ and $v := a'm' \bmod mm'$. To compute $\gamma(P)$, for some P in $\mathbf{h}$, we first compute $P(z^u, z^v)$, keeping all exponents reduced modulo mm'. This requires no arithmetic operations and results in a polynomial $\bar{P}$ of degree less than mm', which we eventually reduce modulo $\Phi_{mm'}$ (the latter is obtained by the composed product algorithm of [4] in quasi-linear time). By [3, Theorem 8.8.7], we have the bound $s \in O(\varphi(s)\log(\log(s)))$, so that s is in $\tilde{O}(\varphi(s))$. Thus, we can reduce $\bar{P}$ modulo $\Phi_{mm'}$ in $\tilde{O}(\varphi(mm'))$ operations, establishing the cost bound for γ.

Conversely, given Q in $\mathsf{F}[z]/\langle\Phi_{mm'}(z)\rangle$, we obtain its preimage by replacing powers of z by powers of xx', reducing all exponents in x modulo m, and all exponents in x' modulo m'. We then reduce the result modulo both $\Phi_m(x)$ and $\Phi_{m'}(x')$. By the same argument as above, the cost is softly linear in $\varphi(mm')$. □

Extension to several cyclotomic rings. The natural generalization of the algorithm above starts with pairwise distinct primes

$\boldsymbol{p} = (p_1, \dots, p_t)$, non-negative exponent $\boldsymbol{c} = (c_1, \dots, c_t)$ and variables $\boldsymbol{x} = (x_1, \dots, x_t)$ over F. Now, we define

$$\mathbb{H} := \mathsf{F}[x_1, \dots, x_t]/\langle \Phi_{{p_1}^{c_1}}(x_1), \dots, \Phi_{{p_t}^{c_t}}(x_t)\rangle;$$

when needed, we will write $\mathbb{H}$ as $\mathbb{H}_{\boldsymbol{p},\boldsymbol{c},\boldsymbol{x}}$. Finally, we let $\mu := {p_1}^{c_1} \cdots {p_t}^{c_t}$; then, the dimension $\dim(\mathbb{H})$ is $\varphi(\mu)$.

Lemma 4.2. *There exists an F-algebra isomorphism $\Gamma : \mathbb{H} \to \mathsf{F}[z]/\langle \Phi_\mu(z)\rangle$ given by $x_1 \cdots x_t \mapsto z$. One can apply Γ and its inverse to any input using $\tilde{O}(\dim(\mathbb{H}))$ operations in F.*

Proof. We proceed iteratively. First, note that the cyclotomic polynomials $\Phi_{{p_i}^{c_i}}$ can all be computed in time $O(\varphi(\mu))$. The isomorphism $\gamma : \mathsf{F}[x_1, x_2]/\langle \Phi_{{p_1}^{c_1}}(x_1), \Phi_{{p_2}^{c_2}}(x_2)\rangle \to \mathsf{F}[z]/\langle \Phi_{{p_1}^{c_1}{p_2}^{c_2}}(z)\rangle$ given in the previous paragraph extends coordinate-wise to an isomorphism

$$\Gamma_1 : \mathbb{H} \to \mathsf{F}[z, x_3, \dots, x_t]/\langle \Phi_{{p_1}^{c_1}{p_2}^{c_2}}(z), \Phi_{{p_3}^{c_3}}(x_3), \dots, \Phi_{{p_t}^{c_t}}(x_t)\rangle.$$

By the previous lemma, Γ_1 and its inverse can be applied to any input in time $\tilde{O}(\varphi(\mu))$. Iterate this process another $t-2$ times, to obtain Γ as a product $\Gamma_{t-1} \circ \cdots \circ \Gamma_1$. Since t is logarithmic in $\varphi(\mu)$, the proof is complete. □

Tensor product of two prime-power cyclotomic rings, same p. We now consider cyclotomic polynomials of prime power orders for a common prime p. As above, we start with two such polynomials. Let thus p be a prime. The key to the following algorithms is the lemma below. Let c, c' be positive integers, with $c \geq c'$, and let x, y be indeterminates over F. Define

$$\mathbf{I} := \mathsf{F}[x]/\Phi_{p^c}(x), \tag{4.1}$$
$$\mathbf{J} := \mathsf{F}[x, y]/\langle \Phi_{p^c}(x), \Phi_{p^{c'}}(y)\rangle = \mathbf{I}[y]/\Phi_{p^{c'}}(y). \tag{4.2}$$

Note $\mathbf{I}$ and $\mathbf{J}$ have respective dimensions $\varphi(p^c)$ and $\varphi(p^c)\varphi(p^{c'})$.

