# Quadratic Probabilistic Algorithms for Normal Bases

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

It is well known that for any finite Galois extension field $\mathsf{K}/\mathsf{F}$, 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 $\mathsf{F}$-basis of $\mathsf{K}$. Such an element $\alpha$ is called *normal* and $G \cdot \alpha$ is called a normal basis. In this paper 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. In an algebraic model, the cost of our algorithm is quadratic in the size of $G$ for metacyclic $G$ and slightly subquadratic for abelian $G$.

**ACM Reference format:**
Mark Giesbrecht, Armin Jamshidpey, and Éric Schost. 2019. Quadratic Probabilistic Algorithms for Normal Bases. In *Proceedings of International Symposium on Symbolic and Algebraic Computation, Beijing, 2019 (ISSAC)*, 11 pages.
https://doi.org/10.475/123_4

## 1 INTRODUCTION

For a finite Galois extension field $\mathsf{K}/\mathsf{F}$, with Galois group $G = \mathrm{Gal}(\mathsf{K}/\mathsf{F})$, an element $\alpha \in \mathsf{K}$ is called *normal* if the set of its Galois conjugates $G \cdot \alpha = \{\sigma(\alpha) : \sigma \in G\}$ forms a basis for $\mathsf{K}$ as a vector space over $\mathsf{F}$. The existence of normal element for any finite Galois extension is classical, and constructive proofs are provided in most algebra texts (see, e.g., [Lang 2002], Section 6.13).

While there is a wide range of well-known applications of normal bases in finite fields, such as fast exponentiation [Gao et al. 2000], 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 [Jamshidpey et al. 2018].

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 [1990], 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 [1998], with a cost of $O(n^{1.815} \log q)$ operations in $\mathbb{F}_q$. In the bit complexity model, Kedlaya and Umans showed how to reduce the exponent of $n$ to $1.5 + \varepsilon$ (for any $\varepsilon > 0$), by leveraging their quasi-linear time algorithm for *modular composition* [Kedlaya and Umans 2011]. Lenstra [1991] introduced a deterministic algorithm to construct a normal element which uses $n^{O(1)}$ operations in $\mathbb{F}_{q^n}/\mathbb{F}_q$. To the best of our knowledge, the algorithm of Augot and Camion [1994] is the most efficient deterministic method, with a cost of $O(n^3 + n^2 \log q)$ operations in $\mathbb{F}_q$.

In characteristic zero, Schlickewei and Stepanov [1993] 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 [1994] gives an algorithm for the more general case of finding a normal basis in an abelian extension $\mathsf{K}/\mathsf{F}$ which requires $n^{O(1)}$ in $\mathsf{F}$. More generally in characteristic zero, for any Galois extension $\mathsf{K}/\mathsf{F}$ of degree $n$ with Galois group given by a collection of $n$ matrices, Girstmair [1999] gives an algorithm which requires $O(n^4)$ operations in $\mathsf{F}$ to construct a normal element in $\mathsf{K}$.

In this paper we present a new randomized algorithm for finding a normal element for abelian and metacyclic extensions, with a runtime quadratic in the degree $n$ of the extension. The costs of all algorithms are measured by counting *arithmetic operations* in $\mathsf{F}$ at unit cost. Questions related to the bit-complexity of our algorithms are challenging, and beyond the scope of this paper.

Our main conventions are the following.

Assumption 1. *Let $\mathsf{K}/\mathsf{F}$ 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 $\mathsf{F}$ of characteristic zero. Then,*

- *elements of $\mathsf{K}$ 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 $\xi$.*

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

Our algorithms combine techniques and ideas due to [von zur Gathen and Giesbrecht 1990; Kaltofen and Shoup 1998]: $\alpha \in \mathsf{K}$ 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 $\alpha$ at random; a generic choice is normal (so we expect $O(1)$ random trials to be sufficient). However, writing down $S_\alpha$ involves $\Theta(n^2)$ elements in $\mathsf{F}$, which precludes a subquadratic runtime. Instead, knowing $\alpha$, 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 $S_\alpha$, giving us an element $s_{\alpha,\ell} \in \mathsf{F}[G]$. For that, we adapt algorithms from [Kaltofen and Shoup 1998] that were written for Galois groups of finite fields.

Having $s_{\alpha,\ell}$ in hand, we need to test its invertibility. In order to do so, we present an algorithm in the abelian case which relies on the fact that $\mathsf{F}[G]$ is 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 that of Fourier transform over $G$, and it is worth mentioning that there is a vast literature on fast algorithms for Fourier transforms (over the base field $\mathbb{C}$). Relevant to our current context, we invite the reader to consult [Clausen and Müller 2004; Maslen et al. 2018] and references therein for details. At this stage, it is not clear how we can apply these methods in our context.

This paper is written from the point of view of obtaining improved asymptotic complexity estimates. 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 $\omega(4/3)$ that describes the cost of certain rectangular matrix products (see the end of this section).

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, with* $(3/4)\cdot\omega(4/3) < 1.99$. *Moreover, the same problem for metacyclic groups can be solved using* $\tilde{O}(|G|^2)$ *operations in* F. *The algorithms are randomized.*

Although the cost is quadratic in the size of input for a general metacyclic group, in many cases it will be (slightly) subquadratic, under specific conditions on the 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 the cost can be improved.

Our algorithms make extensive of known algorithms for polynomial and matrix arithmetic; in particular, we use repeatedly the fact that polynomials of degree $d$ in $\mathsf{F}[x]$, for any field F of characteristic zero, can be multiplied in $\tilde{O}(n)$ operations in F [Schönhage and Strassen 1971]. As a result, arithmetic operations $(+,\times,\div)$ in K can all be done using $\tilde{O}(n)$ operations in F [von zur Gathen and Gerhard 2013].

