COMPUTING THE TORSION POINTS OF A VARIETY DEFINED BY LACUNARY POLYNOMIALS

LOUIS LEROUX

ABSTRACT. We present an algorithm for computing the set of torsion points satisfying a given system of multivariate polynomial equations. Its complexity is quasilinear in the logarithm of the degree and in the height of the input equations but exponential in their number of variables and nonzero terms.

1. INTRODUCTION AND STATEMENT OF RESULTS

Let \( F_1, \ldots, F_k \in \mathbb{Z}[X_1, \ldots, X_n] \) be a family of multivariate polynomials. We consider the problem of computing the solutions of the system of equations

\[ F_1(X_1, \ldots, X_n) = \cdots = F_k(X_1, \ldots, X_n) = 0 \]

in roots of unity. This problem arises naturally when solving trigonometric equations and also when solving some geometric problems; see for instance \([CJ76]\), \([PR98]\).

For \( n = 1 \), this is equivalent to the problem of finding the common cyclotomic factors of \( F_1, \ldots, F_k \). Based on a result of J.H. Conway and A.J. Jones \([CJ76]\), M. Filaseta, A. Granville and A. Schinzel have recently described an algorithm in \([FGS08]\), that solves this problem with a complexity quasilinear in the logarithm of the degree of the input polynomials.

For \( n \geq 2 \), the system (1) can have an infinite number of solutions in roots of unity but a structure theorem of M. Laurent \([Lau84]\) implies that this set can in principle be described in finite terms (see below for more details). In this text, we first simplify the algorithm of \([FGS08]\) and we make its complexity precise (see Algorithm 1 in Section 2); then we extend it to the general multivariate case (see Algorithm 2 in Section 3). The complexity of this algorithm is again quasilinear in the logarithm of the degree of the input polynomials, although exponential in their number of variables and their number of nonzero terms. Hence our algorithm can be regarded as an effective version of Laurent’s theorem, which is particularly well suited for lacunary polynomials, that is, polynomials having a small number of nonzero terms but potentially large degree.

In the following, we fix \( n \in \mathbb{N}^* \). We begin by recalling some basic facts on Diophantine geometry of subvarieties of the algebraic torus \( \mathbb{G}_m^n := (\mathbb{Q}^*)^n \). All the definitions and results stated here can also be found in \([S96]\) or in \([Z09]\) and we...
refer to these texts for the proofs and for more details. The set $\mathbb{G}_m^n$ is a group with the usual multiplication:

$$(x_1, \ldots, x_n) \cdot (y_1, \ldots, y_n) = (x_1y_1, \ldots, x_ny_n).$$

An algebraic subgroup $H \subset \mathbb{G}_m^n$ is a subgroup of $\mathbb{G}_m^n$ which is an algebraic variety, i.e., it is Zariski closed. A point of finite order in the group $\mathbb{G}_m^n$ will be called a torsion point. The set of torsion points in $\mathbb{G}_m^n$ is exactly $\mu^n_\infty$, where $\mu_\infty$ denotes the set of roots of unity in $\mathbb{Q}$. A torsion coset is a coset of the form

$$\eta H := \{ \eta \cdot h \mid h \in H \},$$

where $H$ is an algebraic subgroup and $\eta \in \mu^n_\infty$. For $V$ a subvariety of $\mathbb{G}_m^n$, we set

$$V_{\text{tors}} := V \cap \mu^n_\infty,$$

for the set of torsion points lying in $V$. A celebrated theorem of Laurent [Lau84] asserts that the Zariski closure $\overline{V_{\text{tors}}}$ of $V_{\text{tors}}$ is the union of a finite number of torsion cosets:

$$\overline{V_{\text{tors}}} = B_1 \cup \cdots \cup B_t. \tag{2}$$

This result was previously conjectured by S. Lang [Lan83]. It was proved by Y. Ihara, J.P. Serre and J. Tate when $V$ is a curve [Lan83] and by Laurent in the general case.

For $x = (x_1, \ldots, x_n) \in \mathbb{G}_m^n$ and $\Lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{Z}^n$, we set:

$$x^\Lambda := x_1^{\lambda_1} \cdots x_n^{\lambda_n} \in \mathbb{G}_m.$$

If $\Lambda$ is a subgroup of $\mathbb{Z}^n$ of dimension $k$, then the set

$$H_\Lambda := \{ x \in \mathbb{G}_m^n \mid x^\Lambda = 1, \forall \Lambda \in \Lambda \}$$

is an algebraic subgroup of $\mathbb{G}_m^n$ of dimension $n - k$. Furthermore, the map $\Lambda \mapsto H_\Lambda$ is a bijection between the set of subgroups of $\mathbb{Z}^n$ and the set of algebraic subgroups of $\mathbb{G}_m$ [S96, Lemma 2].

Let $M_{k,n}(\mathbb{Z})$ be the set of matrices with $k$ rows and $n$ columns with integer coefficients. For $L = (\lambda_{i,j})_{1 \leq i \leq k, 1 \leq j \leq n} \in M_{k,n}(\mathbb{Z})$, and $x = (x_1, \ldots, x_n) \in \mathbb{G}_m^n$ we define

$$x^L := x_1^{\lambda_{11}} \cdots x_n^{\lambda_{kn}} \in \mathbb{G}_m^k,$$

where $\lambda_1, \ldots, \lambda_k$ are the rows of the matrix $L$. Now let $\Lambda$ be a subgroup of $\mathbb{Z}^n$ of dimension $k$, let $\{\lambda_1, \ldots, \lambda_k\}$ be a $\mathbb{Z}$-basis of $\Lambda$ and let $L$ be the matrix of size $k \times n$ whose rows consist of the vectors $\lambda_1, \ldots, \lambda_k$. Also, let $\omega \in \mu^k_\infty$; from the previous discussion, the set

$$B(L, \omega) := \{ x \in \mathbb{G}_m^n \mid x^L = \omega \} \tag{3}$$

is a torsion coset and, in fact, all torsion cosets can be described in this way. Our algorithm takes as input a finite number of polynomials and outputs a finite number of torsion cosets whose union is $\overline{V_{\text{tors}}}$ as in (2). Furthermore, each torsion coset of dimension $n - r$ will be represented by a pair $(L, \omega) \in M_{k,n}(\mathbb{Z}) \times \mu^k_\infty$ as in (3). We remark that it is easy to describe all the torsion points lying in such a coset since

$$B(L, \omega)_{\text{tors}} := \{ \zeta \in \mu^k_\infty \mid \zeta^L = \omega \}.$$
Let us now describe the input of the algorithm. Let
\[ F(X_1, \ldots, X_n) = \sum_{i=1}^{N} a_i X_1^{\alpha_i} \]
be a polynomial in \( n \) variables with integer coefficients. The *lacunary encoding* of
\( F \) is the list
\[ [(a_1, \alpha_1), \ldots, (a_N, \alpha_N)] \]
of its nonzero coefficients with corresponding exponents and its *height* is defined by:
\[ h(F) := \max_{1 \leq i \leq N} \{ \log |a_i| \} \].
If \( F \) is of total degree \( d \) and of height \( h \), the number of bits to encode this representation is
\[ O(N(nkN(M \log(d))) \log \log(d) + h) \].