Lemma 4.3. *There is an F-algebra isomorphism $\theta : \mathbf{J} \to \mathbf{I}^{\varphi(p^{c'})}$ such that one can apply θ or its inverse to any inputs using $\tilde{O}(\dim(\mathbf{J}))$ operations in F.*

Proof. Let ξ be the residue class of x in $\mathbb{A}$. Then, in $\mathbf{I}[y]$, $\Phi_{p^{c'}}(y)$ factors as

$$\Phi_{p^{c'}}(y) = \prod_{\substack{1 \leq i \leq p^{c'}-1 \\ \gcd(i,p)=1}} (y - \rho_i),$$

with $\rho_i := \xi^{ip^{c-c'}}$ for all i. Even though $\mathbf{I}$ may not be a field, the Chinese Remainder theorem implies that $\mathbf{J}$ is isomorphic to $\mathbf{I}^{\varphi(p^{c'})}$; the isomorphism is given by

$$\begin{array}{cccc} \theta : & \mathbf{J} & \to & \mathbf{I} \times \cdots \times \mathbf{I}, \\ & P & \mapsto & (P(\xi, \rho_1), \dots, P(\xi, \rho_{\varphi(p^{c'})})). \end{array}$$

Arithmetic operations $(+, -, \times)$ in $\mathbf{I}$ can be done in $\tilde{O}(\varphi(p^c))$ operations in F. Starting from $\rho_1 \in \mathbf{I}$, all other roots ρ_i can then be computed in $O(\varphi(p^{c'}))$ operations in $\mathbf{I}$ or $\tilde{O}(\dim(\mathbf{J}))$ operations in F.

Applying θ and its inverse is done by means of fast evaluation and interpolation [11, Chapter 10] in $\tilde{O}(\varphi(p^{c'}))$ operations in $\mathbf{I}$, that is, $\tilde{O}(\deg(\mathbf{J}))$ operations in F (the algorithms do not require that $\mathbf{I}$ be a field). □

Extension to several cyclotomic rings. Let p be as before, and consider now non-negative integers $\boldsymbol{c} = (c_1, \dots, c_t)$ and variables $\boldsymbol{x} = (x_1, \dots, x_t)$. We define the F-algebra

$$\mathbb{A} := \mathsf{F}[x_1, \dots, x_t]/\langle \Phi_{p^{c_1}}(x_1), \dots, \Phi_{p^{c_t}}(x_t)\rangle,$$

which we will sometimes write $\mathbb{A}_{p,\boldsymbol{c},\boldsymbol{x}}$ to make the dependency on p and the c_i's clear. Up to reordering the c_i's, we can assume that $c_1 \geq c_i$ holds for all i, and define as before $\mathbf{I} := \mathsf{F}[x_1]/\Phi_{p^{c_1}}(x_1)$.

Lemma 4.4. *There exists an F-algebra isomorphism $\Theta : \mathbb{A} \to \mathbf{I}^{\dim(\mathbb{A})/\dim(\mathbf{I})}$. This isomorphism and its inverse can be applied to any inputs using $\tilde{O}(\dim(\mathbb{A}))$ operations in F.*

Proof. Without loss of generality, we can assume that all c_i's are non-zero (since for $c_i = 0$, $\Phi_{p^{c_i}}(x_i) = x_i - 1$, so $\mathsf{F}[x_i]/\langle \Phi_{p^{c_i}}(x_i)\rangle = \mathsf{F}$). We proceed iteratively. First, rewrite $\mathbb{A}$ as

$$\mathbb{A} = \mathbf{I}[x_2, x_3, \dots, x_t]/\langle \Phi_{p^{c_2}}(x_2), \Phi_{p^{c_3}}(x_3), \dots, \Phi_{{p_t}^{c_t}}(x_t)\rangle.$$

The isomorphism $\theta : \mathbf{I}[x_2]/\Phi_{p^{c_2}}(x_2) \to \mathbf{I}^{\varphi(p^{c_2})}$ introduced in the previous paragraph extends coordinate-wise to an isomorphism

$$\Theta_1 : \mathbb{A} \to (\mathbf{I}[x_3, \dots, x_t]/\langle \Phi_{p^{c_3}}(x_3), \dots, \Phi_{p^{c_t}}(x_t)\rangle)^{\varphi(p^{c_2})};$$

