SUBQUADRATIC-TIME ALGORITHMS FOR NORMAL BASES

Mark Giesbrecht, Armin Jamshidpey,
and Éric Schost

November 20, 2020

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 testing whether a given $\alpha \in \mathsf{K}$ is normal, when G is either a finite abelian or a metacyclic group. The algorithm is based on the fact that deciding whether α is normal can be reduced to deciding whether $\sum_{g \in G} g(\alpha) g \in \mathsf{K}[G]$ is invertible; it requires a slightly subquadratic number of operations. Once we know that α is normal, we show how to perform conversions between the power basis of K/F and the normal basis with the same asymptotic cost.

Keywords. Normal bases; Galois groups; polycyclic groups; metacyclic groups; fast algorithms

Subject classification. 12Y05, 12F10, 11y16, 68Q25

1. Introduction

For a finite Galois field extension K/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 $G \cdot \alpha = \{g(\alpha) : g \in G\}$ forms a basis for K as a vector space over F. The existence of a 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 (e.g., (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, Lemire & Schost 2018).

A number of algorithms are available for finding a normal element in characteristic zero and in 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 & 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 & Shoup (1998), with a cost of $O(n^{1.82} \log q)$ operations in $\mathbb{F}_q$. In the bit complexity model, Kedlaya and Umans showed how to reduce the exponent of n to 1.63, by leveraging their quasi-linear time algorithm for *modular composition* (Kedlaya & 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 & 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 & 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 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 (1999) 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 that decides whether a given element in either an abelian or a metacyclic extension is normal, with a runtime subquadratic in the degree n of the extension. The costs of all algorithms are measured by counting *arithmetic operations* in 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 1.1. *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 $\beta = \sum_{0 \le i < n} \beta_i \xi^i \in \mathsf{K}$, the fact that g is an F-automorphism implies that $g(\beta)$ is equal to $\beta(\gamma)$, the polynomial composition of β (as a polynomial in ξ) at γ, reduced modulo P.

Our algorithms combine techniques and ideas of von zur Gathen & Giesbrecht (1990) and Kaltofen & 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 $\mathsf{K}[G]$. However, writing down S_α involves $\Theta(n^2)$ elements in F, which precludes a subquadratic runtime. 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 that, we adapt algorithms from (Kaltofen & Shoup 1998) that were developed 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 polynomial ring modulo an ideal $(x_i^{e_i} - 1)_{1 \le i \le m}$, where e_i's are positive integers. For metacyclic groups, we exploit the block-Hankel structure of the matrix of multiplication by $s_{\alpha,\ell}$.

These latter questions on the cost of arithmetic operations in $\mathsf{F}[G]$ are closely related to that of Fourier transforms 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, consider (Clausen & Müller 2004) and (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 (where we work over an arbitrary F, not necessarily algebraically closed).

We would like to thank the referee for pointing out that **?** address lower and upper bounds for multiplying in associative algebras and there is a possibility that these results could be applied to our problem of testing invertibility in $\mathsf{F}[G]$, though how to do this remains a topic for future consideration.

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 first 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 1.2. *Under Assumption 1.1, if G is either abelian or metacyclic, one can test whether $\alpha \in \mathsf{K}$ is normal using $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in F, where $(3/4)\cdot\omega(4/3) < 1.99$. The algorithms are randomized of the Monte Carlo type.*

Once α is known to be normal, we also discuss the cost of conversion between the power basis $1, \xi, \ldots, \xi^{n-1}$ of K and its normal basis $G \cdot \alpha$. The conversion problem between normal and power bases are discussed in Kaltofen & Shoup (1998) (randomized) and **?** (deterministic) with different assumptions. Inspired by previous work of Kaltofen & Shoup (1998), we obtain the following results.

THEOREM 1.3. *Under Assumption 1.1, if G is either abelian or metacyclic and $\alpha \in \mathsf{K}$ is known to be normal, we can perform basis conversion between the power basis $1, \xi, \ldots, \xi^{n-1}$ of K and its normal basis $G \cdot \alpha$ using $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in F. The algorithms are randomized of the Monte Carlo type.*

In both theorems, the runtime is barely subquadratic, and the exponent 1.99 is obtained through fast matrix multiplication algorithms that are most likely impractical for reasonable n. However, these results show in particular that we can perform basis conversions without writing down the normal basis itself (which would require $\Theta(n^2)$ elements in F).

REMARK 1.4. *Both above algorithms are randomized of the Monte Carlo type. In our model, this means that they are allowed to draw random elements for a prescribed subset of* F*, and for a control parameter* ϵ*, produce the correct answer with probability greater than* $1-\epsilon$ *(see Remark 2.8).*

Section 2 of this paper is devoted to definitions and preliminary discussions. In Section 3, a subquadratic-time algorithm is presented for the randomized reduction of our main question to invertibility testing in $\mathsf{F}[G]$; this algorithm applies to any finite polycyclic group, and in particular to abelian and metacyclic groups. In Section 4, we show that the problems of testing invertibility in $\mathsf{F}[G]$ and performing divisions can be solved in quasi-linear time for an abelian group; for metacyclic groups, we give a subquadratic time algorithm based on structured linear algebra algorithms (this will finish the proof of Theorem 1.2). Finally, Section 5 proves Theorem 1.3.

Our algorithms make extensive use of known algorithms for polynomial and matrix arithmetic; in particular, we use repeatedly the fact that polynomials of degree n in $\mathsf{F}[x]$, for any field F of characteristic zero, can be multiplied in $\tilde{O}(n)$ operations in F (Schönhage & 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 & Gerhard 2013). We also assume that generating a random element in F takes constant time.

For matrix arithmetic, we will rely on some non-trivial results on rectangular matrix multiplication initiated by Lotti & 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 ω). The sharpest values known to date for most rectangular formats are by Le Gall & Urrutia (2018); for $k = 1$, the best known value is $\omega \leq 2.373$ by Le Gall (2014). Over a field, further matrix operations (such as inversion) can also be done in $O(n^\omega)$ base field operations.

Part of the results of this paper (Theorem 1.2 for abelian groups) were already published in the conference paper (Giesbrecht *et al.* 2019).