For matrix arithmetic, we will rely on some non-trivial results on rectangular matrix multiplication initiated by Lotti and Romani [1983]. For $k \in \mathbb{R}$, we denote by $\omega(k)$ a constant such that over any ring, matrices of sizes $(n,n)$ by $(n,\lceil n^k\rceil)$ can be multiplied in $O(n^{\omega(k)})$ ring operations (so $\omega(1)$ is the usual exponent of square matrix multiplication, which we simply write $\omega$). The sharpest values known to date for most rectangular formats are from [Le Gall and Urrutia 2018]; for $k=1$, the best known value is $\omega \le 2.373$ by Le Gall [2014]. Over a field, we will frequently use the fact that further matrix operations (determinant or inverse) can be done in $O(n^\omega)$ base field operations.

## 2 PRELIMINARIES

One of the well-known proofs of the existence of a normal element for a finite Galois extension [Lang 2002, Theorem 6.13.1] suggests a randomized algorithm for finding such an element. Assume K/F is a finite Galois extension with Galois group $G = \{g_1,\dots,g_n\}$. 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 each $i \in \{1,\dots,n\}$, applying $g_i$ to equation (2.1) yields

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

Using (2.1) and (2.2), one can form the linear system $\mathbf{M}_G(\alpha)\boldsymbol{c} = \mathbf{0}$, with $\boldsymbol{c} = [c_1 \ \cdots \ c_n]^T$ and where, for $\alpha \in \mathsf{K}$,

$$\mathbf{M}_G(\alpha) = \begin{bmatrix} g_1g_1(\alpha) & g_1g_2(\alpha) & \cdots & g_1g_n(\alpha) \\ g_2g_1(\alpha) & g_2g_2(\alpha) & \cdots & g_2g_n(\alpha) \\ \vdots & \vdots & \vdots & \vdots \\ g_ng_1(\alpha) & g_ng_2(\alpha) & \cdots & g_ng_n(\alpha) \end{bmatrix} \in M_n(\mathsf{K}).$$

Classical proofs then proceed to show that there exists $\alpha \in \mathsf{K}$ with $\det(\mathbf{M}_G(\alpha)) \neq 0$.

This approach can be used as the basis of a randomized algorithm for finding a normal element: choose a random element $\alpha$ 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 [1999]. The main contribution of this paper is to show how to speed up this verification.

Before entering that discussion, we briefly discuss the probability that $\alpha$ be a normal element: 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 $(a_0,\dots,a_{n-1})$. 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 $\alpha$ be normal is at least $1 - n/|X|$.

If $G$ is cyclic, [von zur Gathen and Giesbrecht 1990] avoid computing a determinant by computing the GCD of $S_\alpha := \sum_{i=0}^{n-1} g_i(\alpha)x^i$ and $x^n - 1$. In effect, this amounts to testing whether $S_\alpha$ is invertible in the group ring $\mathsf{K}[G]$, which is isomorphic to $\mathsf{K}[x]/\langle x^n-1\rangle$. 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 $\alpha$ being normal is equivalent to $S_\alpha$ 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 $S_\alpha$ itself still involves $\Theta(n^2)$ elements in F. The following lemma is the main new ingredient in our algorithm: it gives a randomized reduction to testing whether a suitable projection of $S_\alpha$ 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 $\mathsf{F}$-linear projection $\ell : \mathsf{K} \to \mathsf{F}$.*

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

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

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 $\ell_k$'s by indeterminates $L_k$'s, the determinant becomes a polynomial in $P \in \mathsf{F}[L_1, \ldots, L_n]$. Viewing $P$ in $\mathsf{K}[L_1, \ldots, L_n]$, we have $P(1, \xi, \ldots, \xi^{n-1}) = \det(\mathbf{M}_G(\alpha))$, which is non-zero by assumption. Hence, $P$ is not identically zero, and the conclusion follows.

($\Leftarrow$) Assume $\mathbf{M}_G(\alpha)$ is not invertible. Following the proof of [Jamshidpey et al. 2018, 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 $i \in \{1, \ldots, n\}$ 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 $j \in \{1, \ldots, n\}$. 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 $g(\boldsymbol{u}) = \boldsymbol{u}$, and hence $\boldsymbol{u} \in \mathsf{F}^n$.

Now we show that $\ell(\mathbf{M}_G(\alpha))$ is not invertible for all choices of $\ell$. By Equation (2.3), we can write

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

Since $\boldsymbol{u}$ has entries in $\mathsf{F}$, $\mathbf{M}_G(\alpha)\boldsymbol{u} = 0$ yields $\mathbf{M}^{(j)}\boldsymbol{u} = 0$ for $j \in \{1, \ldots, n\}$. Hence,

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

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

Our algorithm can be sketched as follows: choose random $\alpha$ in $\mathsf{K}$ 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 PROJECTIONS OF THE ORBIT SUM

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. We will see that they follow closely ideas used in [Kaltofen and Shoup 1998] over finite fields.

Suppose $G = \langle g \rangle$, so that given $\alpha$ in $\mathsf{K}$ and $\ell : \mathsf{K} \to \mathsf{F}$, our goal is to compute

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

Kaltofen and Shoup [1998] 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 original algorithm of Kaltofen and Shoup [1998] relies on the properties of the Frobenius mapping to achieve subquadratic runtime. In our case, we cannot apply these results directly; instead, we have to revisit the proofs of Kaltofen and Shoup [1998], Lemmata 3, 4 and 8, now considering rectangular matrix multiplication. Our exponents involve the constant $\omega(4/3)$, for which we have the upper bound $\omega(4/3) < 2.654$: this follows from the upper bounds on $\omega(1.3)$ and $\omega(1.4)$ given by Le Gall and Urrutia [2018], and the fact that $k \mapsto \omega(k)$ is convex [Lotti and Romani 1983]. In particular, $3/4 \cdot \omega(4/3) < 1.99$. Note also the inequality $\omega(k) \geq 1 + k$ for $k \geq 1$, since $\omega(k)$ describes products with input and output size $O(n^{1+k})$.