When \( \ell \in \mathbb{N} \), we set \( M(\ell) \) for the complexity of multiplying two integers of bit
length smaller than \( \ell \). We can now state our main result.

**Theorem 1.1.** There exists a deterministic algorithm which has the following property. Let \( F_1, \ldots, F_k \in \mathbb{Z}[X_1, \ldots, X_n] \) be polynomials given by their lacunary encoding and let \( V \) be the variety defined by those polynomials. The algorithm computes a family of torsion cosets \( B_1, \ldots, B_t \) such that
\[ \bigcup_{1 \leq i \leq t} B_i = V_{\text{tors}} \]
where the torsion cosets are represented as in (3). Furthermore, this algorithm performs at most
\[ O(N(nkN(M \log(d)) \log \log(d) + h)) \]
b宸告p operations, where \( N \) denotes the maximum number of nonzero terms in each \( F_i \),
\( d \) denotes an upper bound for their total degree, and \( h = \max_{1 \leq i \leq k} (h(F_i)) \).

**Remark.** We have
\[ M(\ell) = O(\ell \log(\ell) \log(\ell)) \]
because of the algorithm of A. Schönhage and V. Strassen ([GG03]) and so the complexity of the algorithm underlying Theorem [LJ] is
\[ O_{\varepsilon} \left( N^{nkN} \log(d)^{1+\varepsilon} + h \right) \quad \forall \varepsilon > 0. \]

Some related problems have been studied in the univariate case. Let \( F \in \mathbb{Z}[X] \) given by its lacunary encoding and let \( m \) be a nonnegative integer. The problem of testing whether \( F \) vanishes at a root of unity of order \( m \) is called the *cyclotomic test* (CT). In [CTV09], Q. Cheng, S.P. Tarasov and M.N. Vyalyi have shown that this problem can be solved in time polynomial in the size of the input; that is, CT \( \in \mathbb{P} \).

The *generalized cyclotomic test* (GCT) consists of determining if \( F \) vanishes in some root of unity at all, that is, if there exists some \( m \in \mathbb{N} \) such that \( (F, m) \) is a positive instance of CT. The fact that CT \( \in \mathbb{P} \) implies that GCT \( \in \mathbb{NP} \). On the other hand, D. Plaisted has previously shown that this problem is NP-hard [PS4] Thm 5.1 and so GCT is NP-complete.

Filaseta and Schinzel proposed a subexponential algorithm for solving GCT [FS94]. In collaboration with Granville, they extend this algorithm to compute
all of the cyclotomic factors of $F$. In this text, we simplify this last algorithm and we clarify the dependence of its complexity in the number $N$ of terms, which turns out to be exponential. We also improve the practical implementation and time execution of this algorithm. The exponential behaviour of the complexity with respect to $N$ seems unavoidable and, indeed, we think that the size of the output is exponential in the size of the input in the worst case. In Section 2, we exhibit a family of examples which supports this conjecture.

For $n \geq 2$, the only previous constructive methods are described in [Ru93], [BS02], [AS08], [Ro07]. Their complexity is not detailed in these references but certainly, it is at least of type $d^2n^h$, where $n$ denotes their number of variables, $d$ denotes a bound for the degree of the input polynomials, and $h$ denotes a bound for their height, because of the systematic application of iterated resultants. Thus these methods are better suited for dense polynomials of small degree and having a small number of variables. On the contrary, our result is particularly efficient for polynomials with few terms but potentially very high degree ($10^{100}$ for example); see for instance the examples in Subsection 2.3.

We deduce from the analysis of the algorithm underlying Theorem 1.1 the following upper bound for the number of torsion cosets in a subvariety of $G_m^n$:

**Corollary 1.2.** Let $F_1, \ldots, F_k \in \mathbb{Z}[X_1, \ldots, X_n]$ be polynomials, each having at most $N$ nonzero terms and let $V$ be the subvariety of $G_m^n$ defined by those polynomials. Then $V_{\text{tors}}$ is the union of at most

\[
(N!)^k \exp \left( 3(n+1) \sqrt{kN \log kN} \right)
\]

torsion cosets.

Historically, the first effective upper bound for the number of maximal torsion cosets has been obtained by E. Bombieri and U. Zannier in [BZ95]. Shortly afterwards, W. Schmidt found an upper bound depending only on the number of variables and on the degree of the input polynomials [S96]. The main interest of the bound in Corollary 1.2 is that it is independent of the degree of the polynomials defining $V$. However, we mention that a similar bound might be alternatively obtained with the methods in [S96].

It is worth noting that the number of connected (or irreducible) torsion cosets does necessarily depend on the degree. Currently, the best bound is due to F. Beukers and C.J. Smyth [BS02] when $n = 2$ and to F. Amoroso and E. Viada in the general case [AV09]. The result of Amoroso and Viada says that the number of connected torsion cosets in a variety defined by polynomials of degree at most $d$ is bounded by:

\[
d^n (200n^5 \log(n^2d))^{n^2(n-1)^2}.
\]

Other results in this direction were obtained by G. Rémont in [Re02], Amoroso and S. David in [AD06] and David and P. Philippon in [DP07].