2. Preliminaries

One of the well-known proofs of the existence of a normal element for a finite Galois extension, as for example reported by 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 $G = \{g_1, \ldots, g_n\}$. 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 each $i \in \{1, \ldots, n\}$, 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 $\mathbf{M}\boldsymbol{c} = \mathbf{0}$, with $\boldsymbol{c} = [c_1 \ \cdots \ c_n]^T$ and where, for $\alpha \in \mathsf{K}$,

$$\mathbf{M} = \begin{bmatrix} g_1g_1(\alpha) & g_1g_2(\alpha) & \cdots & g_1g_n(\alpha) \\ g_2g_1(\alpha) & g_2g_2(\alpha) & \cdots & g_2g_n(\alpha) \\ \vdots & \vdots & \vdots & \vdots \\ g_ng_1(\alpha) & g_ng_2(\alpha) & \cdots & g_ng_n(\alpha) \end{bmatrix} \in M_n(\mathsf{K}). \tag{2.3}$$

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

This approach can be used as the basis of a procedure to test if a given $\alpha \in \mathsf{K}$ is normal, by computing all the entries of the matrix $\mathbf{M}$ and using 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; the main contribution of this paper is to show how to speed up this verification.

Before entering that discussion, we briefly comment on the probability that α be a normal element: if we write $\alpha = a_0 + \cdots + a_{n-1}\xi^{n-1}$, the determinant of $\mathbf{M}$ 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 lemma implies that the probability that α be normal is at least $1 - n/|X|$.

If G is cyclic generated by an element g, with $g_1 = \mathrm{id}$ and $g_{i+1} = gg_i$ for all i, von zur Gathen & Giesbrecht (1990) avoid computing a determinant by computing the GCD of $S_\alpha := \sum_{i=1}^n g_i(\alpha)x^{i-1}$ and $x^n - 1$. In effect, this amounts to testing whether S_α 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, matrix $\mathbf{M}$ above is the matrix of left multiplication by the orbit sum

$$S_\alpha := \sum_{i=1}^{n} g_i(\alpha)g_i \in \mathsf{K}[G],$$

where we index rows by $g_1, \ldots, g_n$ and columns by their inverses $g_1^{-1}, \ldots, g_n^{-1}$. In terms of notation, for any field L (typically, we will take either $\mathsf{L} = \mathsf{F}$ or $\mathsf{L} = \mathsf{K}$), and β in $\mathsf{L}[G]$, we will write $\mathbf{M}_{\mathsf{L}}(\beta)$ for the left multiplication matrix by β in $\mathsf{L}[G]$, using the two bases shown above. In other words, the matrix $\mathbf{M}$ of (2.3) is $\mathbf{M}_{\mathsf{K}}(S_\alpha)$.

The previous discussion shows that 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 is the main new ingredient in our algorithm: it gives a randomized reduction to testing whether a suitable projection of S_α in $\mathsf{F}[G]$ is a unit.

LEMMA 2.4. *For $\alpha \in \mathsf{K}$, $\mathbf{M}_{\mathsf{K}}(S_\alpha)$ is invertible if and only if*

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

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

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

$$\sum_{k=0}^{n-1} a_{ijk}\xi^k,\ a_{ijk} \in F \tag{2.5}$$

and for $\ell : \mathsf{K} \to \mathsf{F}$, the corresponding entry in $\ell(\mathbf{M}_{\mathsf{K}}(S_\alpha))$ 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 in $\mathsf{K}[L_1, \ldots, L_n]$, we have $P(1, \xi, \ldots, \xi^{n-1}) = \det(\mathbf{M}_{\mathsf{K}}(S_\alpha))$, which is non-zero by assumption. Hence, P is not identically zero, and the conclusion follows.

($\Leftarrow$) Assume $\mathbf{M}_{\mathsf{K}}(S_\alpha)$ 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 $\mathbf{M}_{\mathsf{K}}(S_\alpha)$.

The elements of G act on rows of $\mathbf{M}_{\mathsf{K}}(S_\alpha)$ entrywise and the action permutes the rows the matrix. Assume $\varphi : G \to \mathfrak{S}_n$ (where $\mathfrak{S}_n$ is the full symmetric group) 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 $\mathbf{M}_{\mathsf{K}}(S_\alpha)$.

Since $\mathbf{M}_{\mathsf{K}}(S_\alpha)$ is singular, there exists a non-zero $\boldsymbol{v} \in \mathsf{K}^n$ such that $\mathbf{M}_{\mathsf{K}}(S_\alpha)\boldsymbol{v} = 0$; 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 $\boldsymbol{u} = v_i^{-1}\boldsymbol{v}$. Then, $\mathbf{M}_{\mathsf{K}}(S_\alpha)\boldsymbol{u} = 0$, 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 $\mathbf{M}_{\mathsf{K}}(S_\alpha)g(\boldsymbol{u}) = 0$, hence $g(\boldsymbol{u}) - \boldsymbol{u}$ is in the kernel of $\mathbf{M}_{\mathsf{K}}(S_\alpha)$. 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 for all choices of ℓ, $\ell(\mathbf{M}_{\mathsf{K}}(S_\alpha))$ is not invertible. By Equation (2.5), we can write

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

Since $\boldsymbol{u}$ has entries in F, $\mathbf{M}_{\mathsf{K}}(S_\alpha)\boldsymbol{u} = 0$ yields $\mathbf{M}^{(j)}\boldsymbol{u} = 0$ for

$j \in \{1, \dots, n\}$. Hence,

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

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

Our algorithm can be sketched as follows: given α in K, choose a random $\ell : \mathsf{K} \to \mathsf{F}$, and let

$$s_{\alpha,\ell} := \sum_{i=1}^{n} \ell(g_i(\alpha)) g_i \in \mathsf{F}[G]. \tag{2.6}$$

Note that $\ell(\mathbf{M}_{\mathsf{K}}(S_\alpha))$ is equal to $\mathbf{M}_{\mathsf{F}}(s_{\alpha,\ell})$, that is, the multiplication matrix by $s_{\alpha,\ell}$ in $\mathsf{F}[G]$, where, as above, we index rows by $g_1, \dots, g_n$ and columns by $g_1^{-1}, \dots, g_n^{-1}$. Then, the previous lemma can be rephrased as follows:

LEMMA 2.7. *For $\alpha \in \mathsf{K}$, α is normal if and only if $s_{\alpha,\ell}$ is invertible in $\mathsf{F}[G]$ for a generic F-linear projection $\ell : \mathsf{K} \to \mathsf{F}$.*

Thus, 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 $\mathsf{F}[G]$.