## 3.1 Multiple automorphism evaluation and applications

The key to the algorithms below is the remark following Assumption 1, which reduces automorphism evaluation to modular composition of polynomials. Over finite fields, this idea goes back to von zur Gathen and Shoup [1992], 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 $\xi$) as $\gamma(\gamma)$; this can be done with $\tilde{O}(n^{(\omega+1)/2})$ operations in $\mathsf{F}$ using Brent and Kung's modular composition algorithm [Brent and Kung 1978]. The algorithms below describe similar operations along these lines, involving several simultaneous evaluations.

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

Proof. (Compare [Kaltofen and Shoup 1998, 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 $\alpha_1, \ldots, \alpha_s$ 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 $i = 0, \ldots, t$. There are $t$ products in $\mathsf{K}$ to perform, so this amounts to $\tilde{O}(n^{7/4})$ operations in $\mathsf{F}$.

Having $\gamma_i$'s in hand, one can form the matrix $\mathbf{\Gamma} := [\Gamma_0 \ \cdots \ \Gamma_{t-1}]^T$, where each column $\Gamma_i$ is the coefficient vector of $\gamma_i$ (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 $\mathbf{B} := \mathbf{A}\,\mathbf{\Gamma}$; as per our 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 to get all $\alpha_i(\gamma)$ is to write them as $\alpha_i(\gamma) = \sum_{0\le j<m} a_{i,j}(\gamma)\gamma_t^j$. Using Horner's scheme, this takes $O(sm)$ operations in K, which is $\tilde{O}(n^{7/4})$ operations in F. Since we pointed out that $\omega(3/4) \ge 7/4$, the leading exponent in all costs seen so far is $(3/4)\cdot\omega(4/3)$. □

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

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

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

Proof. (Compare [Kaltofen and Shoup 1998, Lemma 4].) For a given $m \in \{1, \ldots, r\}$, suppose we have computed

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

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

Then, we can obtain $G_{i_1,\ldots,i_m}$ for $0 \le i_j \le s_j$ if $1 \le j < m$, and $0 \le i_m < 2k_m$, by computing $\eta(G_{i_1,\ldots,i_m})$, for all indices $i_1, \ldots, i_m$ available to us, that is, $0 \le i_j \le s_j$ if $1 \le j < m$, and $0 \le i_m < k_m$. 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 also compute $\eta^2$ by means of one modular composition, whose cost is negligible.

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 $m = 1, \ldots, r$; 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 (our reference [Kaltofen and Shoup 1998] also had such a discussion). Seen as an F-linear map, the operator $g : \alpha \mapsto g(\alpha)$ admits a transpose, which maps 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* [Canny et al. 1989; Kaminski et al. 1988] 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 $T + O(N + M)$. In particular, given $s$ linear forms $\ell_1, \ldots, \ell_s$ and $g$ in $G$, transposing Lemma 3.1 shows that we can compute $\ell_1 \circ g, \ldots, \ell_s \circ g$ in time $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$. The following lemma sketches the construction.

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

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

Let $t, m$ and $\gamma_0, \ldots, \gamma_t$ be as in the proof of Lemma 3.1. Next, compute the "giant steps" $\gamma_t^j = \gamma^{tj}$, $j = 0, \ldots, m-1$ and for $i = 1, \ldots, s$ and $j = 0, \ldots, m-1$, deduce the linear forms $L_{i,j}$ defined by $L_{i,j}(\alpha) := \ell_i(\gamma^{tj}\alpha)$ for all $\alpha$ in K. Each of them can be obtained by a *transposed multiplication* in time $\tilde{O}(n)$ [Shoup 1995, Section 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 $\gamma_0, \ldots, \gamma_{t-1}$ (as columns) to obtain all values $\ell_i(\gamma^j)$, for $i = 1, \ldots, s$ an $j = 0, \ldots, n-1$. This can be accomplished with $O(n^{(3/4)\cdot\omega(4/3)})$ operations in F. □

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

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

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

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

Proof. (Compare [Kaltofen and Shoup 1998, Lemma 8].) We proceed as in Lemma 3.2. For $m = 1, \ldots, r$, assume we know $L_{i_1,\ldots,i_m} := \ell \circ (g_1^{i_1} \cdots g_m^{i_m})$, for $0 \le i_j \le s_j$ if $1 \le j < m$, and $0 \le i_m < k_m$. Using the previous lemma, we compute all $L_{i_1,\ldots,i_m} \circ {g_m}^{k_m}$, which gives us $L_{i_1,\ldots,i_m}$ for indices $0 \le i_m < 2k_m$. The cost 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

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

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 $\sum_{i_1,\ldots,i_r} c_{i_1,\ldots,i_r} {g_1}^{e_1} \cdots {g_r}^{e_r}$, with $0 \le i_j < e_j$ for all $j$.