The outline of the paper is as follows. In Section 2, we present our simplification of the algorithm of Filaseta, Granville and Schinzel. In Section 3, we generalize it for a multivariate polynomial and we will finally explain in Section 4 how we can adapt it for general varieties.
2. Cyclotomic factors of univariate polynomials

Let \( F(X) \in \mathbb{Z}[X] \) be a polynomial of degree \( d \). Computing the set of \( \zeta \in \mu_\infty \) such that \( F(\zeta) = 0 \) is equivalent to computing the set of integers \( m \) for which

\[
\Phi_m(X) | F(X),
\]

where \( \Phi_m \) denotes the \( m \)th cyclotomic polynomial and we would like to compute this set in time polynomial in \( \log(d) \). However, the total number of cyclotomic factors of \( F \) is not polynomial in \( \log(d) \) in the worst case, as shown by the following example. Let \( x \) be an integer greater than 17 and let

\[
F(X) = X^d - 1,
\]

where \( d \) is the product \( p_1 \cdots p_r \) of all prime numbers smaller than \( x \). An effective version of Tchebychev’s Theorem \([RS62]\) gives the inequalities:

\[
\pi(x) \geq \frac{x}{\log x} \quad \text{and} \quad \sum_{p \leq x} \log p \leq 1.02 x \quad \text{for } x \geq 17,
\]

where \( \pi(x) := \# \{ p \text{ prime } : p \leq x \} \). From these, we deduce:

\[
r \geq \frac{x}{\log x} \quad \text{and} \quad d \leq \exp(1.02 x).
\]

Furthermore,

\[
F(X) = \prod_{m|d} \Phi_m(X) \quad \text{and} \quad \# \{ m \in \mathbb{N} : m|d \} = 2^r \geq 2^{\pi(x)} \geq 2^{O\left( \frac{\log(d)}{\log \log(d)} \right)},
\]

so the number of cyclotomic factors of \( F \) is not polynomial in the logarithm of its degree.

Even in the case when the number of factors is small, separating them is as difficult as factorizing integers. To illustrate this problem, let us consider

\[
F(X) = X^{pq} - 1,
\]

where \( p \) and \( q \) are distinct prime numbers. We have

\[
F(X) = \Phi_1(X) \Phi_p(X) \Phi_q(X) \Phi_{pq}(X)
\]

and finding this factorization is equivalent to factor \( pq \). But this problem is known to be hard and, at the moment, no known algorithm can factor \( pq \) in polynomial time in \( \log(pq) \).

To avoid these problems, we will represent the output of our algorithm differently. It will be given as a set \( S_F \) of pairs of integers \( (m, e) \) such that

\[
V(F)_{\text{tors}} = \bigcup_{(m,e) \in S_F} V(\Phi_m(X^e)),
\]

where \( V(F) := \{ x \in \mathbb{Q} : F(x) = 0 \} \). Giving such a representation reduces the problem of computing the complete list of cyclotomic factors of \( F \) to integer factorizations with the following lemma:

Lemma 2.1. Let \( m, e \in \mathbb{N}^* \). We set \( e_1 := \prod_{p|m} p^{\text{ord}_p(e)} \) and \( e_2 := e/e_1 \). Then

\[
\Phi_m(X^e) = \Phi_{me_1}(X^{e_2}) = \prod_{d|e_2} \Phi_{me_1,d}(X).
\]
Proof. To prove the first equality, it suffices to remark that the roots of $\Phi_{me_1}(X^{e_2})$ are also roots of $\Phi_m(X^e)$. Since those polynomials are squarefree, have the same degree and leading coefficient 1, we conclude that they coincide. A similar argument shows that $\Phi_m(X^e)$ divides the polynomial $\prod_{d|e_2} \Phi_{me_1,d}(X)$. Moreover, these polynomials have leading coefficient 1 and we can show that they have the same degree by using the formula
\[ \sum_{d|e_2} \varphi(d) = e_2, \]
where $\varphi$ denotes the Euler totient function. Finally, these three polynomials are equal. \hfill \square

We point out once again that our algorithm returns pairs of integers $(m,e)$ such that $\Phi_m(X^e)|F(X)$ without trying to list the corresponding cyclotomic factors in order to keep a polynomial complexity in the logarithm of the degree of $F$. Once we have identified a factor of a polynomial, a natural problem that arises is to compute its multiplicity. A disadvantage of our representation is that it gives no information about multiplicity. However, a result of G. Hajós [Ha53] asserts that if a polynomial $F \in \mathbb{C}[X]$ has $N$ nonzero terms, then the multiplicity of its factor is at most $N$. This bound suggests the following strategy to compute the multiplicity of the cyclotomic factors of a polynomial $F$ with integers coefficients: apply our algorithm to the polynomial $F$ and to its $N-1$ first derivatives and compute in each case the complete list of cyclotomic factors from the output. Finally, it only remains to compare those different lists of factors. Nevertheless, as we have already seen, computing the complete list of cyclotomic factors from the output of our algorithm cannot be done in polynomial time in the logarithm of the degree of $F$.

As we were interested in the generalization of this algorithm in the multivariate case, we neglected as a first step this question of multiplicity. However, it would be interesting to understand how we can improve our algorithm to obtain more information about the multiplicity with the same running time.

In the rest of this section, we establish the following result:

**Theorem 2.2.** There exists a deterministic algorithm which has the following property: for $F \in \mathbb{Z}[X]$ given by its lacunary encoding, the algorithm determines a set
\[ S_F := \{(m_1, e_1), \ldots, (m_t, e_t)\} \]
of couples of integers such that
\[ V(F)_{\text{tors}} = \bigcup_{i=1}^t V(\Phi_{m_i}(X^{e_i})). \]
Furthermore, this algorithm requires
\[ O\left(N^N (M(\log(d)) \log \log(d) + h)\right) \]
bit operations, where $d$ denotes the degree of $F$, $h$ its height and $N$ its number of nonzero terms.

The algorithm underlying Theorem 2.2 is Algorithm 1 in Subsection 2.2. This result is essentially [FGS08, Thm C], without regard to multiplicities of the cyclotomic factors, but we give a much simpler proof which allows us to make explicit the dependence of the complexity on the number of nonzero terms $N$ and to speed up its practical execution. More importantly, this proof extends to the multivariate
case, as we will see in the next section. We first give a few preliminary results before proving Theorem 2.2 and finally we will construct a family of polynomials having many “separated” cyclotomic factors in Subsection 2.3.

2.1. Preliminary results. We first recall the cost of standard arithmetic in \( \mathbb{Z} \):

The product of two integers of bit length bounded by \( \ell \) can be computed in

\[
M(\ell) = O(\ell \log(\ell) \log \log(\ell)) = O(\ell^{1+\varepsilon}) \quad \forall \varepsilon > 0
\]

bit operations with the algorithm of Schönhage and Strassen [SS71].