REMARK 2.8. *If α is not normal, S_α is not a unit. In this case, the proof of Lemma 2.4 established that $s_{\alpha,\ell}$ is not a unit for* any *ℓ, so our algorithm always returns the correct answer in this case.*

If α is normal, the polynomial P in the proof of Lemma 2.4, is a homogeneous polynomial of degree n in $(L_1, \dots, L_n)$. Thus, if we choose the coefficients of ℓ uniformly at random in any fixed finite subset $X \subset \mathsf{F}$, by the Lipton-DeMillo-Schwartz-Zippel lemma, we return the correct answer with probability at least $1 - n/|X|$.

3. Computing projections of the orbit sum

In this section we present an algorithm to compute $s_{\alpha,\ell}$ when $G = \{g_1, \dots, g_n\}$ is *polycyclic* (we give a definition of this family of groups and recall some well known results about them in

Subsection 3.2). To motivate our algorithm, we start by the simple case of a *cyclic* group. We will see that they follow closely ideas used by Kaltofen & Shoup (1998) over finite fields.

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

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

Kaltofen & 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

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

First, we compute all $G_i := g^i(\alpha)$ for $0 \le i < t$. Then we compute all $L_j := \ell \circ g^{tj}$ for $0 \le j < m$, 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 & 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 & Shoup 1998, Lemmata 3 and 4), 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 & Urrutia (2018), and the fact that $k \mapsto \omega(k)$ is convex (Lotti & Romani 1983). In particular, $3/4 \cdot \omega(4/3) < 1.99$. Note also the inequality $\omega(k) \ge 1 + k$ for $k \ge 1$, 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 1.1, which reduces automorphism evaluation to modular composition of polynomials. Over finite fields, this idea goes back to von zur Gathen & Shoup (1992), where it is 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 Brent and Kung's modular composition algorithm (Brent & Kung 1978). The algorithms below describe similar operations along these lines, involving several simultaneous evaluations. In this subsection, we work under Assumption 1.1 and we make no special assumption on G.

LEMMA 3.2. *Given $\alpha_1, \ldots, \alpha_s$ in 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 F.*

PROOF. (Compare (Kaltofen & Shoup 1998, Lemma 3)) As noted above, for $i \le s$, $g(\alpha_i) = \alpha_i(\gamma)$, 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

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

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 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

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

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 $(3/4) \cdot \omega(4/3) \ge 7/4$, the leading exponent in all costs seen so far is $(3/4) \cdot \omega(4/3)$. □

LEMMA 3.3. *Consider* $g_1, \ldots g_r$ *in* $G = \mathrm{Gal}(\mathsf{K}/\mathsf{F})$, *positive integers* $(s_1, \ldots s_r)$ *and elements* $\alpha_{i_1,\ldots,i_r}$ *in* K, *for* $i_m = 0, \ldots, s_m$, $m = 1, \ldots, r$. *If* $\prod_{i=1}^r s_i = O(\sqrt{n})$ *and* $r = O(\log(n))$, *we can compute*

$$g_r^{i_r} \cdots g_1^{i_1}(\alpha_{i_1,\ldots,i_r}) \text{ for } i_m = 0, \ldots, s_m,\ m = 1, \ldots, r$$

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

PROOF. Define $\mathcal{I} = \{(i_1, \ldots, i_r) \mid i_m = 0, \ldots, s_m \text{ for } m = 1, \ldots, r\}$. For $(i_1, \ldots, i_r)$ in $\mathcal{I}$ and non-negative integers $\ell_1, \ldots, \ell_r$, define

$$\alpha_{i_1,\ldots,i_r}^{(\ell_1,\ldots,\ell_r)} = g_r^{\ell_r} \cdots g_1^{\ell_1}(\alpha_{i_1,\ldots,i_r}).$$

Assume then that for some t in $\{0, \ldots, r-1\}$, we know

$$S_t = (\alpha_{i_1,\ldots,i_r}^{(i_1,\ldots,i_t,0,\ldots,0)} \ \mid\ (i_1, \ldots, i_r) \in \mathcal{I});$$

we show how to compute

$$S_{t+1} = (\alpha_{i_1,\ldots,i_r}^{(i_1,\ldots,i_{t+1},0,\ldots,0)} \ \mid\ (i_1, \ldots, i_r) \in \mathcal{I}).$$

Since our input is S_0, it will be enough to go through this process for all values of t to obtain the output S_r of the algorithm.

For a given index t, and for $m \geq 0$ define further

$$S_{t,m} = (\alpha_{i_1,\ldots,i_r}^{(i_1,\ldots,i_t,i_{t+1} \bmod 2^m,0,\ldots,0)} \ \mid\ (i_1, \ldots, i_r) \in \mathcal{I});$$

in particular, $S_{t,0} = S_t$ and $S_{t,\lfloor \log_2(s_{t+1})\rfloor+1} = S_{t+1}$. Hence, given $S_{t,m}$, it is enough to show how to compute $S_{t,m+1}$, for indices $m = 0, \ldots, \lfloor \log_2(s_{t+1})\rfloor$. This is done by writing

$$S_{t,m+1} = (\beta_{i_1,\ldots,i_r,t,m} \ \mid\ (i_1, \ldots, i_r) \in \mathcal{I}),$$

with

$$\beta_{i_1,\ldots,i_r,t,m} = \begin{cases} \alpha_{i_1,\ldots,i_r}^{(i_1,\ldots,i_t,i_{t+1} \bmod 2^m,0,\ldots,0)} & \text{if } i_{t+1} \bmod 2^{m+1} = i_{t+1} \bmod 2^m \\ g_{t+1}^{2^m}(\alpha_{i_1,\ldots,i_r}^{(i_1,\ldots,i_t,i_{t+1} \bmod 2^m,0,\ldots,0)}) & \text{otherwise.} \end{cases}$$

The automorphisms $g_{t+1}^{2^m}$ can be computed iteratively by modular composition; the bottleneck is the application of $g_{t+1}^{2^m}$ to a subset of

$S_{t,m}$. Using Lemma 3.2, since $S_{t,m}$ has $O(\sqrt{n})$ elements, this takes $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in F.

For a given index t, this is repeated $\lfloor \log_2(s_{t+1}) \rfloor \le \log_2(s_{t+1})+1$ times. Adding up for all indices t, this amounts to $O(\log(s_1 \cdots s_r)+r)$ repetitions, which is $O(\log(n))$ by assumption; the conclusion follows. □

We now present dual versions of the previous two lemmas (note that Kaltofen & Shoup (1998) also have 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 3.2 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 3.4. *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 $\ell_1 \circ g, \ldots, \ell_s \circ g$ using $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in F.*

Proof. Given ℓ_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 3.2. 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 α 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 3.3, whose proof follows the same pattern.