Proposition 3.5. *Suppose that $G$ is abelian, with notation as above. For $\alpha$ 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

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

where $\ell$ 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 $L_{j_1,\ldots,j_r}(G_{i_1,\ldots,i_r})$, for $1 \le m \le r, 0 \le i_m < s_m, 0 \le j_m < s_m$. Here, $L_{j_1,\ldots,j_r} := \ell \circ (g_1^{s_1 j_1} \cdots g_r^{s_r j_s})$ are linear projections presented as row vectors and $G_{i_1,\ldots,i_r} := g_1^{i_1} \cdots g_r^{i_r}(\alpha)$ 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

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

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

**Step 2.** Compute all $g_i^{s_i}$, $i = 1, \ldots, r$; this involves $O(\log(n))$ modular compositions, so the cost is negligible compared to that of Step 1.

**Step 3.** Use Lemma 3.4 to compute

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

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

**Step 4.** Multiply the matrix with rows the coefficients of all $L_{j_1,\ldots,j_r}$ by the matrix with columns the coefficients of all $G_{i_1,\ldots,i_r}$; this yields all required values. We compute this product in $O(n^{(1/2)\cdot\omega(2)})$ operations in $\mathsf{F}$, 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 [Johnson 1976, p. 88] or [Curtis and Reiner 1988, 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

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

or

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

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

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

Conversion between the two representations involves no operation in $\mathsf{F}$, 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, with notation as above. For $\alpha$ in $\mathsf{K}$ and $\ell : \mathsf{K} \to \mathsf{F}$, $s_{\alpha,\ell} \in \mathsf{F}[G]$ is computable using $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in $\mathsf{F}$.*

Proof. Suppose first that $s \le m$; then, we use the presentation (3.4) of the elements of $G$. Take $\alpha$ in $\mathsf{K}$ 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 are under the assumptions of the lemma. This takes $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in $\mathsf{F}$.

**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 $\mathsf{F}$.

**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 $\mathsf{F}$, 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 THE GROUP ALGEBRA

In this section we consider the problem of invertibility testing in $\mathsf{F}[G]$, specifically for abelian and metacyclic groups $G$: given an element $\beta$ in $\mathsf{F}[G]$, for a field $\mathsf{F}$ and a group $G$, determine whether $\beta$ is a unit in $\mathsf{F}[G]$. As well as being necessary in our application to normal bases, we believe these problems are of independent interest.

Since we are in characteristic zero, Wedderburn's theorem implies the existence of an $\mathsf{F}$-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 $\mathsf{F}$. 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 $\beta_1 \in M_{d_1}(\mathbb{C}), \ldots, \beta_r \in M_{d_r}(\mathbb{C})$ 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 $\mathsf{F}$-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 $\beta$ requires we work in a degree $n/2$ extension of $\mathbb{Q}$, implying a quadratic runtime.

In this section, we give algorithms for the problem of invertibility testing for the families of group seen so far, abelian and metacyclic. For the former, we actually 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 $\mathsf{F}$-algebras of the form $\mathsf{F}[z]/\langle P_i(z)\rangle$, for certain polynomials $P_i \in \mathsf{F}[z]$ (recovering $\beta$ from its image is softly-linear time as well). Not only does this allow us to test whether $\beta$ is invertible, this would also make it

possible to find its inverse in $\mathsf{F}[G]$ (or to compute products in $\mathsf{F}[G]$) in softly-linear time. We are not aware of previous results of this kind. For metacyclic groups, we describe an injective $\mathsf{F}$-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, the group algebra $\mathsf{F}[G]$ of such a group is the tensor product of cyclic algebras. Using this property, given an element $\beta$ in $\mathsf{F}[G]$, our goal in this section is to determine whether $\beta$ is a unit.

The previous property implies that $\mathsf{F}[G]$ 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 $\mathsf{F}$-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 $\varepsilon > 0$ [Li et al. 2009, Theorem 2]. From this it may be possible to deduce similar upper bounds on the complexity of invertibility tests, following [Dahan et al. 2006], 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 [Poli 1994] 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

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

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

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

Proof. Without loss of generality, we prove the first claim over $\mathbb{Q}$; the result over $\mathsf{F}$ 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 $\gamma$. Start from a Bézout relation $am + a'm' = 1$, for some $a, a'$ in $\mathbb{Z}$. Since $x^m = x'^{m'} = 1$ in $\mathbb{h}$, 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 $\mathbb{h}$, 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 [Bostan et al. 2006] in quasi-linear time). By [Bach and Shallit 1996, 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 $\gamma$.

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 $\mathsf{F}$. 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 exist an $\mathsf{F}$-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 $\Gamma$ and its inverse to any input using $\tilde{O}(\dim(\mathbb{H}))$ operations in $\mathsf{F}$.*

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, $\Gamma_1$ and its inverse can be applied to any input in time $\tilde{O}(\varphi(\mu))$. Iterate this process another $t-2$ times, to obtain $\Gamma$ 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 $p$.** In the following two paragraphs, we discuss the opposite situation as above: we now work with 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 $\mathsf{F}$. Define

$$\mathbb{a} := \mathsf{F}[x]/\Phi_{p^c}(x), \tag{4.1}$$

$$\mathbb{b} := \mathsf{F}[x,y]/\langle\Phi_{p^c}(x),\Phi_{p^{c'}}(y)\rangle = \mathbb{a}[y]/\Phi_{p^{c'}}(y). \tag{4.2}$$

Note that $\mathbb{a}$ and $\mathbb{b}$ have respective dimensions $\varphi(p^c)$ and $\varphi(p^c)\varphi(p^{c'})$.