Θ_1 and its inverse can be evaluated in quasi-linear time $\tilde{O}(\dim(\mathbb{A}))$. We now work in all copies of $\mathbf{I}[x_3, \dots, x_t]/\langle \Phi_{p^{c_3}}(x_3), \dots, \Phi_{p^{c_t}}(x_t)\rangle$ independently, and apply the procedure above to each of them. Altogether we have $t-1$ such steps to perform, giving us an isomorphism

$$\Theta = \Theta_{t-1} \circ \cdots \circ \Theta_1 : \mathbb{A} \to \mathbf{I}^{\varphi(p^{c_2}) \cdots \varphi(p^{c_t})}.$$

The exponent can be rewritten as $\dim(\mathbb{A})/\dim(\mathbf{I})$, as claimed. All Θ_i's and their inverses can be computed in time $\tilde{O}(\dim(\mathbb{A}))$, and we do $t-1$ of them, where t is $O(\log(\dim(\mathbb{A})))$. □

Decomposing certain p-group algebras. The prime p and indeterminates $\boldsymbol{x} = (x_1, \dots, x_t)$ are as before; we now consider positive integers $\boldsymbol{b} = (b_1, \dots, b_t)$, and the F-algebra

$$\begin{array}{rcl} \mathbb{B} & := & \mathsf{F}[x_1, \dots, x_t]/\langle x_1^{p^{b_1}} - 1, \dots, x_t^{p^{b_t}} - 1\rangle \\ & = & \mathsf{F}[x_1]/\langle x_1^{p^{b_1}} - 1\rangle \otimes \cdots \otimes \mathsf{F}[x_t]/\langle x_t^{p^{b_t}} - 1\rangle. \end{array}$$

If needed, we will write $\mathbb{B}_{p,\boldsymbol{b},\boldsymbol{x}}$ to make the dependency on p and the b_i's clear. This is the F-group algebra of $\mathbb{Z}/p^{b_1}\mathbb{Z} \times \cdots \times \mathbb{Z}/p^{b_t}\mathbb{Z}$.

Lemma 4.5. *There exists a positive integer N, non-negative integers $\boldsymbol{c} = (c_1, \dots, c_N)$ and an F-algebra isomorphism*

$$\Lambda : \mathbb{B} \to \mathbb{D} = \mathsf{F}[z]/\langle \Phi_{p^{c_1}}(z)\rangle \times \cdots \times \mathsf{F}[z]/\langle \Phi_{p^{c_N}}(z)\rangle.$$

One can apply the isomorphism and its inverse to any input using $\tilde{O}(\dim(\mathbb{B}))$ operations in F.

Proof. For $i \leq t$, we have the factorization

$$x_i^{p^{b_i}} - 1 = \Phi_1(x_i)\Phi_p(x_i)\Phi_{p^2}(x_i) \cdots \Phi_{p^{b_i}}(x_i);$$

note that $\Phi_1(x_i) = x_i - 1$. The factors may not be irreducible, but are pairwise coprime, so we have a Chinese Remainder isomorphism

$$\lambda_i : \mathsf{F}[x_i]/\langle x_i^{p^{b_i}} - 1\rangle \to \mathsf{F}[x_i]/\langle \Phi_1(x_i)\rangle \times \cdots \times \mathsf{F}[x_i]/\langle \Phi_{p^{b_i}}(x_i)\rangle.$$

It and its inverse can be computed in $\tilde{O}(p^{b_i})$ operations in F [11, Chapter 10]. This gives an F-algebra isomorphism

$$\lambda : \mathbb{B} \to \prod_{c_1=0}^{b_1} \cdots \prod_{c_t=0}^{b_t} \mathbb{A}_{p,\boldsymbol{c},\boldsymbol{x}},$$

with $\boldsymbol{c} = (c_1, \dots, c_t)$. Together with its inverse, λ can be computed in $\tilde{O}(\dim(\mathbb{B}))$ operations in F. Composing with the result in Lemma 4.4, this gives us an isomorphism

$$\Lambda : \mathbb{B} \to \mathbb{D} := \prod_{c_1=0}^{b_1} \cdots \prod_{c_t=0}^{b_t} \mathbf{I}_{\boldsymbol{c}}^{D_{\boldsymbol{c}}},$$

where $\mathbf{I}_{\boldsymbol{c}} = \mathsf{F}[z]/\langle \Phi_{p^c}(z) \rangle$, with $c = \max(c_1, \dots, c_t)$ and $D_{\boldsymbol{c}} = \dim(\mathbb{A}_{t,\boldsymbol{c},\boldsymbol{x}})/\dim(\mathbf{I}_{\boldsymbol{c}})$. As before, Λ and its inverse can be computed in quasi-linear time $\tilde{O}(\dim(\mathbb{B}))$. □

As for $\mathbb{B}$, we will write $\mathbb{D}_{p,\boldsymbol{b},\boldsymbol{x}}$ if needed; it is well-defined, up to the order of the factors.