Lemma 3.5. *Consider* $g_1, \ldots, g_r$ *in* $G = \mathrm{Gal}(\mathsf{K}/\mathsf{F})$*, positive integers* $(s_1, \ldots, s_r)$ *and* F*-linear forms* $\ell_{i_1,\ldots,i_r}$*, for* $i_m = 0, \ldots, s_m$*,* $m = 1, \ldots, r$*. If* $\prod_{i=1}^r s_i = O(\sqrt{n})$ *and* $r = O(\log(n))$*, we can compute*

$$\ell_{i_1,\ldots,i_r} \circ g_r^{i_r} \cdots g_1^{i_1} \text{ for } i_m = 0, \ldots, s_m,\ m = 1, \ldots, r$$

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

Proof. We proceed as in Lemma 3.3, reversing the order of the steps. Using the same index set $\mathcal{I}$ as before, define, for $(i_1, \ldots, i_r)$ in $\mathcal{I}$ and non-negative integers $k_1, \ldots, k_r$

$$\ell_{i_1,\ldots,i_r}^{(k_1,\ldots,k_r)} = \ell_{i_1,\ldots,i_r} \circ g_r^{k_r} \cdots g_1^{k_1}.$$

For $t = r, \ldots, 0$, assuming that we know

$$L_{t+1} = (\ell_{i_1,\ldots,i_r}^{(0,\ldots,0,i_{t+1},\ldots,i_r)} \ \mid\ (i_1, \ldots, i_r) \in \mathcal{I}),$$

we compute

$$L_t = (\ell_{i_1,\ldots,i_r}^{(0,\ldots,0,i_t,i_{t+1},\ldots,i_r)} \ \mid\ (i_1, \ldots, i_r) \in \mathcal{I}).$$

This time, for $m \geq 0$, we set

$$L_{t+1,m} = (\ell_{i_1,\ldots,i_r}^{(0,\ldots,0,\lfloor i_t \rfloor_m,i_{t+1},\ldots,i_r)} \ \mid\ (i_1, \ldots, i_r) \in \mathcal{I}),$$

where for a non-negative integer x, $\lfloor x \rfloor_m = x - (x \bmod (2^m - 1))$ is obtained by setting to zero the coefficients of $1, 2, \ldots, 2^{m-1}$ in the base-two expansion of x.

Starting from $L_{t+1} = L_{t,\lceil \log_2(s_t) \rceil + 1}$, we compute all $L_{t+1,m}$ for $m = \lceil \log_2(s_t) \rceil, \ldots, 0$, since $L_{t+1,0} = L_t$. This is done essentially as in Lemma 3.3, but using Lemma 3.4 this time, in order to do right-composition by $g_t^{2^m}$. The cost analysis is as in Lemma 3.3. □

3.2. Computing the orbit sum projection for polycyclic groups. Our main algorithm in this section applies to a family of groups known as *polycyclic*; see (Holt *et al.* 2005, Chapter 8) for more details on such groups.

Our group G is called polycyclic if it has a normal series

$$G = G_r \trianglerighteq G_{r-1} \trianglerighteq \cdots \trianglerighteq G_1 \trianglerighteq G_0 = 1,$$

where G_j/G_{j-1} is cyclic; without loss of generality, we assume that $G_{j-1} \neq G_j$ holds for all j, so that r is $O(\log(n))$, with $n = |G|$. Finitely generated nilpotent or abelian groups are polycyclic. In general any finite solvable group is polycyclic; our key families of examples in the next section (abelian and metacyclic groups) thus fit into this category.

If G is polycyclic then, up to renumbering, its elements can be written as

$$g_r^{i_r} \cdots g_1^{i_1}, \text{ with } 0 \le i_j < e_j \text{ for } 1 \le j \le r,$$

where $G_j/G_{j-1} = \langle g_j G_{j-1} \rangle$ and $e_j = |G_j/G_{j-1}|$. Elements of $\mathsf{K}[G]$, or $\mathsf{F}[G]$ are written as polynomials $\sum_{i_1,\dots,i_r} c_{i_1,\dots,i_r} {g_r}^{i_r} \cdots {g_1}^{i_1}$, with $0 \le i_j < e_j$ for all j, and coefficients in either K or F.

PROPOSITION 3.6. *Suppose that G is polycyclic, 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 (2.6), can be computed using $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in F.*

PROOF. Our goal is to compute

$$\ell(g_r^{i_r} \dots g_1^{i_1}(\alpha)), \tag{3.7}$$

for all indices such that $0 \le i_j < e_j$ holds for $1 \le j \le r$; here, ℓ is an F-linear projection $\mathsf{K} \to \mathsf{F}$.

Our construction is inspired by that sketched in the cyclic case. Define z to be the unique index in $\{1, \dots, r\}$ such that $e_1 \cdots e_{z-1} < \sqrt{n}$ and $e_1 \cdots e_z \ge \sqrt{n}$. Then, all elements in (3.7) can be computed with the following steps, the sum of whose costs proves the proposition.

Step 1. Apply Lemma 3.3, with $\alpha_{i_1,\dots,i_r} = \alpha$ for all $i_1, \dots, i_r$, to get

$$G_{i_z,\dots,i_1} = g_z^{i_z} \cdots g_1^{i_1}(\alpha),$$

for all indices $i_1, \dots, i_z$ such that $0 \le i_m < e_m$ holds for $m = 1, \dots, z-1$ and $0 \le i_z < \lceil \sqrt{n}/(e_1 \cdots e_{z-1}) \rceil$. This amounts to taking $s_1 = e_1, \ldots, s_{z-1} = e_{z-1}$, $s_z = \lceil \sqrt{n}/(e_1 \cdots e_{z-1}) \rceil$ and $s_m = 1$ for $m > z$ in the lemma. For the lemma to apply, we have to check that the product of these indices $s_1, \ldots, s_r$ is $O(\sqrt{n})$. Indeed, this product is at most

$$e_1 \cdots e_{z-1} \left(\frac{\sqrt{n}}{e_1 \cdots e_{z-1}} + 1 \right) \leq \sqrt{n} + e_1 \cdots e_{z-1} \leq 2\sqrt{n}.$$

Hence, the lemma applies, and the cost of this step is $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$.