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

Proof. Let $\xi$ be the residue class of $x$ in $\mathbb{A}$. Then, in $\mathbb{a}[y]$, $\Phi_{p^{c'}}(y)$ 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 $\mathbb{a}$ may not be a field, the Chinese Remainder theorem implies that $\mathbb{b}$ is isomorphic to $\mathbb{a}^{\varphi(p^{c'})}$; the isomorphism is given by

$$\begin{array}{rccc} \theta: & \mathbb{b} & \to & \mathbb{a}\times\cdots\times\mathbb{a}, \\ & P & \mapsto & (P(\xi,\rho_1),\ldots,P(\xi,\rho_{\varphi(p^{c'})}). \end{array}$$

In terms of complexity, arithmetic operations $(+,-,\times)$ in $\mathbb{a}$ can all be done in $\tilde{O}(\varphi(p^c))$ operations in F. Starting from $\rho_1 \in \mathbb{a}$, all other roots $\rho_i$ can then be computed in $O(\varphi(p^{c'}))$ operations in $\mathbb{a}$, that is, $\tilde{O}(\dim(\mathbb{b}))$ operations in F.

Applying $\theta$ and its inverse is done by means of fast evaluation and interpolation [von zur Gathen and Gerhard 2013, Chapter 10] in $\tilde{O}(\varphi(p^{c'}))$ operations in $\mathbb{a}$, that is, $\tilde{O}(\deg(\mathbb{b}))$ operations in F (the algorithms do not require that $\mathbb{a}$ be a field). □

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

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

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 $\mathbb{a} := \mathsf{F}[x_1]/\Phi_{p^{c_1}}(x_1)$.

Lemma 4.4. *There exists an* F*-algebra isomorphism* $\Theta : \mathbb{A} \to \mathbb{a}^{\dim(\mathbb{A})/\dim(\mathbb{a})}$. *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

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

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

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

$\Theta_1$ and its inverse can be evaluated in quasi-linear time $\tilde{O}(\dim(\mathbb{A}))$. We now work in all copies of $\mathbb{a}[x_3,\ldots,x_t]/\langle\Phi_{p^{c_3}}(x_3),\ldots,\Phi_{p^{c_t}}(x_t)\rangle$ independently, and apply the procedure above to each of them. Altogether we have $t-1$ such steps to perform, giving us an isomorphism

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

The exponent can be rewritten as $\dim(\mathbb{A})/\dim(\mathbb{a})$, as claimed. In terms of complexity, all $\Theta_i$'s and their inverses can be computed in quasi-linear 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 $\boldsymbol{x} = (x_1,\ldots,x_t)$ are as before; we now consider positive integers $\boldsymbol{b} = (b_1,\ldots,b_t)$, and the F-algebra

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

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 $\boldsymbol{c} = (c_1,\ldots,c_N)$ 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 they are pairwise coprime, so that we have a Chinese Remainder isomorphism

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

Together with its inverse, this can be computed in $\tilde{O}(p^{b_i})$ operations in F [von zur Gathen and Gerhard 2013, Chapter 10]. By distributivity of the tensor product over direct products, 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 $\boldsymbol{c} = (c_1,\ldots,c_t)$. Together with its inverse, $\lambda$ 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

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

where $\mathbb{a}_{\boldsymbol{c}} = \mathsf{F}[z]/\langle\Phi_{p^c}(z)\rangle$, with $c = \max(c_1,\ldots,c_t)$ and $D_{\boldsymbol{c}} = \dim(\mathbb{A}_{t,\boldsymbol{c},\boldsymbol{x}})/\dim(\mathbb{a}_{\boldsymbol{c}})$. As before, $\Lambda$ 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 $p_1,\ldots,p_s$, 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 $b_{i,1}\leq\cdots\leq b_{i,t_i}$; this is the invariant factor decomposition of $G_i$, with $b_{i,1}+\cdots+b_{i,t_i} = a_i$.

We henceforth assume that generators $\gamma_{1,1},\ldots,\gamma_{s,t_s}$ of respectively $G_{1,1},\ldots,G_{s,t_s}$ are known, and that elements of $\mathsf{F}[G]$ are given on the power basis in $\gamma_{1,1},\ldots,\gamma_{s,t_s}$. Were this not the case, given arbitrary generators $g_1,\ldots,g_r$ of $G$, with orders $e_1,\ldots,e_r$, a brute-force solution would factor each $e_i$ (factoring $e_i$ takes $o(e_i)$ bit operations on a standard RAM), so as to write $\langle g_i\rangle$ as a product of cyclic groups of prime power orders, from which the required decomposition follows.

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

The proof occupies the rest of this paragraph. 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 $\mathsf{F}$-algebra, to

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

with $\boldsymbol{b}_i = (b_{i,1}, \ldots, b_{i,t_i})$ and $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,t_i})$. Given $\beta$ on the power basis in $\gamma_{1,1}, \ldots, \gamma_{s,t_s}$, 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 $c_{i,1}, \ldots, c_{i,N_i}$ 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

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

for all indices $\boldsymbol{j} = (j_1, \ldots, j_s)$, with $j_1 = 1, \ldots, N_1, \ldots, j_s = 1, \ldots, N_s$; call $\Gamma$ 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 softly linear 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 $\mathbb{H}_{\boldsymbol{p}, \boldsymbol{c}_j, \boldsymbol{z}}$, with

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

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

$$\Gamma' : \prod_j \mathbb{H}_{\boldsymbol{p}, \boldsymbol{c}_j, \boldsymbol{z}} \to \prod_j \mathsf{F}[z] / \langle \Phi_{d_j}(z) \rangle,$$

for certain integers $d_j$. 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 $\mathsf{F}[z] / \langle \Phi_{d_j}(z) \rangle$. Invertibility in such an algebra can be tested in softly linear time by applying the fast extended GCD algorithm [von zur Gathen and Gerhard 2013, Chapter 11], so our conclusion follows.

Together with Proposition 3.5, the above result proves the first part of Theorem 1.1.