The gcd of two integers of bit length bounded by \( \ell \) can be computed in

\[
O((\ell \log(\ell))^2 \log \log(\ell)) = O(\ell^{1+\varepsilon}) \quad \forall \varepsilon > 0
\]

bit operations with the algorithm of Knuth and Schönhage [K70]. We refer to [GG03] for the description and analysis of those algorithms. If \( \zeta \in \mu_\infty \), we denote by \( \text{ord}(\zeta) \) the order of \( \zeta \) in the group \( \mu_\infty \), which equals the least common multiple of the order of its coordinates. For \( m \in \mathbb{N}^* \), we also define

\[
\Psi(m) := 2 + \sum_{p|m}(p - 2),
\]

where \( p \) runs over the prime numbers dividing \( m \). The main ingredient of the proof of Theorem 2.2 is the following result due to Conway and Jones [CJ76]:

**Theorem 2.3** (Conway-Jones 1976). Let \( \zeta_m \) be a root of unity of order \( m \). Also, let \( a_1, \ldots, a_N, \alpha_1, \ldots, \alpha_N \) be integers and \( S := a_1 \zeta_m^{\alpha_1} + \cdots + a_N \zeta_m^{\alpha_N} \). If all of the following hold, then \( m \) is squarefree and \( \Psi(m) \leq N \):

1. \( S = 0 \),
2. no proper subsum of \( S \) vanishes,
3. \( \gcd(\alpha_2 - \alpha_1, \ldots, \alpha_N - \alpha_1, m) = 1 \).

We say that a vanishing sum of roots of unity is *minimal* if condition (2) is satisfied. Condition (3) is equivalent to:

\[
\text{ord}(\zeta_m^{\alpha_2 - \alpha_1}, \ldots, \zeta_m^{\alpha_N - \alpha_1}) = m.
\]

We also need the following lemma:

**Lemma 2.4.** Let \( m_1, \ldots, m_s \in \mathbb{N}^* \), then

\[
\Psi(m_1 \cdots m_s) \leq \sum_{i=1}^s \Psi(m_i) - 2(s - 1).
\]

**Proof.** We have that

\[
\Psi(m_1 \cdots m_s) = 2 + \sum_{p|m_1 \cdots m_s}(p - 2)
\]

\[
\leq 2 + \sum_{i=1}^s \sum_{p|m_i}(p - 2) \leq \sum_{i=1}^s \Psi(m_i) - 2(s - 1). \quad \Box
\]

We also need some upper bounds for the cardinality of certain sets to compute a bound for the complexity of Algorithm 1:

**Lemma 2.5.** Let \( N \in \mathbb{N}^* \) and \( E_N \) be the set of partitions of \( \{1, \ldots, N\} \) which only contain subsets with at least two elements. Then:

\[
\#E_N \leq N!.
\]
Proof. Let $\sigma$ be an element of $\mathfrak{S}_N$, the group of permutations of the set $\{1, \ldots, N\}$. Let $\sigma = c_1 \circ \cdots \circ c_s$ be its decomposition into disjoint cycles, then we associate to $\sigma$ the partition $\{J_1, \ldots, J_s\}$ of the set $\{1, \ldots, N\}$ such that $J_j$ is the support of $c_j$ for all $1 \leq j \leq s$. Since this application is a surjection between $\mathfrak{S}_N$ and the set of partitions of the set $\{1, \ldots, N\}$ and since $E$ is strictly included in this set, we have $\#E \leq N!$. \hfill \Box

Remark. We can improve this estimate to $\#E \leq \frac{N!}{e}$ by considering the subset of $\mathfrak{S}_N$ consisting of the permutations without fixed points, whose image also contains $E_N$.

Lemma 2.6. Let $N, n \in \mathbb{N}$. We set

$$Q_N := \{m \in \mathbb{N} : \Psi(m) \leq N \text{ and } m \text{ is squarefree}\}$$

and

$$Q_{n,N} := \{\omega \in \mathbb{N}_\infty^\mathbb{N} : \text{ord}(\omega) \in Q_N\}.$$  

Then

$$\#Q_N \leq \exp \left(3 \sqrt{N \log N}\right)$$

and

$$\#Q_{n,N} \leq \exp \left(3(n+1) \sqrt{N \log N}\right).$$

Proof. We first prove the first inequality. Let $m \in Q_N$ and let $m = \prod_{i=1}^r p_i$ its factorization into primes. Since $\Psi(m) \leq N$, we have

$$\sum_{i=1}^r p_i \leq N + 2(r - 1).$$

Furthermore,

$$N - 2 \geq \sum_{i=1}^r (p_i - 2) \geq \sum_{i=2}^r (p_i - 2) \geq \sum_{i=1}^{r-1} (2i - 1) = (r - 1)^2.$$  

So, $r \leq \sqrt{N - 2} + 1$ and

$$\sum_{i=1}^r p_i \leq N + 2\sqrt{N - 2}.$$  

Let us now obtain a better upper bound for $r$. By direct computation, we obtain that

$$r \leq 3 \sqrt{\frac{N}{\log N}} \quad \text{for } N \leq 10,000.$$  

We assume now that $N \geq 10,000$. An effective version of Tchebychev’s Theorem (see \cite{RS62}) gives the inequalities:

$$\pi(x) \leq 1.26 \frac{x}{\log x} \quad \text{and} \quad \sum_{p \leq x} p \geq \frac{x^2}{2\log x} \quad \text{for } x \geq 347.$$  

Then for $x := 1.15 \sqrt{N \log N}$, we have $x \geq 347$ and thus:

$$\sum_{p \leq x} p \geq \frac{1.15^2 N \log N}{2\log(1.15 \sqrt{N \log N})} \geq N + 2\sqrt{N - 2}. $$
The last inequality comes from a simple study of functions. Combining the two inequalities \( (4) \) and \( (6) \), we obtain
\[
\sum_{p \leq x} p \geq \sum_{i=1}^{r} p_i
\]
and then:
\[
r \leq \pi(1.15 \sqrt{N \log N}) \leq 1.26 \frac{1.15 \sqrt{N \log N}}{\log(1.15 \sqrt{N \log N})}
\]
according to \( (5) \), and finally,
\[
r \leq 1.26 \frac{1.15 \sqrt{N \log N}}{\log(\sqrt{N})} \leq 3\sqrt{N/\log N}.
\]
So each integer in the set \( Q_N \) has at most \( 3\sqrt{N/\log N} \) prime factors which are all smaller than \( N \). It follows that
\[
\#Q_N \leq N^3 \sqrt{N/\log N} = \exp\left(3\sqrt{N/\log N}\right).
\]
Now we prove the second inequality. Each \( \omega \in Q_{n,N} \) of order \( m \in Q_N \) can be represented by the \( (n+1) \)-uplet,
\[(d_1, \ldots, d_n, m),\]
where the integers \( d_i \) are between 0 and \( m-1 \) so that
\[
\omega = (\zeta^{d_1}_m, \ldots, \zeta^{d_n}_m),
\]
where \( \zeta_m = \exp\left(\frac{2i\pi}{m}\right) \). We have established that \( m \leq \exp\left(3\sqrt{N \log N}\right) \) and so there are at most
\[
\exp\left(3(n+1)\sqrt{N \log N}\right)
\]
such \( (n+1) \)-uplets and the result follows. \( \square \)