Step 2. Compute $G_z = g_z^{s_z}$, for s_z as above. The cost is that of $O(\log(n))$ modular compositions, which is negligible compared to the cost of the previous step.

Step 3. Use Lemma 3.5 with $\ell_{i_r,\ldots,i_1} = \ell$ for all $i_1, \ldots, i_r$, to compute

$$\begin{aligned} L_{j_r,\ldots,j_z} &= \ell \circ (g_r^{j_r} \cdots g_{z+1}^{j_{z+1}} G_z^{j_z}) \\ &= \ell \circ (g_r^{j_r} \cdots g_{z+1}^{j_{z+1}} g_z^{s_z j_z}), \end{aligned}$$

for all indices $0 \leq j_z < \lceil e_z/s_z \rceil$ and $0 \leq j_m < e_m$ for $m > z$. This amounts to using the lemma with indices $s'_1 = \cdots = s'_{z-1} = 1$, $s'_z = \lceil e_z/s_z \rceil$ and $s'_m = e_m$ for $m > z$. Again, we have to verify that $s'_1 \cdots s'_r$ is $O(\sqrt{n})$. Indeed, we have

$$\begin{aligned} s'_1 \cdots s'_r = \left\lceil \frac{e_z}{s_z} \right\rceil e_{z+1} \cdots e_r &\leq \left(\frac{e_z}{s_z} + 1 \right) e_{z+1} \cdots e_r \\ &\leq \frac{e_z \cdots e_r}{s_z} + e_{z+1} \cdots e_r. \end{aligned}$$

By definition, we have $s_z \geq \sqrt{n}/(e_1 \cdots e_{z-1})$, so $e_z \cdots e_r/s_z \leq e_1 \cdots e_r/\sqrt{n} = \sqrt{n}$. Because we assume $e_1 \cdots e_z \geq \sqrt{n}$, the second term is also at most $\sqrt{n}$, so the product $s'_1 \cdots s'_r$ is at most $2\sqrt{n}$. Hence, Lemma 3.5 applies, and computes all $L_{j_r,\ldots,j_z}$ using $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in F.

Step 4. Multiply the matrix with rows the coefficients of all $L_{j_r,\ldots,j_z}$ by the matrix whose columns are the coefficients of all $G_{i_z,\ldots,i_1}$. This yields the values

$$\ell(g_r^{j_r} \cdots g_{z+1}^{j_{z+1}} g_z^{s_z j_z + i_z} g_{z-1}^{i_{z-1}} \cdots g_1^{i_1}(\alpha)),$$

for indices as follows:

- $0 \le i_m < e_m$ for $m = 0, \ldots, z-1$;
- $0 \le i_z < s_z$ and $0 \le j_z < \lceil e_z/s_z \rceil$;
- $0 \le j_m < e_m$ for $m = z+1, \ldots, r$.

This shows that we obtain all required values. We compute this product in $O(n^{(1/2)\cdot\omega(2)})$ operations in F, which is in $O(n^{(3/4)\cdot\omega(4/3)})$. $\square$

4. Arithmetic in the Group Algebra

In this section we consider the problems of invertibility testing and division in $\mathsf{F}[G]$: given elements β, η in $\mathsf{F}[G]$, for a field F and a group G, determine whether β is a unit in $\mathsf{F}[G]$, and if so, compute $\beta^{-1}\eta$. We focus on two particular families of polycyclic groups, namely abelian and metacyclic groups 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 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 $\beta_1 \in M_{d_1}(\mathbb{C}), \ldots, \beta_r \in M_{d_r}(\mathbb{C})$ are invertible. This boils down to linear algebra over $\mathbb{C}$, and takes $O(d_1^\omega + \cdots + d_r^\omega)$ operations. Since $d_1^2 + \cdots + d_r^2 = n$, with $n = |G|$, this is $O(n^{\omega/2})$ 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.

In this section, we give algorithms for the problems of invertibility testing and division for the two particular families of polycyclic groups mentioned so far, namely abelian and metacyclic. For the former, 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 also makes 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 rely on the block-Hankel structure of the matrix of multiplication by β. Through structured linear algebra algorithms, this allows us to solve both problems (invertibility and division) in subquadratic (albeit not softly-linear time) 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 β in $\mathsf{F}[G]$, our goal in this section is to determine whether β is a unit, and if so to compute expressions such as $\beta^{-1}\eta$, for η in $\mathsf{F}[G]$.

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 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 test or division, 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 test or division in algebras of the form $\mathsf{F}[z]/\langle P_i(z)\rangle$, for various polynomials P_i. 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 $n = |G|$).

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 $i \geq 0$, 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* $\gamma : \mathbb{h} \to \mathsf{F}[z]/\langle \Phi_{mm'}(z) \rangle$ *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 $\mathbb{h}$, 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 & 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 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 $\varphi(mm')$. $\square$

Extension to several cyclotomic rings. The natural generalization of the algorithm above starts with pairwise distinct primes $\boldsymbol{p} = (p_1, \ldots, p_t)$, non-negative exponent $\boldsymbol{c} = (c_1, \ldots, c_t)$ and variables $\boldsymbol{x} = (x_1, \ldots, x_t)$ over F. Now, we define

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

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

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

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. $\square$

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 F. Define

$$\mathbb{a} := \mathsf{F}[x]/\Phi_{p^c}(x), \tag{4.3}$$

$$\mathbb{b} := \mathsf{F}[x, y]/\langle \Phi_{p^c}(x), \Phi_{p^{c'}}(y)\rangle = \mathbb{a}[y]/\Phi_{p^{c'}}(y). \tag{4.4}$$

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

LEMMA 4.5. *There is an* F*-algebra isomorphism* $\theta : \mathbb{b} \to \mathbb{a}^{\varphi(p^{c'})}$ *such that one can apply* θ *or its inverse to any inputs using* $\tilde{O}(\dim(\mathbb{b}))$ *operations in* F*.*

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

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

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 ρ_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 θ and its inverse is done by means of fast evaluation and interpolation (von zur Gathen & 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 $c_1 \ge c_i$ holds for all i, and define as before $\mathbb{a} := \mathsf{F}[x_1]/\Phi_{p^{c_1}}(x_1)$.