## 4.2 Metacyclic Groups

In this last section, we study the invertibility problem for a metacyclic group $G$. Instead of using an $\mathsf{F}$-algebra isomorphism, as we did above, we will 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 [Curtis and Reiner 1988, §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 $\zeta$ be the image of $z$ in $\mathbb{A}$.

Lemma 4.7. *The mapping*

$$\begin{array}{rccc} \psi : & \mathsf{F}[G] & \to & M_s(\mathbb{A}) \\ & \sigma & \mapsto & \mathrm{Diag}(\zeta, \zeta^r, \ldots, \zeta^{r^{s-1}}) \\ & \tau & \mapsto & \left[\begin{array}{c|c} 0 & \zeta \\ \hline \mathbf{I}_{s-1} & 0 \end{array}\right], \end{array}$$

*an injective homomorphism of* $\mathsf{F}$*-algebras.*

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 $\psi$ is a well-defined $\mathsf{F}$-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 $j = 0, \ldots, s-1$, 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 $\beta$ is in $\mathrm{Ker}(\psi)$, we get $F_i(\zeta) = 0$, that is, $F_i \bmod (z^m - 1) = 0$, for $0 \leq i < s$. Since all $F_i$'s have degree less than $m$, they are all zero. □

We conclude this section with two algorithms that test whether $\psi(\beta) \in M_s(\mathbb{A})$ is invertible, for a given $\beta$ 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 available, one would simply use the Chinese Remainder theorem and work in field extensions of $\mathsf{F}$).

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

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 $\mathsf{F}$. 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 [Curtis and Reiner 1988, 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 $\beta$ 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 $\psi$ implies that this is the case if and only if $\beta$ 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 $\mathsf{F}$, using the determinant algorithm of [Kaltofen and Villard 2004, Section 6]. □

Before giving our second algorithm, let us point out that matrix-vector products by $\psi(\beta)$ can be done fast.

Lemma 4.9. *Given $\beta$ 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 $\mathsf{F}$.*

Proof. We use the basis of $\mathsf{F}[G]$ given in (3.5), writing $\beta = \sum_{i=0}^{m-1} \left( \sum_{j=0}^{s-1} b_{i,j} \tau^i \right) \sigma^j \in \mathsf{F}[G]$. We rewrite this as $\beta = \sum_{i=0}^{m-1} B_i(\tau)\sigma^i$, 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 $\{0,\dots,m-1\}$. The vector $\psi(\sigma)^i\boldsymbol{v}$ can be obtained by multiplying each entry of $\boldsymbol{v}$ by a power of $\zeta$; this takes $\tilde{O}(sm)$ operations in $\mathsf{F}$. 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 softly linear time $\tilde{O}(sm)$.

Adding a factor of $m$ to account for indices $i=0,\dots,m-1$, we get the result. □

Second algorithm for Corollary 4.8. The second algorithm uses Wiedemann's [1986] algorithm, and its extension by Kaltofen and Saunders [1991]. Extra care will be needed to accommodate the fact that $\mathbb{A}$ has zero-divisors. Let $F_1,\dots,F_s$ 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 $i=1,\dots,s$. We write $\pi_i : \mathbb{A}\to\mathbb{A}_i$ for the canonical projection, and extend the notation to matrices over $\mathbb{A}$.

For $\beta$ 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 [Kaltofen and Saunders 1991, 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 $r_i := \operatorname{rank}(\mathbf{M}_i'')$, $i=1,\dots,s$. 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 [Kaltofen and Saunders 1991, 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 $(\boldsymbol{u}_i^T(\mathbf{M}_i''\mathbf{X}_i)^j\boldsymbol{v}_i)_{j\ge0}$ 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 $(\gamma_j)_{j\ge0}$, 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 $\mathsf{F}$. Hence, all required terms can be obtained in $\tilde{O}(s^2m^2)=\tilde{O}(n^2)$ operations in $\mathsf{F}$.

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 $r_1,\dots,r_s$. Since $\mathbb{A}$ is not a field, we rely on the algorithm of [Accettella et al. 2003; Dahan et al. 2006]. Using $\tilde{O}(sm)$ operations in $\mathsf{F}$, it reveals a partial factorization of $z^m-1$ as $G_1\cdots G_t$ (the factors may not be irreducible) and integers $\rho_j$, $j=1,\dots,t$, such that for all $i\le s$, $j\le t$, 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. □

## 5 BASIS CONVERSION

**Normal to Power basis** Suppose $G=\{g_1,\dots,g_n\}$, $\alpha$ is a normal element of $K$ and $u = \begin{bmatrix}u_1 & \dots & u_n\end{bmatrix} \in F^n$ is given as the coefficient vector of $u$ in the normal basis. In order to write $u$ in the power basis one can compute

$$\begin{bmatrix} \overline{g_1(\alpha)} \mid \cdots \mid \overline{g_n(\alpha)} \end{bmatrix} \cdot \begin{bmatrix}u_1 & \cdots & u_n\end{bmatrix}^t, \tag{5.1}$$

where $\overline{g_i(\alpha)}$ is the coefficient vector of $g_i(\alpha)$. The result of (5.1) is the coefficient vector of

$$\sum_{i=1}^{n} u_i g_i(\alpha). \tag{5.2}$$

Note that (5.1) shows that conversion from normal to power basis is transpose of the problem of computing the projections of conjugates of $\alpha$ which is solved in Section 3. Here we present an explicit algorithm to solve this problem i.e. to compute (5.2) for abelian and metacyclic $G$. Before stating the algorithm for basis conversion, we present the last ingredient needed.