Main result. Let G be an abelian group. We can write the elementary divisor decomposition of G as $G = G_1 \times \cdots \times G_s$, where each G_i is of prime power order $p_i^{a_i}$, for pairwise distinct primes $p_1, \dots, p_s$, so that $|G| = p_1^{a_1} \cdots p_s^{a_s}$. Each G_i can itself be written as a product of cyclic groups, $G_i = G_{i,1} \times \cdots \times G_{i,t_i}$, where the factor $G_{i,j}$ is cyclic of order ${p_i}^{b_{i,j}}$, with $b_{i,1} \leq \cdots \leq b_{i,t_i}$ and $b_{i,1} + \cdots + b_{i,t_i} = a_i$. We henceforth assume that generators $\gamma_{1,1}, \dots, \gamma_{s,t_s}$ of respectively $G_{1,1}, \dots, G_{s,t_s}$ are known, and that elements of $\mathsf{F}[G]$ are given on the power basis in $\gamma_{1,1}, \dots, \gamma_{s,t_s}$.

Proposition 4.6. *Given $\beta \in \mathsf{F}[G]$, written in the basis $\gamma_{1,1}, \dots, \gamma_{s,t_s}$, one can test if β is a unit in $\mathsf{F}[G]$ in time $\tilde{O}(|G|)$.*

From the factorization $G = G_1 \times \cdots \times G_s$, we deduce that the group algebra $\mathsf{F}[G]$ is the tensor product $\mathsf{F}[G_1] \otimes \cdots \otimes \mathsf{F}[G_s]$. Furthermore, the factorization $G_i = G_{i,1} \times \cdots \times G_{i,t_i}$ implies that $\mathsf{F}[G_i]$ is isomorphic, as an F-algebra, to

$$\mathsf{F}[x_{i,1}, \dots, x_{i,t_i}] / \left\langle x_{i,1}^{p_i^{b_1}} - 1, \dots, x_{i,t_i}^{p_i^{b_{i,t_i}}} - 1 \right\rangle = \mathbb{B}_{p_i, \boldsymbol{b}_i, \boldsymbol{x}_i},$$

with $\boldsymbol{b}_i = (b_{i,1}, \dots, b_{i,t_i})$ and $\boldsymbol{x}_i = (x_{i,1}, \dots, x_{i,t_i})$. Given β on the power basis in $\gamma_{1,1}, \dots, \gamma_{s,t_s}$, we obtain its image B in $\mathbb{B}_{p_1,\boldsymbol{b}_1,\boldsymbol{x}_1} \otimes \cdots \otimes \mathbb{B}_{p_s,\boldsymbol{b}_s,\boldsymbol{x}_s}$ simply by renaming $\gamma_{i,j}$ as $x_{i,j}$, for all i, j.

For $i \leq s$, by Lemma 4.5, there exist integers $c_{i,1}, \dots, c_{i,N_i}$ such that $\mathbb{B}_{p_i,\boldsymbol{b}_i,\boldsymbol{x}_i}$ is isomorphic to an algebra $\mathbb{D}_{p_i,\boldsymbol{b}_i,z_i}$, with factors $\mathsf{F}[z_i]/\langle \Phi_{{p_i}^{c_{i,j}}}(z_i) \rangle$. By distributivity of the tensor product over direct products, we deduce that $\mathbb{B}_{p_1,\boldsymbol{b}_1,\boldsymbol{x}_1} \otimes \cdots \otimes \mathbb{B}_{p_s,\boldsymbol{b}_s,\boldsymbol{x}_s}$ is isomorphic to the product of algebras

$$\prod\nolimits_{\boldsymbol{j}} \mathsf{F}[z_1, \dots, z_s] / \langle \Phi_{{p_1}^{c_{1,j_1}}}(z_1), \dots, \Phi_{{p_s}^{c_{s,j_s}}}(z_s) \rangle, \tag{4.3}$$

for all indices $\boldsymbol{j} = (j_1, \dots, j_s)$, with $j_1 = 1, \dots, N_1, \dots, j_s = 1, \dots, N_s$; call Γ the isomorphism. Given B in $\mathbb{B}_{p_1,\boldsymbol{b}_1,\boldsymbol{x}_1} \otimes \cdots \otimes \mathbb{B}_{p_s,\boldsymbol{b}_s,\boldsymbol{x}_s}$, Lemma 4.5 also implies that $B' := \Gamma(B)$ can be computed in time $\tilde{O}(|G|)$ (apply the isomorphism corresponding to $\boldsymbol{x}_1$ coordinate-wise with respect to all other variables, then deal with $\boldsymbol{x}_2$, etc). The codomain in (4.3) is the product of all $\mathbb{H}_{\boldsymbol{p},\boldsymbol{c}_{\boldsymbol{j}},\boldsymbol{z}}$, with

$$\boldsymbol{p} = (p_1, \dots, p_s), \quad \boldsymbol{c} = (c_{1,j_1}, \dots, c_{s,j_s}), \quad \boldsymbol{z} = (z_1, \dots, z_s).$$