2.2. **Proof of Theorem 2.2** In the following, let
\[
F(X) = \sum_{i=1}^{N} a_i X^{\alpha_i} \in \mathbb{Z}[X]
\]
and \( \zeta_m \) be a root of unity of order \( m \) such that
\[
F(\zeta_m) = 0. \tag{7}
\]
We will determine equivalent conditions to equation \( (7) \) that we could test algorithmically. Let \( \{J_1, \ldots, J_s\} \) be a partition of the set \( \{1, \ldots, N\} \) such that
\[
\sum_{i \in J_j} a_i \zeta_m^{\alpha_i} = 0 \quad \forall \ 1 \leq j \leq s
\]
and such that every such sum is minimal. For \( 1 \leq j \leq s \), let
\[
F_j(X) := \sum_{i \in J_j} a_i X^{\alpha_i},
\]
such that \( F(X) = \sum_{j=1}^{s} F_j(X) \). We also define for \( 1 \leq j \leq s \), integers \( e_j, b_j \) and polynomials \( G_j \) such that
\[
F_j(X) = X^{b_j} G_j(X^{e_j}), \tag{8}
\]
where \( G_j(0) \neq 0 \) and such that the exponents of the monomials appearing in \( G_j \) are co-prime. We finally set:
\[
m_j := \frac{m}{\gcd(m, e_j)}.
\]
If \( \zeta_{m_j} \) denotes a root of unity of order \( m_j \), the equation (7) is then equivalent to:

\[
(9) \quad G_j(\zeta_{m_j}) = 0 \quad \text{and} \quad m_j = \frac{m}{\gcd(m,e_j)} \quad \forall \ 1 \leq j \leq s.
\]

Furthermore, by construction, the sums \( G_j(\zeta_{m_j}) \) satisfy the conditions of Theorem 2.3.

So we have

\[
\Psi(m_j) \leq N_j \quad \forall \ 1 \leq j \leq s,
\]

where \( N_j := \#J_j \) and \( m_j \) is squarefree.

We set \( e := \gcd(e_1, \ldots, e_s) \) and, for \( 1 \leq j \leq s \), we also set \( e'_j := e_j/e \). Finally, we set

\[
m' := \frac{m}{\gcd(m,e)}
\]

and we observe that

\[
\frac{m}{\gcd(m,e_j)} = \frac{m'}{\gcd(m',e'_j)} \quad \forall \ 1 \leq j \leq s.
\]

The condition (9) is then equivalent to:

\[
(10) \quad G_j(\zeta_{m_j}) = 0 \quad \text{and} \quad m_j = \frac{m'}{\gcd(m',e'_j)} \quad \forall \ 1 \leq j \leq s.
\]

Let us prove that

\[
m' = \lcm(m_1, \ldots, m_s).
\]

From the condition (10), we have \( m_j | m' \) for all \( 1 \leq j \leq s \), so

\[
\lcm(m_1, \ldots, m_s) | m'.
\]

Let us prove the reverse divisibility relation. Suppose \( p \) is a prime and \( u \) a positive integer for which \( p^u | m \). Since \( m_j \) is squarefree for each \( j \), the definition of \( m_j \) implies that \( p^{u-1} | e_j \) for each \( j \). Hence, \( p^{u-1} | e \). The definition of \( m' \) now implies that \( m' \) is squarefree. Therefore, it suffices to show that if \( p \) is a prime divisor of \( m' \), then \( p \) divides \( m_j \) for some \( j \). Assume \( p \) is a prime divisor of \( m' \) which does not divide \( m_j \) for every \( j \). Then (10) implies that \( p \) divides \( \gcd(m',e'_j) \) for each \( j \). Hence, \( p | e'_j \) for each \( j \), contradicting that \( \gcd(e'_1, \ldots, e'_s) = 1 \).

We can now bound \( \Psi(m') \) as follows:

\[
\Psi(m') = \Psi(\lcm(m_1, \ldots, m_s)) \leq \Psi(m_1 \cdots m_s) \leq \sum_{j=1}^{s} \Psi(m_j) - 2(s-1) \quad \text{according to Lemma 2.4}
\leq \sum_{j=1}^{s} N_j - 2(s-1) \leq N - 2(s-1).
\]

We also remark that after a partition of \( \{1, \ldots, N\} \) and a squarefree integer \( m' \) are fixed, the condition (10) can be tested since the polynomials \( G_j \) and the integers \( m_j, e'_j \) only depend on the partition and on \( m' \). The algorithm will test the condition (10) for all possible choices of partitions and for all possible choices of integers \( m' \).

We can do this as follows:
Algorithm 1

Input: A polynomial $F(X) = \sum_{i=1}^{N} a_i X^{\alpha_i}$ given by its lacunary encoding.
Output: A set $S_F$ satisfying the conditions of Theorem 2.2

1. $S_F \leftarrow \emptyset$.
2. For each partition $\{J_1, \ldots, J_s\}$ of $\{1, \ldots, N\}$ such that $\#J_j \geq 2$, for all $1 \leq j \leq s$ do:
   (a) For all $1 \leq j \leq s$, compute the polynomials $G_j$ and the integers $e_j$ associated to the set $J_j$, as in (8).
   (b) Compute $e := \gcd(e_1, \ldots, e_s)$ and for $1 \leq j \leq s$, compute $e_j' := e_j/e$.
   (c) For each squarefree integer $m'$ satisfying $\Psi(m') \leq N - 2(s-1)$ do:
      (i) For $1 \leq j \leq s$, compute $m_j := \frac{m'}{\gcd(m', e_j')}$.
      (ii) If $\Phi_{m_j}(X)|G_j(X)$ for all $1 \leq j \leq s$, then do $S_F \leftarrow S_F \cup \{(m', e)\}$.
3. Output $S_F$ and terminate.