LEMMA 4.6. *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

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

Θ_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 Θ_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

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

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 4.7. *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 & 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, λ can be computed in $\tilde{O}(\dim(\mathbb{B}))$ operations in F. Composing with the result in Lemma 4.6, 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, 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 $n = |G|$ writes $n = p_1^{a_1} \cdots p_s^{a_s}$. 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 $b_{i,1} \le \cdots \le b_{i,t_i}$; 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 $\gamma_{1,1}, \dots, \gamma_{s,t_s}$ 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 $\gamma_{1,1}, \dots, \gamma_{s,t_s}$. Were this not the case, given arbitrary generators $g_1, \dots, g_r$ of G, with orders $e_1, \dots, e_r$, 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 4.8. *Given $\beta \in \mathsf{F}[G]$, written on the power basis $\gamma_{1,1}, \dots, \gamma_{s,t_s}$, one can test if β is a unit in $\mathsf{F}[G]$ using $\tilde{O}(n)$ operations in F. If it is the case, given η in $\mathsf{F}[G]$, one can compute $\beta^{-1}\eta$ in the same asymptotic runtime.*

In view of Lemma 2.7, Proposition 3.6 and the claim on the cost of invertibility testing prove the first part of Theorem 1.2; the second part of this proposition will allow us to prove Theorem 1.3 in the next section.

The proof of the proposition 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 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 $i \le s$, by Lemma 4.7, 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

$$\prod_{\boldsymbol{j}} \mathsf{F}[z_1, \dots, z_s]/\langle \Phi_{{p_1}^{c_{1,j_1}}}(z_1), \dots, \Phi_{{p_s}^{c_{s,j_s}}}(z_s)\rangle, \tag{4.9}$$

for 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 4.7 also implies that $B' := \Gamma(B)$ can be computed in softly linear time $\tilde{O}(n)$ (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 (4.9) 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

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

for certain integers $d_{\boldsymbol{j}}$. The lemma implies that given B', $B'' := \Gamma'(B')$ can be computed in softly linear time $\tilde{O}(n)$ as well. Invertibility of $\beta \in \mathsf{F}[G]$ is equivalent to B'' 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 (von zur Gathen & Gerhard 2013, Chapter 11), so the first part of the proposition follows.

Given η in $\mathsf{F}[G]$, we can similarly compute its image H'' in $\prod_{\boldsymbol{j}} \mathsf{F}[z]/\langle \Phi_{d_{\boldsymbol{j}}}(z)\rangle$, with the same asymptotic runtime as for β. If we suppose β (and thus B'') invertible, division in each $\mathsf{F}[z]/\langle \Phi_{d_{\boldsymbol{j}}}(z)\rangle$ takes softly linear time in the degree $\phi_{d_{\boldsymbol{j}}}$; as a result, we obtain $B''^{-1}H''$ in time $\tilde{O}(n)$. One can finally invert all isomorphisms we applied, in order to recover $\beta^{-1}\eta$ in $\mathsf{F}[G]$; this also takes time $\tilde{O}(n)$. Summing all costs, this establishes the second part of the proposition.

4.2. Metacyclic Groups. In this subsection, we study the invertibility and division problems for a metacyclic group G. A group G is metacyclic if it has a normal cyclic subgroup H such that G/H is cyclic: this is the case $r = 2$ in the definition we gave of polycyclic groups. For instance, any group with a squarefree order is metacyclic (see (Johnson 1976, p. 88) or (Curtis & Reiner 1988, p. 334) for more background).

For such groups, we will use a standard specific notation, rather than the general one introduced in (3.2) for arbitrary polycyclic ones: we will write (σ, τ) instead of (g_1, g_2) and (m, s) instead of (e_1, e_2). Then, a metacyclic group G can be presented as

$$\langle \sigma, \tau : \sigma^m = 1, \tau^s = \sigma^t, \tau^{-1}\sigma\tau = \sigma^u \rangle, \tag{4.10}$$

for integers m, t, u, s, with $u, t \leq m$ and $u^s = 1 \bmod t$, $ut = t \bmod m$. 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 (4.10), $n = |G|$ is equal to ms, and all elements in a metacyclic group can be presented uniquely as either

$$\{\sigma^i\tau^j,\ 0 \leq i \leq m-1,\ 0 \leq j \leq s-1\} \tag{4.11}$$

or

$$\{\tau^j\sigma^i,\ 0 \leq i \leq m-1,\ 0 \leq j \leq s-1\}. \tag{4.12}$$

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 F, using the commutation relation $\sigma^k\tau^c = \tau^c\sigma^{ku^c}$ for $k, c \geq 0$.

To test invertibility in $\mathsf{F}[G]$, a possibility would be to rely on the Wedderburn decomposition of $\mathsf{F}[G]$, but the structure of group algebras of metacyclic groups is not straightforward to exploit; see for instance (Curtis & Reiner 1988, §47) for algebraically closed F, or, when $\mathsf{F} = \mathbb{Q}$, (Vergara & Martínez 2002) for dihedral and quaternion groups. Instead, we will highlight the structure of the multiplication matrices in $\mathsf{F}[G]$.

Take β in $\mathsf{F}[G]$. In eq. (2.3), we introduced the matrix $\mathbf{M}_{\mathsf{F}}(\beta)$ of left multiplication by β in $\mathsf{F}[G]$, where columns and rows were indexed using an arbitrary ordering of the group elements. We will now reorder the rows and columns of $\mathbf{M}_{\mathsf{F}}(\beta)$ using the two presentations of G seen in (4.11) and (4.12), in order to highlight its block structure. In what follows, for non-negative integers a, b, c, we will write $\beta_{a,b,c}$ for the coefficient of $\tau^a\sigma^b\tau^c$ in the expansion of β on the F-basis of $\mathsf{F}[G]$.

We first rewrite $\mathbf{M}_{\mathsf{F}}(\beta)$ by reindexing its columns by

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

and its rows by

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

This matrix displays a $s \times s$ block structure. Each block has itself size $m \times m$; for $1 \le u, v \le s$ and $1 \le a, b \le m$, the entry of index (a, b) in the block of index (u, v) is the coefficient of $\tau^u\sigma^a\sigma^b\tau^v$ in β, that is, $\beta_{u,a+b,v}$. In other words, all blocks are Hankel matrices.