Apply Lemma 4.2 to all $\mathbb{H}_{\boldsymbol{p},\boldsymbol{c}_{\boldsymbol{j}},\boldsymbol{z}}$ to obtain an F-algebra isomorphism

$$\Gamma' : \prod\nolimits_{\boldsymbol{j}} \mathbb{H}_{\boldsymbol{p},\boldsymbol{c}_{\boldsymbol{j}},\boldsymbol{z}} \to \prod\nolimits_{\boldsymbol{j}} \mathsf{F}[z]/\langle \Phi_{d_{\boldsymbol{j}}}(z) \rangle,$$

for certain integers $d_{\boldsymbol{j}}$. The lemma implies that given B', $B'' := \Gamma'(B')$ can be computed in softly linear time $\tilde{O}(|G|)$ as well. Invertibility of $\beta \in \mathsf{F}[G]$ is equivalent to A'' being invertible, that is, to all its components being invertible in the respective factors $\mathsf{F}[z]/\langle \Phi_{d_{\boldsymbol{j}}}(z) \rangle$. Invertibility in such an algebra can be tested in softly linear time by applying the fast extended GCD algorithm [11, Chapter 11], so our conclusion follows. With Proposition 3.5, this proves the first part of Theorem 1.1.

4.2 Metacyclic Groups

We next study the invertibility problem for a metacyclic group G. We use an injective homomorphism, whose image will be easy to compute. This is the object of the following lemma, where the map is inspired by the one used in [8, §47].

Assume that $G = \langle \sigma, \tau : \sigma^m = 1, \tau^s = \sigma^t, \tau^{-1}\sigma\tau = \sigma^r \rangle$, where $r^s = 1 \bmod m$ and $rt = t \bmod m$; in particular, $n = |G|$ is equal to ms. Define $\mathbb{A} := \mathsf{F}[z]/\langle z^m - 1 \rangle$ and let ζ be the image of z in $\mathbb{A}$.

Lemma 4.7. *The mapping $\psi : \mathsf{F}[G] \to M_s(\mathbb{A})$ where*

$$\sigma \mapsto \mathrm{Diag}(\zeta, \zeta^r, \dots, \zeta^{r^{s-1}}), \ \tau \mapsto \left[\begin{array}{c|c} 0 & \zeta \\ \hline \mathbf{I}_{s-1} & 0 \end{array}\right]$$

is an injective F-algebra homomorphism.

Proof. It is straightforward to verify that $\psi(\sigma)^m = \mathbf{I}_m$, $\psi(\tau)^s = \psi(\sigma)^t$ and $\psi_i(\sigma)\psi_i(\tau) = \psi(\tau)\psi_i(\sigma)^r$; this shows that ψ is a well-defined F-algebras homomorphism.

Take $\beta \in \mathsf{F}[G]$, and write it $\beta = \sum_{j=0}^{s-1} \left(\sum_{i=0}^{m-1} b_{i,j}\sigma^i \right) \tau^j$. For $j = 0, \dots, s-1$, define $F_j(x) := \sum_{i=0}^{m-1} b_{i,j}x^i \in \mathsf{F}[x]$ and, for $1 \leq i, j \leq s$, $F_{i,j} := F_{i-1}(\zeta^{r^{j-1}})$. Then, $\psi(\beta)$ is the matrix

$$\begin{bmatrix} F_{1,1} & \cdots & \zeta F_{3,s-1} & \zeta F_{2,1} \\ F_{2,2} & F_{1,2} & \cdots & \zeta F_{3,s} \\ \vdots & \ddots & \ddots & \vdots \\ F_{s,s} & \cdots & F_{2,s} & F_{1,s} \end{bmatrix}. \tag{4.4}$$

If β is in $\mathrm{Ker}(\psi)$, we get $F_i(\zeta) = 0$, that is, $F_i \bmod (z^m - 1) = 0$, for $0 \leq i < s$. All F_i's have degree less than m, so they are all zero. □

We finally give two algorithms that test whether $\psi(\beta) \in M_s(\mathbb{A})$ is invertible, for a given β in $\mathsf{F}[G]$. Minor difficulties will arise as we work over $\mathbb{A}$, since $\mathbb{A}$ is not a field, but a product of fields (if the irreducible factorization of $z^m - 1$ in $\mathsf{F}[z]$ is known, we can use the Chinese Remainder theorem and work in field extensions of F).