Let us prove that the set $S_F$ output by the algorithm satisfies the conclusion of Theorem 2.2. Suppose that $(m', e)$ is an element of the set $S_F$. From the discussion above Algorithm 1, we have $F(\zeta) = 0$ for every root of unity $\zeta$ such that $\zeta^{m'}$ is of order $m'$. From this, we can conclude that the polynomial $\Phi_{m'}(X^e)$ divides $F(X)$. Thus we have

$$V(F)_{\text{tors}} \supseteq \bigcup_{(m', e) \in S_F} V(\Phi_{m'}(X^e)).$$

Let us prove the reverse inclusion. Let $m \in \mathbb{N}^*$ and $\zeta_m$ be a root of unity of order $m$ such that $F(\zeta_m) = 0$. From the discussion above Algorithm 1, there exists a partition of the support of $F$ and a squarefree integer $m'$ such that $m' = \frac{m}{\gcd(m, e)}$, where the integer $e$ is the integer computed at step (2.b). Furthermore, all the conditions (10) hold. Assume that this particular partition is chosen at step 2. and that the integer $m'$ is chosen at step (2.c), then the conditions of step (2.c.i) are satisfied and the couple $(m', e)$ belongs to the set $S_F$. From the equality $m' = \frac{m}{\gcd(m, e)}$, we conclude that $\Phi_{m'}(\zeta_m^e) = 0$ and we have:

$$V(F)_{\text{tors}} \subseteq \bigcup_{(m', e) \in S_F} V(\Phi_{m'}(X^e)).$$

This completes the correctness of Algorithm 1.

We explain how we compute the set $Q_N$ of integers considered in step (2.c). We recall that the set $Q_N$ was defined in Lemma 2.6 by

$$Q_N = \{m \in \mathbb{N} : m \text{ is squarefree and } \Psi(m) \leq N\}.$$

We begin by constructing the set $P_N = \{p_1, \ldots, p_t\}$ of prime numbers less than $N$, which can be done with the Eratosthenes sieve in $O(N^2 \log N)$ bit operations.
Then we compute the subsets \( \{q_1, \ldots, q_r\} \) of \( P_N \) for which

\[
2 + \sum_{i=1}^{r} (q_i - 2) \leq N.
\]

Checking the inequality \((11)\) requires to sum up at most \( N \) integers of bit length less than \( \log N \), which can be done in \( O(N \log N) \) bit operations for a given subset \( \{q_1, \ldots, q_r\} \) of \( P_N \). Furthermore, \( \#P_N < N \), so this set has less than \( 2^N \) subsets and checking the inequality \((11)\) for all of them requires \( O(2^N N \log N) \) bit operations.

Finally, for each subset \( \{q_1, \ldots, q_r\} \) of \( P_N \) satisfying the inequality \((11)\), we compute the product \( \prod_{i=1}^{r} q_i \). The set of all these products gives \( Q_N \). Computing each of these products requires at most \( N \) products of integers of bit length less than \( \log N \) and since this product never exceeds \( \exp(3\sqrt{N \log N}) \) according to Lemma \((2.3)\), this can be done in \( O(N^2 \log N) \) bit operations. Performing this operation for all of the \( \{q_1, \ldots, q_r\} \) can be done in \( O(2^N N \log N) \) bit operations and this is also the number of bit operations needed to compute \( Q_N \).

Let us now explain how we check the divisibility relation \( \Phi_{m_j}(X) | G_j(X) \) at step \((2.c.i)\). To this end, we use a simple algorithm described by Filaseta and Schinzel in [FS04, Theorem 3]. It requires the factorization of \( m_j \) and is based on the equivalence

\[
\Phi_{m_j}(X) | G_j(X) \iff X^{m_j} - 1 \mid G(X) \times \prod_{p|m_j} (X^m/p - 1),
\]

where the product runs over the prime numbers dividing \( m \). We refer to [FS04] for the complete description of this algorithm and its running time. In our case, since \( m_j \) is squarefree, the first step of their algorithm can be avoided and the divisibility relation \( \Phi_{m_j}(X) | G_j(X) \) can be checked in \( O(N^3 (\log N)^2 2^{3\sqrt{N \log N}} (h + M (\log(d)))) \) bit operations.

Finally, we estimate the overall complexity of Algorithm 1. We recall that at step \((2.a)\), the polynomials \( G_j \) are defined by

\[
F_j(X) = X^{b_j} G_j(X^{e_j}).
\]

In order to compute these polynomials, we have to perform \( N \) subtractions of integers of bit length bounded by \( \log(d) \), then we have to perform at most \( N \) gcd computations of integers of bit length bounded by \( \log(d) \), and finally compute \( N \) divisions of integers of bit length bounded by \( \log(d) \), and all of these computations require \( O(N M (\log(d)) \log \log(d)) \) bit operations.

At step \((2.b)\), we have to compute \( s - 1 \) gcd of integers of bit length bounded by \( \log(d) \) to compute \( e \), then we have to perform \( s \) Euclidean divisions of integers of bit length bounded by \( \log(d) \) to compute the integers \( e_j' \). So in the whole, this requires \( O(N M (\log(d)) \log \log(d)) \) bit operations.

At step \((2.c)\), in order to compute the integers \( m_j \), we perform \( s \) divisions and \( s \) gcd computations of integers of bit length bounded by \( d \), and this requires again \( O(N M (\log(d)) \log \log(d)) \) bit operations. In order to check the divisibility relations, we apply \( s \) times the algorithm of Filaseta and Schinzel [FS04, Theorem 3] and this can be done in

\[
O \left( N^4 (\log N)^2 2^{3\sqrt{N \log N}} (h + M (\log(d))) \right)
\]

bit operations.
Moreover, since there are at most \( N! \) partitions to consider at step (2) by Lemma 2.5 and since there are at most 
\[
\exp \left( 3\sqrt{N\log N} \right)
\]
integers to consider at step (2.c) by Lemma 2.6, the total number of bit operations performed by the algorithm is:
\[
O \left( N^N (M(\log(d)) \log \log(d) + h) \right);
\]
this terminates the proof of Theorem 2.2. \( \square \)

2.3. A family of polynomials having many “separated” cyclotomic factors. We don’t know if the size of the output of Algorithm 1 is always polynomial in the size of the input. When \( N = 2n \) is an even integer, we construct polynomials for which the output of the algorithm cannot be regrouped in less than \( n! \) pairs of integers, but unfortunately the degree of those polynomials is very large.

**Proposition 2.7.** Let \( n \in \mathbb{N} \) and \( p_1, \ldots, p_n \) be \( n! \) prime numbers greater than \( 2n \). Then there exists a polynomial \( f \in \mathbb{Z}[X] \) having \( 2n \) nonzero terms and satisfying

1. \( X^{p_r} - 1 \mid f(X) \quad \forall 1 \leq r \leq n! \),
2. \( X^{p_r p_{r'}} - 1 \mid f(X) \quad \forall 1 \leq r < r' \leq n! \).