Using the algorithm of Bostan *et al.* (2017) (see also (Eberly *et al.* 2007, Appendix A)), this structure allows us to solve a system such as $\mathbf{M}_{\mathsf{F}}(\beta)\boldsymbol{x} = \boldsymbol{y}$ in Las Vegas time $\tilde{O}(s^{\omega-1}n)$ (or raise an error if there is no solution). In addition, if the right-hand side is zero and $\mathbf{M}_{\mathsf{F}}(\beta)$ is not invertible, the algorithm returns a non-zero kernel element. This last remark allows us to test whether β is invertible in Las Vegas time $\tilde{O}(s^{\omega-1}n)$; if so, given the coefficient vector $\boldsymbol{y}$ of some η in $\mathsf{F}[G]$, we can compute $\beta^{-1}\eta$ in the same asymptotic runtime.

It is also possible to reorganize the rows and columns of $\mathbf{M}_{\mathsf{F}}(\beta)$, using indices

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

for its columns and

$$\begin{bmatrix}\sigma^0\tau^0 & \cdots & \sigma^{m-1}\tau^0\cdots & \sigma^0\tau^{s-1} & \cdots & \sigma^{m-1}\tau^{s-1}\end{bmatrix}$$

for its rows. The resulting matrix has an $m \times m$ block structure, where each $s \times s$ block is Hankel. As a result, it allows us to solve the problems above, this time using $\tilde{O}(m^{\omega-1}n)$ operations in F. Since we have either $s \le \sqrt{n}$ or $m \le \sqrt{n}$, this implies the following.

PROPOSITION 4.13. *Given $\beta \in \mathsf{F}[G]$, one can test if β is a unit in $\mathsf{F}[G]$ using $\tilde{O}(n^{(\omega+1)/2})$ operations in F. If it is the case, given η in $\mathsf{F}[G]$, one can compute $\beta^{-1}\eta$ in the same asymptotic runtime.*

Combined with Proposition 3.6, the former statement provides the last part of the proof of Theorem 1.2.

5. Basis Conversion

We conclude this paper with algorithms for basis conversion: assuming we know that α is normal, we show how to perform the change-of-basis between the power basis of K/F and the normal basis $G \cdot \alpha$. The techniques used below are inspired by those used by (Kaltofen & Shoup 1998, Section 4) in the case of extensions of finite fields.

5.1. From normal to power basis. Suppose $G = \{g_1, \ldots, g_n\}$, α is a normal element of K/F and we are given $u \in \mathsf{K}$ as $u = \sum_{i=1}^{n} u_i g_i(\alpha)$. In order to write u in the power basis, we have to compute the matrix-vector product

$$\begin{bmatrix} \boldsymbol{\gamma_1} & \cdots & \boldsymbol{\gamma_n} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}, \tag{5.1}$$

where for $i = 1, \ldots, n$, $\boldsymbol{\gamma}_i \in \mathsf{F}^{n\times 1}$ is the coefficient vector of $g_i(\alpha)$. As already pointed out by Kaltofen and Shoup for finite fields, this shows that conversion from normal to power basis is the transpose problem of computing the "projected" orbit sum $s_{\alpha,\ell}$, which we solved in Section 3.

The transposition principle then allows us to derive runtime estimates for the conversion problem; below, we present an explicit procedure derived from the algorithm in Subsection 3.2. As in that section, we give the algorithm in the general case of a polycyclic group G presented as

$$G = \{g_r^{i_r} \cdots g_1^{i_1}, \text{ with } 0 \le i_j < e_j \text{ for } 1 \le j \le r\}.$$

With indices $i_1, \ldots, i_r$ as above, we are given a family of coefficients $u_{i_1,\ldots,i_r}$ in F, and we expand the sum $u = \sum_{i_1,\ldots,i_r} u_{i_1,\ldots,i_r} g_r^{i_r} \cdots g_1^{i_1}(\alpha)$ on the power basis of K/F. For this, we let $z \in \{1, \ldots, r\}$ be the index defined in Subsection 3.2.

Step 1. Apply Lemma 3.3, with $\alpha_{i_1,\ldots,i_r} = \alpha$ for all $i_1, \ldots, i_r$, to get

$$G_{i_z,\ldots,i_1} = g_z^{i_z} \cdots g_1^{i_1}(\alpha),$$

for all indices $i_1, \ldots, i_z$ such that $0 \le i_m < e_m$ holds for $m = 1, \ldots, z-1$ and $0 \le i_z < s_z = \lceil \sqrt{n}/(e_1 \cdots e_{z-1}) \rceil$. As in Subsection 3.2, the cost of this step is $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$.

Step 2. Compute $G_z = g_z^{s_z}$, for s_z as above. The cost is is negligible compared to the cost of the previous step.

Step 3. Compute the matrix product $\mathbf{U}\mathbf{\Gamma}$, where

- $\mathbf{U}$ is the matrix over F having $\lceil e_z/s_z \rceil e_{z+1} \cdots e_r$ rows and $e_1 \cdots e_{z-1} s_z$ columns built as follows. Rows are indexed by $(j_z, \ldots, j_r)$, with $0 \le j_z < \lceil e_z/s_z \rceil$ and $0 \le j_m < e_m$ for all other indices; columns are indexed by $(i_1, \ldots, i_z)$, with $0 \le i_z < s_z$ and $0 \le i_m < e_m$ for all other indices; the entry at rows $(j_z, \ldots, j_r)$ and column $(i_1, \ldots, i_z)$ is $u_{i_1,\ldots,i_z+s_z j_z, j_{z+1},\ldots,j_r}$.
- $\mathbf{\Gamma}$ is the matrix with $e_1 \cdots e_{z-1} s_z$ rows (indexed in the same way as the columns of $\mathbf{U}$) and n columns, whose row of index $(i_1, \ldots, i_z)$ contains the coefficients of $G_{i_z,\ldots,i_1}$ (on the power basis of K)

As established in Subsection 3.2, the row and column dimensions of $\mathbf{U}$ are $O(\sqrt{n})$, so this product can be computed in $O(n^{(1/2)\cdot\omega(2)})$ operations in F. The rows of the resulting matrix give the coefficients of

$$H_{j_{z+1},\ldots,j_r} = \sum_{i_1,\ldots,i_z} u_{i_1,\ldots,i_z+s_z j_z,\ldots,j_r} g_z^{i_z} \cdots g_1^{i_1}(\alpha),$$

for all indices $(j_z, \dots, j_r)$ and $(i_1, \dots, i_z)$ as above.