Corollary 4.8. *Given β in $\mathsf{F}[G]$, one can test if β is a unit in $\mathsf{F}[G]$ either by a deterministic algorithm that uses $\tilde{O}(s^{2.7}m)$ operations in F, or a Monte Carlo one that uses $\tilde{O}(n^2)$ operations in F.*

The second statement provides the last part of the proof of Theorem 1.1. Note that the first algorithm gives a better cost in many cases. For instance, if $s \leq m$, the first algorithm uses $O(n^{1.85})$ operations in F. This happens if s is prime, since then the number $(m - \gcd(m, r-1))/s$ is a positive integer, which implies $s \leq m$ (see [8, Theorem 47.12, Corollary 47.14]).

First algorithm. The first algorithm uses fast linear algebra algorithms over the ring $\mathbb{A}$. Here, we start from β written as $\beta = \sum_{j=0}^{s-1} \left(\sum_{i=0}^{m-1} b_{i,j}\sigma^i \right) \tau^j \in \mathsf{F}[G]$. Then, the proof of the previous lemma shows an explicit formula for $\psi(\beta)$. In order to compute this matrix, we note that $\zeta^{r^{j-1}} = \zeta^{r^{j-1} \bmod m}$; computing this element and its powers requires no arithmetic operation, so that the coefficients of each $F_{i,j}$ are obtained in linear time $O(m)$. Hence the matrix $\psi(\beta)$ can be computed in time $O(s^2 m)$.

Next, we have to determine whether $\psi(\beta)$ is a unit (the injectivity of ψ implies that this is the case if and only if β itself is a unit). This amounts to computing the determinant of this matrix, which can be done in $\tilde{O}(s^{2.7}m)$ operations in F, using the determinant algorithm of [19, Section 6]. □

Lemma 4.9. *Given β in $\mathsf{F}[G]$ and $\boldsymbol{v}$ in $\mathbb{A}^s$, one can compute $\psi(\beta)\boldsymbol{v} \in \mathbb{A}^s$ using $\tilde{O}(sm^2)$ operations in* F.

Proof. We use the basis of $\mathsf{F}[G]$ of (3.5), writing $\beta = \sum_{i=0}^{m-1} \left(\sum_{j=0}^{s-1} b_{i,j}\tau^i \right) \sigma^j = \sum_{i=0}^{m-1} B_i(\tau)\sigma^i$, for some $B_0, \ldots, B_{m-1}$ in $\mathsf{F}[z]$ of degree less than s.

Given $\boldsymbol{v}$ as above, we compute all $B_i(\psi(\tau))\psi(\sigma)^i\boldsymbol{v}$ independently, and add them to obtain $\psi(\beta)\boldsymbol{v}$. Hence, let us fix an index i in $\{0, \ldots, m-1\}$. The vector $\psi(\sigma)^i\boldsymbol{v}$ can be obtained by multiplying each entry of $\boldsymbol{v}$ by a power of ζ; this takes $\tilde{O}(sm)$ operations in F. Then, since $\psi(\tau)$ is the matrix of multiplication by y in $\mathbb{A}[y]/\langle y^s - \zeta\rangle$, $B_i(\psi(\tau))$ is the matrix of multiplication by $B_i(y)$ in $\mathbb{A}[y]/\langle y^s - \zeta\rangle$. Thus, applying this matrix to a vector also takes time $\tilde{O}(sm)$. Adding a factor of m to account for all indices i gives the result. □

Second algorithm for Corollary 4.8. The second algorithm uses Wiedemann's 1986 algorithm, and its extension by Kaltofen and Saunders [17]. Extra care will be needed to accommodate the fact that $\mathbb{A}$ has zero-divisors. Let $F_1, \ldots, F_s$ be the (unknown) irreducible factors of $z^m - 1$ in $\mathsf{F}[z]$ and define $\mathbb{A}_i := \mathsf{F}[z]/\langle F_i\rangle$ for $i = 1, \ldots, s$. We write $\pi_i : \mathbb{A} \to \mathbb{A}_i$ for the canonical projection, and extend the notation to matrices over $\mathbb{A}$.