**Proof.** Let \( f(X) := \sum_{i=1}^n X^{\alpha_i} - \sum_{i=n+1}^{2n} X^{\alpha_i} \). We will compute values of \( \alpha_i \) for which \( f \) satisfies Proposition 2.7. We consider all the partitions \( \{J_1, \ldots, J_n\} \) of the set \( \{1, \ldots, 2n\} \) made by \( 2 \)-element sets \( J_1, \ldots, J_n \) such that each \( J_j \) contains an element in \( \{1, \ldots, n\} \) and another one in \( \{n+1, \ldots, 2n\} \). For instance,

\[
\{\{1, n + 1\}, \{2, n + 2\}, \ldots, \{n, 2n\}\}
\]
is one of these partitions. We remark that there are \( n! \) such partitions. We enumerate these partitions as

\[
\pi_r := \{J_{r,1}, \ldots, J_{r,n}\} \quad \forall 1 \leq r \leq n!.\]

To each partition \( \pi_r = \{J_{r,1}, \ldots, J_{r,n}\} \) we associate injectively a prime number \( p_r \) strictly larger than \( 2n \). According to the Chinese remainder theorem, we can choose \( \alpha_i \) such that:

\[
i \in J_{r,k} \iff \alpha_i \equiv k - 1 \mod p_r \quad \forall 1 \leq r \leq n!.
\]
By construction, all the \( \alpha_i \) are distinct and we have:

\[
X^{p_r} - 1 \mid f(X) \quad \forall 1 \leq r \leq n!.
\]
We finally prove the second part of the proposition. Let us suppose, for example, that \( X^{p_1 p_2} - 1 \) divides \( f(X) \). We then have \( \Phi_{p_1 p_2}(X) \mid f(X) \). Thus \( f(\zeta) = 0 \), where \( \zeta \) denotes a root of unity of order \( p_1 p_2 \). Also, let

\[
f(\zeta) = \sum_{i=1}^n \zeta^{\alpha_i} - \sum_{i=n+1}^{2n} \zeta^{\alpha_i} =: S_1 + \cdots + S_t,
\]
be a decomposition of \( f(\zeta) \) into minimal vanishing sums. Let \( m_j \) be the order of \( S_j \).

Since \( m_j \mid p_1 p_2 \), we have \( m_j \in \{1, p_1, p_2, p_1 p_2\} \) and \( m_j \) satisfies \( \Psi(m_j) \leq \# S_j \leq 2n \) according to Theorem 2.3. We have that \( \Psi(p_1) = p_1 > 2n \), \( \Psi(p_2) = p_2 > 2n \), \( \Psi(p_1 p_2) = p_1 + p_2 - 2 > 2n \) and so \( m_j = 1 \). So each \( S_j \) is a multiple of a vanishing
sum of roots of unity of the shape $\pm 1 \cdots \pm 1 = 0$. Since $S_j$ is minimal, it must have
length 2 and we also have $t = n$. Thus for $1 \leq j \leq n$, we can rewrite $S_j$ as
\[ S_j = \zeta^{\alpha_j} - \zeta^{\alpha_{j+1}} \]
with
\[ 1 \leq a_j \leq n < b_j \leq 2n \quad \text{and} \quad \alpha_{a_j} = \alpha_{b_j} \mod p_1 p_2. \]
From the condition $1 \leq a_j \leq n < b_j \leq 2n$, for all $1 \leq j \leq n$, we can conclude that
the decomposition of $f(\zeta)$ into minimal vanishing sums corresponds to one of the
partitions $\pi_r$, where $1 \leq r \leq n!$. Furthermore, the condition $\alpha_{a_j} = \alpha_{b_j} \mod p_1 p_2$, for
all $1 \leq j \leq n$, implies that
\[ \alpha_{a_j} \equiv \alpha_{b_j} \mod p_1 \quad \text{and} \quad \alpha_{a_j} \equiv \alpha_{b_j} \mod p_2 \quad \forall 1 \leq j \leq n. \]
Thus we necessarily have $\pi_r = \pi_1$ and $\pi_r = \pi_2$. This is a contradiction since
$\pi_1 \neq \pi_2$. From here we conclude that $X^{p_1 p_2} - 1 \nmid f(X)$.  \hfill $\Box$

Let $N = 2n$ be an even integer and let $f$ be the polynomial constructed in the
proof of Proposition 2.7. It is easy to see that Algorithm 1, when applied to $f$, will
output at least $n!$ pairs of integers. We prove now that the degree of $f$ is doubly
exponential in $N$. So the size of the input is also exponential in $N$.

We first remark that $\deg(f)$ can be bounded by $p_1 p_2 \cdots p_n$ according to the
Chinese remainder theorem, so we can bound $\deg(f)$ by a double exponential in $n$.
Let us now find a lower bound for $\deg(f)$. We have $\deg(f) \geq |\alpha_{n+1} - \alpha_1|$ so we
will bound by above $|\alpha_{n+1} - \alpha_1|$. We consider all the partitions for which 1 and
$n + 1$ are regrouped in the same pair. There are $(n-1)!$ such partitions. We denote
by $\pi_1', \ldots, \pi_{(n-1)}'$ those partitions and by $p'_1, \ldots, p'_{(n-1)}$, the corresponding prime
numbers. Since $\alpha_1$ and $\alpha_{n+1}$ are equal modulo every of these prime numbers, we
have:
\[ \alpha_1 - \alpha_{n+1} \equiv 0 \mod p'_1 p'_2 \cdots p'_{(n-1)}! \]
Furthermore, $\alpha_1 - \alpha_{n+1} \neq 0$ (because they are not equal modulo the other prime
numbers) so we have:
\[ |\alpha_1 - \alpha_{n+1}| \geq p'_1 p'_2 \cdots p'_{(n-1)}! \]
Thus $|\alpha_1 - \alpha_{n+1}|$ is greater than the product of the $(n-1)!$ first integers and we have
\[ |\alpha_1 - \alpha_{n+1}| \geq (n-1)!! \]
Finally, the degree of $f$ is doubly exponential in $n$ so this example does not suffice
to prove that the size of the output may not be polynomial in the size of the input.

**Example.** We used a construction similar to the above to obtain the examples
\[ f_1(X) = 1 + X + X^2 - X^{2501941} - X^{6088721} - X^{6282199} \]
and
\[ f_2(X) = 1 + X + X^2 + X^3 - X^{106493315971896428139150081202897150286932} \\
- X^{10417696267221214855704118228748809801239} \\
- X^{2747750133111287905524860455880456232062} \\
- X^{626914938199634951807585972855218426387}. \]
Using the polynomial $f_1(X)$ in Algorithm 1 leads to the output
\[ \{(2,2), (1,7), (1,11), (1,13), (1,17), (1,19), (1,23)\}; \]
and we cannot regroup any of these pairs to shorten the description of the cyclotomic factors of \( f_1(X) \). Using the polynomial \( f_2(X) \) in Algorithm 1 leads to an output of 24 pairs which again cannot be regrouped to shorten the list of cyclotomic factors.