Step 4. Compute and add all

$$g_r^{j_1} \cdots g_{z+1}^{j_{z+1}} G_z^{j_z}(H_{j_{z+1},\dots,j_r}),$$

for indices $(j_z, \dots, j_r)$ as above; their sum is precisely the input element $u = \sum_{i_1,\dots,i_r} u_{i_1,\dots,i_r} g_r^{i_r} \cdots g_1^{i_1}(\alpha)$, written on the power basis.

This is done by a second call to Lemma 3.3, for the same asymptotic cost as in Step 1. Summing all costs, we arrive at an overall runtime of $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in F for the conversion from normal to power basis. This proves the first half of Theorem 1.3.

5.2. Power basis to normal basis. Now assume $u \in \mathsf{K}$ is given in the power basis. Still writing the elements of G as $g_1, \dots, g_n$, the goal is to find coefficients c_i's in F such that

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

Starting from this equality, for any element g_j of G, we have

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

Then, if ℓ is a random F-linear projection $\mathsf{K} \to \mathsf{F}$, we get

$$\sum_{i=1}^{n} c_i \ell(g_j g_i(\alpha)) = \ell(g_j(u)), \quad 1 \le j \le n.$$

Introducing

$$u' = \sum_{i=1}^{n} c_i g_i^{-1} \in \mathsf{F}[G]$$

and writing as before

$$s_{\alpha,\ell} = \sum_{j=1}^{n} \ell(g_j(\alpha)) g_j \quad \text{and} \quad s_{u,\ell} = \sum_{j=1}^{n} \ell(g_j(u)) g_j \quad \text{in } \mathsf{F}[G],$$

the n equations above are equivalent to the equality $s_{\alpha,\ell}\, u' = s_{u,\ell}$ in $\mathsf{F}[G]$.

We use the algorithm of Section 3 to compute both $s_{\alpha,\ell}$ and $s_{u,\ell}$; this takes $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ operations in F, for G polycyclic. If α is normal, $s_{\alpha,\ell}$ is a unit for a generic ℓ. Then, if we further assume that G is either abelian or metacyclic, it suffices to apply the division algorithms given in the previous section to recover u', and thus all coefficients $c_1, \ldots, c_n$. In both cases, the runtime of the division is negligible compared to the cost $\tilde{O}(n^{(3/4)\cdot\omega(4/3)})$ of the first step. Altogether, this finishes the proof of Theorem 1.3.

References

D. Augot & P. Camion (1994). A deterministic algorithm for computing a normal basis in a finite field. In *Proc. EUROCODE'94*, P. Charpin, editor.

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

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

A. Bostan, C.-P. Jeannerod, C. Mouilleron & É. Schost (2017). On Matrices With Displacement Structure: Generalized Operators and Faster Algorithms. *SIAM Journal on Matrix Analysis and Applications* **38**(3), 733–775.

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

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

M. Clausen & M. Müller (2004). Generating fast Fourier transforms of solvable groups. *J. Symbolic Comput.* **37**(2), 137–156. ISSN 0747-7171.

C. Curtis & I. Reiner (1988). *Representation theory of finite groups and associative algebras.* Wiley Classics Library. John Wiley & Sons, Inc., New York, New York. ISBN 0-471-60845-9, xiv+689 .

X. Dahan, M. Moreno Maza, É. Schost & Y. Xie (2006). On the complexity of the D5 principle. In *Proc. of* Transgressive Computing 2006. Granada, Spain.

W. Eberly, M. Giesbrecht, P. Giorgi, A. Storjohann & G. Villard (2007). Faster Inversion and Other Black Box Matrix Computations Using Efficient Block Projections. In *ISSAC '07*, 143–150. ACM.

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

J. von zur Gathen & J. Gerhard (2013). *Modern Computer Algebra (third edition)*. Cambridge University Press, Cambridge, U.K. ISBN 9781107039032.

J. von zur Gathen & M. Giesbrecht (1990). Constructing normal bases in finite fields. *J. Symbolic Comput.* **10**(6), 547–570. ISSN 0747-7171.

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

M. Giesbrecht, A. Jamshidpey & É. Schost (2019). Quadratic-Time Algorithms for Normal Elements. In *ISSAC '19*, 179–186. ACM. URL http://doi.acm.org/10.1145/3326229.3326260.

K. Girstmair (1999). An algorithm for the construction of a normal basis. *J. Number Theory* **78**(1), 36–45. ISSN 0022-314X.

D. Holt, B. Eick & E. O'Brien (2005). *Handbook of computational group theory*. Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton, FL. ISBN 1-58488-372-3.

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

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

E. Kaltofen & V. Shoup (1998). Subquadratic-time factoring of polynomials over finite fields. *Math. Comp.* **67**(223), 1179–1197. ISSN 0025-5718.

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

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

S. Lang (2002). *Algebra*, volume 211 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 3rd edition.

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

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

H. W. Lenstra, Jr. (1991). Finding isomorphisms between finite fields. *Math. Comp.* **56**(193), 329–347. ISSN 0025-5718.

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

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

D. Maslen, D. N. Rockmore & S. Wolff (2018). The efficient computation of Fourier transforms on semisimple algebras. *J. Fourier Anal. Appl.* **24**(5), 1377–1400. ISSN 1069-5869.

A. Poli (1994). A deterministic construction for normal bases of abelian extensions. *Comm. Algebra* **22**(12), 4751–4757. ISSN 0092-7872.

H. Schlickewei & S. Stepanov (1993). Algorithms to construct normal bases of cyclic number fields. *J. Number Theory* **44**(1), 30–40. ISSN 0022-314X.

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

V. Shoup (1995). A new polynomial factorization algorithm and its implementation. *J. Symbolic Comput.* **20**(4), 363–397. ISSN 0747-7171.

C. Giraldo Vergara & F. Brochero Martínez (2002). Wedderburn decomposition of some special rational group algebras. *Lect. Mat.* **23**(2), 99–106. ISSN 0120-1980.

Manuscript received

Mark Giesbrecht
Cheriton School of Computer Science University of Waterloo
Waterloo, ON, Canada N2L 3G1
mwg@uwaterloo.ca
https://cs.uwaterloo.ca/~mwg/

Armin Jamshidpey
Cheriton School of Computer Science University of Waterloo
Waterloo, ON, Canada N2L 3G1
armin.jamshidpey@uwaterloo.ca
https://cs.uwaterloo.ca/~a5jamshi/

Éric Schost
Cheriton School of Computer Science University of Waterloo
Waterloo, ON, Canada N2L 3G1
eschost@uwaterloo.ca
https://cs.uwaterloo.ca/~eschost/