Lemma 5.1. *Under the Assumption 1, suppose $g_1,\dots g_r \in G$ and $h_{i_1\cdots i_r}\in K$ for $0\le j\le r$, $1\le i_j\le s_j$, $\prod_{j=1}^r s_j \le \lceil\sqrt{|G|}\rceil$ are given. the elements*

$$g_r^{i_r}\cdots g_1^{i_1}(h_{i_1\cdots i_r})\, 0\le j\le r,\ 1\le i_j\le s_j,\ \prod_{j=1}^{r} s_j \le \lceil\sqrt{|G|}\rceil$$

*can be computed using $\tilde{O}(|G|^{(3/4)\cdot\omega(4/3)})$ operations over $F$.*

Proof. Let $h_{i_1\cdots i_r}^{(0\cdots0)} = h_{i_1\cdots i_r}$ and

$$g_r^{l_r}\cdots g_1^{l_1}(h_{i_1\cdots i_r}^{(u_1\cdots u_r)}) = h_{i_1\cdots i_r}^{(u_1+l_1\cdots u_r+l_r)}$$

for $0\le j\le r$ and $0\le l_j$. Assume $0\le t<r$ is fixed and

$$h_{i_1\cdots i_r}^{(i_1\cdots i_{t-1}0\cdots0)},\ 0\le j<t,\ 1\le i_j\le s_j,$$

is already computed. We compute

$$h_{i_1\cdots i_r}^{(i_1\cdots i_t0\cdots0)},\ 0\le j\le t,\ 1\le i_j\le s_j,$$

by iteratively applying the following method. In $m$-th iteration we apply $g_t^{2^{m-1}}$ to $h_{i_1\cdots i_r}^{(i_1\cdots i_t0\cdots0)}$ for $j\le t-1$, $0\le i_j\le s_j$ and

$$i_t \in \left(\cup_{k=1}^{\infty}\{(2k-1)2^{m-1},\dots,(2k)2^{m-1}-1\}\right)\cap\{0,\dots,s_t\}$$

At each iteration we have to compute $O(|\lceil\sqrt{G}\rceil|)$ modular composition which can be done by applying Lemma 3.1 and $g^{2^i}$ can be computed by applying 3.2. This gives the claimed complexity. □

**Abelian extension.** Under Assumption 1, suppose $G$ is given by (3.2), $\alpha$ is a normal element and $u = \begin{bmatrix}u_1\cdots u_n\end{bmatrix}^t \in F^n$ is an element of $K$. Moreover, let $\prod s_i = O(\lceil\sqrt{n}\rceil)$. The basis conversion from normal to power basis is equivalent of computing (5.2) and the later problem can be solved in three steps. Before presenting the algorithm, note that (5.2) can be written as

$$\sum_{\substack{1\le j\le r\\ 0\le i_j<e_j}} u_{i_1\cdots i_r} g_1^{i_1}\cdots g_r^{i_r}(\alpha) = \sum_{\substack{1\le j\le r\\ 0\le i_j\le s_j}} \overline{g_1}^{i_1}\cdots\overline{g_r}^{i_r}(h_{i_1\cdots i_r}),$$

where

$$h_{i_1\cdots i_r} = \sum_{\substack{1\le k\le r\\ 0\le j_k<s_k}} u_{j_1\cdots j_r}^{(i_1\cdots i_r)} g_1^{j_1}\cdots g_r^{j_r}(\alpha)$$

and $\overline{g_j} = g_j^{s_j}$.

**Step 1.** We begin the process by computing

$$g_1^{i_1}\cdots g_r^{i_r}(\alpha),\ 1\le j\le r, 0\le i_j\le s_i.$$

Using Lemma 3.2, this can be done using $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in $F$.

**Step 2.** In order to get $h_{i_1\cdots i_r}$ multiply the matrix with rows $g_1^{i_1}\cdots g_r^{i_r}(\alpha)$ by the matrix with columns the coefficients of $h_{i_1\cdots i_r}$. This operation can be done in $O(n^{1/2\cdot\omega(2)})$.

**Step 3.** Use Lemma 5.1 to compute (5.2).

**Metacyclic extension.** with the same assumptions as in the abelian case, suppose $G$ is given by (3.3). In this case (5.2) can be rewritten as

$$\sum_{i=0}^{m}\sum_{0}^{s} u_{ij}\sigma^i\tau^j(\alpha) = \sum_{i=0}^{\lceil\sqrt{m}\rceil}\overline{\sigma}^i(h_i),$$

where $h_i = \sum_{j=0}^{\sqrt{m}/\sqrt{s}}\sum_{k=0}^{s} a_{jk}^{(i)}\sigma^j\tau^k(\alpha)$ and $\overline{\sigma} = \sigma^{\lceil\sqrt{ms}\rceil}$. Note that without loss of generality we can assume $s \le m$. Similar to the abelian case, we can do the computation in three steps.

**Step 1.** Compute $\sigma^i\tau^j(\alpha)$ for $0 \le i \le \lceil\sqrt{m}/\sqrt{s}\rceil$ and $0 \le j < s$. Using Lemma 3.2, this can be done using $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in $F$.

**Step 2.** In order to get $h_i$ multiply the matrix with rows $\sigma^i\tau^j(\alpha)$ by the matrix with columns the coefficients of $h_i$. This operation can be done in $O((ms)^{1/2\cdot\omega(2)})$.