For β in $\mathsf{F}[G]$, $\mathbf{M} := \psi(\beta)$ is invertible if and only if all $\mathbf{M}_i := \pi_i(\mathbf{M})$ are. We are going to use the algorithm of [17, Section 4] to compute the rank of all these matrices (these ranks are well-defined, since all $\mathbb{A}_i$'s are fields). Let $\mathbf{L}$ and $\mathbf{U}$ be respectively random lower triangular and upper triangular Toeplitz matrices over $\mathbb{A}$, and define $\mathbf{M}' := \mathbf{LMU} \in M_s(\mathbb{A})$. Finally, let $\mathbf{M}''$ be $\mathbf{M}'$, to which we adjoin a bottom row and a rightmost column of zeros (so it has size $s + 1$), let $\mathbf{M}''_i := \pi_i(\mathbf{M}'')$ and let $r_i := \mathrm{rank}(\mathbf{M}''_i)$, $i = 1, \ldots, s$. Then, all r_i's are less than $s + 1$, and $\mathbf{M}$ is invertible if and only if $r_i = s$ for all i.

The condition that $\mathbf{M}''_i$ has rank less than $s + 1$ makes it possible to apply [17, Lemma 2]: for generic $\boldsymbol{u}_i, \boldsymbol{v}_i$ in $\mathbb{A}_i^{s+1}$ and diagonal matrix $\mathbf{X}$ in $M_{s+1}(\mathbb{A}_i)$, the minimal polynomial of the sequence $(\boldsymbol{u}_i^T(\mathbf{M}''_i\mathbf{X}_i)^j\boldsymbol{v}_i)_{j\geq 0}$ has degree $r_i + 1$.

To compute these degrees without knowing the factorization $z^m - 1 = F_1 \cdots F_s$, we choose random $\boldsymbol{u}, \boldsymbol{v}$ in $\mathbb{A}^{s+1}$ and diagonal matrix $\mathbf{X}$ in $M_{s+1}(\mathbb{A})$. Then, we compute $2s$ terms in the sequence $(\gamma_j)_{j\geq 0}$, with $\gamma_j := \boldsymbol{u}^T(\mathbf{M}''\mathbf{X})^j\boldsymbol{v}$. Since multiplication by $\mathbf{L}$, $\mathbf{U}$ and $\mathbf{X}$ all take quasi-linear time $\tilde{O}(sm)$, Lemma 4.9 shows that one product by $\mathbf{M}''\mathbf{X}$ takes $\tilde{O}(sm^2)$ operations in F. Hence, all required terms can be obtained in $\tilde{O}(s^2m^2) = \tilde{O}(n^2)$ operations in F.

Finally, we apply the fast Euclidean algorithm to $\sum_{j=0}^{2s-1} \gamma_j y^j$ and y^{2s} in the ring $\mathbb{A}[y]$ to find the ranks $r_1, \ldots, r_s$. Since $\mathbb{A}$ is not a field, we rely on the algorithm of [1, 9]. Using $\tilde{O}(sm)$ operations in F, it reveals a partial factorization of $z^m - 1$ as $G_1 \cdots G_t$ (the factors may not be irreducible) and integers ρ_j, $j = 1, \ldots, t$, such that for all $i \leq s$, $j \leq t$, if F_i divides G_j, then $r_i = \rho_j$. This allows us to determine all r_i's, and thus decide whether $\psi(\beta)$ is singular. □

REFERENCES

[1] C. J. Accettella, G. M. Del Corso, and G. Manzini. 2003. Inversion of two level circulant matrices over $\mathbb{Z}_p$. *Lin. Alg. Appl.* 366 (2003), 5 – 23.

[2] D. Augot and P. Camion. 1994. A deterministic algorithm for computing a normal basis in a finite field. In *Proc. EUROCODE'94*, P. Charpin (Ed.).

[3] E. Bach and J. Shallit. 1996. *Algorithmic Number Theory, Volume 1: Efficient Algorithms.* MIT Press, Cambridge, MA.

[4] A. Bostan, P. Flajolet, B. Salvy, and É. Schost. 2006. Fast computation of special resultants. *J. Symbolic Comput.* 41, 1 (2006), 1–29.

[5] R. P. Brent and H. T. Kung. 1978. Fast algorithms for manipulating formal power series. *Journal of the Association for Computing Machinery* 25, 4 (1978), 581–595.

[6] J. Canny, E. Kaltofen, and Y. Lakshman. 1989. Solving systems of non-linear polynomial equations faster. In *ISSAC'89*. ACM, 121–128.

[7] M. Clausen and M. Müller. 2004. Generating fast Fourier transforms of solvable groups. *J. Symbolic Comput.* 37, 2 (2004), 137–156.