**Step 3.** Use Lemma 5.1 to compute (5.2).

**Power Basis to normal basis.**

Now assume $u \in K$ is given in power basis. The goal is to find $c_i$'s in $F$ such that,

$$u = \sum_{i=0}^{n-1} c_i g_i(\alpha).$$

Then for any element $g_j$ of $G$ we have

$$g_j(u) = \sum_{i=0}^{n-1} c_i g_j g_i(\alpha).$$

On the other hand if $l$ is a random projection of $K$ to $F$, we get

$$l(g_j(u)) = \sum_{i=0}^{n-1} c_i l(g_j g_i(\alpha)).$$

Hence, in order to find $c_i$'s it is enough to solve the linear system

$$l(M_G(\alpha))\mathbf{x} = \left[l(g_i(u))\right]_{0\le i<n} \tag{5.3}$$

Hence the conversion problem can be solved in 2 steps

**Step 1.** compute $\left[l(g_i(u))\right]_{0\le i<n}$ which can be done using the algorithms of Section 3. This can be carried out using $\tilde{O}(|G|^{(3/4)\cdot\omega(4/3)})$ operations in $F$.

**Step 2.** compute $l(M_G(\alpha))$ and then solving the linear system (5.3).

**Abelian extension.** It is already mentioned that multiplying by $l(M_G(\alpha))$ is equivalent of multiplication in a multivariate ring, by $s_{\alpha,\ell}$. Hence by considering the group algebra version of (5.3) we have to solve

$$s_{\alpha,\ell}x = s_u$$

for $x$ in $F[G]$. This can be done by using the tools provided in Section ?. Computing the inverse of $s_{\alpha,\ell}$ costs the same as testing its invertiblity. Having the inverse we get the solution with a single multiplication in $F[G]$.

**Metacyclic extension.**

In order to solve (5.3), in the metacyclic case, we start with understanding the structure of $M_G(\alpha)$. Assume the first row of $M_G(\alpha)$ is

$$\left[\sigma^0\tau^0 \quad \cdots \quad \sigma^{n-1}\tau^0 \quad \sigma^0\tau \quad \cdots \quad \sigma^{n-1}\tau \quad \sigma^0\tau^s \quad \cdots \quad \sigma^{n-1}\tau^s\right]$$

Now by considering the actions (respectively) of

$$\tau^0\sigma^1,\ldots,\tau^0\sigma^{n-1},\tau^1\sigma^0,\ldots,\tau^1\sigma^{n-1},\cdots\tau^s\sigma^0,\ldots,\tau^s\sigma^{n-1}$$

on the first row we get $M_G(\alpha)$. Note that here we are using both presentation of elements in $G$ as $\sigma^i\tau^j$ and $\tau^j\sigma^i$. Using above argument it is not hard to see $M_G(\alpha)$ as a block matrix with blocks generated by the action of $\tau^l\sigma^1,\ldots,\tau^l\sigma^{n-1}$ on $\sigma^0\tau^k,\ldots,\sigma^{n-1}\tau^k$. In other word the $lk$-th block of $M$ is $\left(\tau^l\sigma^{(i-1)+(j-1)}\tau^k\right)_{ij}$ for $1 \le i,j \le n$. This shows that each block is a Hankel $n\times n$ matrix.

## REFERENCES


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.

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.). Abbaye de la Bussière sur Ouche, France.

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

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

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.

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

M. Clausen and M. Müller. 2004. Generating fast Fourier transforms of solvable groups. *J. Symbolic Comput.* 37, 2 (2004), 137–156. https://doi.org/10.1016/j.jsc.2002.06.006

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. Reprint of the 1962 original, A Wiley-Interscience Publication.

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.

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.

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

J. von zur Gathen and M. Giesbrecht. 1990. Constructing normal bases in finite fields. *J. Symbolic Comput.* 10, 6 (1990), 547–570. https://doi.org/10.1016/S0747-7171(08)80158-7

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

K. Girstmair. 1999. An algorithm for the construction of a normal basis. *J. Number Theory* 78, 1 (1999), 36–45. https://doi.org/10.1006/jnth.1999.2388

A. Jamshidpey, N. Lemire, and É. Schost. 2018. Algebraic Construction of Quasi-split Algebraic Tori. *ArXiv:* 1801.09629 (2018).

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.

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.

E. Kaltofen and V. Shoup. 1998. Subquadratic-time factoring of polynomials over finite fields. *Math. Comp.* 67, 223 (1998), 1179–1197. https://doi.org/10.1090/S0025-5718-98-00944-2

E. Kaltofen and G. Villard. 2004. On the complexity of computing determinants. *Computational Complexity* 13, 3-4 (2004), 91–130. https://doi.org/10.1007/s00037-004-0185-3

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.

K. Kedlaya and C. Umans. 2011. Fast Polynomial Factorization and Modular Composition. *SICOMP* 40, 6 (2011), 1767–1802.

S. Lang. 2002. *Algebra* (third ed.). Graduate Texts in Mathematics, Vol. 211. Springer-Verlag, New York. xvi+914 pages. https://doi.org/10.1007/978-1-4613-0041-0

F. Le Gall. 2014. Powers of Tensors and Fast Matrix Multiplication. In *ISSAC'14*. ACM, Kobe, Japan, 296–303.

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.

H. W. Lenstra, Jr. 1991. Finding isomorphisms between finite fields. *Math. Comp.* 56, 193 (1991), 329–347. https://doi.org/10.2307/2008545

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.

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

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.

A. Poli. 1994. A deterministic construction for normal bases of abelian extensions. *Comm. Algebra* 22, 12 (1994), 4751–4757. https://doi.org/10.1080/00927879408825099

H. Schlickewei and S. Stepanov. 1993. Algorithms to construct normal bases of cyclic number fields. *J. Number Theory* 44, 1 (1993), 30–40. https://doi.org/10.1006/jnth.1993.1031

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

V. Shoup. 1995. A new polynomial factorization algorithm and its implementation. *J. Symbolic Comput.* 20, 4 (1995), 363–397. https://doi.org/10.1006/jsco.1995.1055

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