[8] C. Curtis and I. Reiner. 1988. *Representation theory of finite groups and associative algebras.* John Wiley & Sons, Inc., New York, New York. xiv+689 pages.

[9] X. Dahan, M. Moreno Maza, É. Schost, and Y. Xie. 2006. On the complexity of the D5 principle. In *Proc. of* Transgressive Computing 2006. Granada, Spain.

[10] S. Gao, J. von zur Gathen, D. Panario, and V. Shoup. 2000. Algorithms for exponentiation in finite fields. *Journal of Symbolic Computation* 29, 6 (2000), 879–889.

[11] J. von zur Gathen and J. Gerhard. 2013. *Modern Computer Algebra (third edition).* Cambridge University Press, Cambridge, U.K.

[12] J. von zur Gathen and M. Giesbrecht. 1990. Constructing normal bases in finite fields. *J. Symbolic Comput.* 10, 6 (1990), 547–570.

[13] J. von zur Gathen and V. Shoup. 1992. Computing Frobenius maps and factoring polynomials. *Computational Complexity* 2, 3 (1992), 187–224.

[14] K. Girstmair. 1999. An algorithm for the construction of a normal basis. *J. Number Theory* 78, 1 (1999), 36–45.

[15] A. Jamshidpey, N. Lemire, and É. Schost. 2018. Algebraic construction of quasi-split algebraic tori. *ArXiv:* 1801.09629 (2018).

[16] D. L. Johnson. 1976. *Presentations of Groups.* Cambridge University Press, Cambridge-New York-Melbourne. v+204 pages. London Mathematical Society Lecture Notes Series, No. 22.

[17] E. Kaltofen and D. Saunders. 1991. On Wiedemann's method of solving sparse linear systems. In *AAECC-9 (LNCS)*, Vol. 539. Springer Verlag, 29–38.

[18] E. Kaltofen and V. Shoup. 1998. Subquadratic-time factoring of polynomials over finite fields. *Math. Comp.* 67, 223 (1998), 1179–1197.

[19] E. Kaltofen and G. Villard. 2004. On the complexity of computing determinants. *Computational Complexity* 13, 3-4 (2004), 91–130.

[20] M. Kaminski, D.G. Kirkpatrick, and N.H. Bshouty. 1988. Addition requirements for matrix and transposed matrix products. *J. Algorithms* 9, 3 (1988), 354–364.

[21] K. Kedlaya and C. Umans. 2011. Fast polynomial factorization and modular composition. *SICOMP* 40, 6 (2011), 1767–1802.

[22] S. Lang. 2002. *Algebra* (third ed.). Graduate Texts in Mathematics, Vol. 211. Springer-Verlag, New York. xvi+914 pages.

[23] F. Le Gall. 2014. Powers of tensors and fast matrix multiplication. In *ISSAC'14*. ACM, Kobe, Japan, 296–303.

[24] F. Le Gall and F. Urrutia. 2018. Improved rectangular matrix multiplication using powers of the Coppersmith-Winograd tensor. In *SODA '18*. SIAM, New Orleans, USA, 1029–1046.

[25] H. W. Lenstra, Jr. 1991. Finding isomorphisms between finite fields. *Math. Comp.* 56, 193 (1991), 329–347.

[26] X. Li, M. Moreno Maza, and É. Schost. 2009. Fast arithmetic for triangular sets: from theory to practice. *J. Symb. Comp.* 44, 7 (2009), 891–907.

[27] G. Lotti and F. Romani. 1983. On the asymptotic complexity of rectangular matrix multiplication. *Theoretical Computer Science* 23, 2 (1983), 171–185.

[28] D. Maslen, D. N. Rockmore, and S. Wolff. 2018. The efficient computation of Fourier transforms on semisimple algebras. *J. Fourier Anal. Appl.* 24, 5 (2018), 1377–1400.

[29] A. Poli. 1994. A deterministic construction for normal bases of abelian extensions. *Comm. Algebra* 22, 12 (1994), 4751–4757.

[30] H. Schlickewei and S. Stepanov. 1993. Algorithms to construct normal bases of cyclic number fields. *J. Number Theory* 44, 1 (1993), 30–40.

[31] A. Schönhage and V. Strassen. 1971. Schnelle Multiplikation großer Zahlen. *Computing* 7 (1971), 281–292.

[32] V. Shoup. 1995. A new polynomial factorization algorithm and its implementation. *J. Symbolic Comput.* 20, 4 (1995), 363–397.

[33] D. Wiedemann. 1986. Solving sparse linear equations over finite fields. *IEEE Transactions on Information Theory* IT-32 (1986), 54–62.