### 3. Torsion cosets of a hypersurface

In this section, we extend the previous result to the hypersurface case. In particular, we prove Theorem 1.1 for the case where \( V \) is a hypersurface and we describe the algorithm underlying in Subsection 3.2.

#### 3.1. Preliminary results

First, we will need the following extension of Theorem 2.3 to the multivariate case.

**Corollary 3.1.** Let \( \zeta \in \mu_{\infty}^m \) be a point of order \( m \) and \( F(X) = \sum_{i=1}^{N} a_i X^{\alpha_i} \in \mathbb{Z}[X_1^{\pm 1}, \ldots, X_n^{\pm 1}] \) be a Laurent polynomial. If all of the following hold, then \( m \) is squarefree and \( \Psi(m) \leq N \):

1. \( F(\zeta) = 0 \),
2. no proper subsum of \( F(\zeta) \) vanishes,
3. \( \sum_{i=2}^{N} \mathbb{Z}(\alpha_i - \alpha_1) = \mathbb{Z}_n \).

**Proof.** We just have to prove that \( S := \sum_{i=1}^{N} a_i \zeta^{\alpha_i - \alpha_1} \) satisfies the three conditions of Theorem 2.3. The first two conditions are satisfied since \( F(\zeta) \) is a minimal vanishing sum by hypothesis. Let \( M := \text{lcm}_{1 \leq i \leq N} \left( \text{ord}(\zeta^{\alpha_i - \alpha_1}) \right) \). It remains to check the last condition, i.e., to check if \( M = m \). Since the divisibility \( M | m \) is clear, we only have to prove the reverse divisibility condition. In the following, we set \( \zeta = (\zeta_1^d \zeta_m^1, \ldots, \zeta_n^d \zeta_m^N) \), where \( \zeta_m \) is a root of unity of order \( m \). Let us first note that for all \( b_1, \ldots, b_N \in \mathbb{Z} \), we have:

\[
\text{ord} \left( \sum_{i=1}^{N} b_i (\alpha_i - \alpha_1) \right) | M.
\]

Moreover, since \( \sum_{i=1}^{N} \mathbb{Z}(\alpha_i - \alpha_1) = \mathbb{Z}_n \), there exists \( b_1, \ldots, b_N \in \mathbb{Z} \) such that \( \sum_{i=1}^{N} b_i (\alpha_i - \alpha_1) = (1, 0, \ldots, 0) \). Thus,

\[
\text{ord}(\zeta_m^d) = \text{ord} \left( \sum_{i=1}^{N} b_i (\alpha_i - \alpha_1) \right) | M,
\]

and similarly,

\[
\text{ord}(\zeta_m^d) | M \quad \text{for } 2 \leq i \leq N.
\]

Finally, we have

\[
m = \text{lcm}_{1 \leq i \leq N} \left( \text{ord}(\zeta_m^d) \right) | M,
\]

and the last condition of Theorem 2.3 is satisfied. \[ \square \]

We will also need to be able to compute a basis of a subgroup of \( \mathbb{Z}_n^r \) from a system of generators. Let \( \beta_1, \ldots, \beta_r \in \mathbb{Z}_n^r \) be vectors generating a subgroup \( R \) of \( \mathbb{Z}_n^r \) of rank \( k \leq r \) and \( M \) the matrix of size \( r \times n \) whose rows are the vectors \( \beta_1, \ldots, \beta_r \). We can compute from \( M \) a matrix \( H \) of size \( k \times n \), which is upper triangular, whose rows consist of a basis of \( R \). The matrix \( H \) is called the Hermite normal form of \( M \) and can be computed with the following lemma:
Lemma 3.2. There exists an algorithm that computes the Hermite normal form of a matrix $M$ of size $k \times n$ with integer coefficients with absolute value bounded by $d$ in
\[ O \left( k^4 n^2 M \log(d) \log \log(d) \right) \]
bit operations.

The algorithm underlying Lemma 3.2 is described in [LS96]. The authors of this article give a proof of this result with a better complexity but this one will suffice for our need.

3.2. Proof of Theorem 1.1 for a hypersurface. Let $F(X) = \sum_{i=1}^{N} a_i X^{\alpha_i} \in \mathbb{Z}[X_1, \ldots, X_n]$ and $\zeta \in \mu_{\infty}$ such that:
\[ F(\zeta) = 0. \tag{12} \]
As we have done in the proof of Theorem 2.2, we will find some conditions equivalent to (12) that we could test algorithmically.

Let $\{J_1, \ldots, J_s\}$ be a partition of $\{1, \ldots, N\}$ such that
\[ \sum_{i \in J_j} a_i \zeta^{\alpha_i} = 0 \quad \forall 1 \leq j \leq s, \]
and such that each of these sums is minimal. We set $N_j := \#J_j$ and, for $1 \leq j \leq s$,
\[ F_j(X) := \sum_{i \in J_j} a_i X^{\alpha_i}. \]
Thus we have $F(X) = \sum_{j=1}^{s} F_j(X)$. Then we renumber the coefficients and the exponents of $F_j$ in the following manner:
\[ F_j(X) := \sum_{i=1}^{N_j} a_j,i X^{\alpha_j,i}. \]
For $1 \leq j \leq s$ let
\[ R_j := \sum_{i=1}^{N_j} \mathbb{Z}(\alpha_j,i - \alpha_{j,i}) \]
be the subgroup of $\mathbb{Z}^n$ spanned by the differences of exponents of $F_j$. We denote by $k_j$ the rank of this group and by $\{\lambda_{j,1}, \ldots, \lambda_{j,k_j}\}$ a basis of $R_j$. Then we can find a Laurent polynomial $G_j$ in $k_j$ variables and having $N_j$ nonzero terms such that:
\[ F_j(X) = X^{\alpha_{j,1}} G_j(X^{\lambda_{j,1}}, \ldots, X^{\lambda_{j,k_j}}). \tag{13} \]
For $1 \leq t \leq k_j$, we set
\[ \omega_{j,t} := \zeta^{\lambda_{j,t}}. \]
We have
\[ G_j(\omega_{j,1}, \ldots, \omega_{j,k_j}) = 0 \quad \text{for } 1 \leq j \leq s. \tag{14} \]
The Laurent polynomial $G_j$ and the point $(\omega_{j,1}, \ldots, \omega_{j,k_j})$ of $\mu_{\infty}^{k_j}$ satisfy the conditions of Corollary 3.1. If $m_j$ denotes the order in $\mu_{\infty}$ of $(\omega_{j,1}, \ldots, \omega_{j,k_j})$, we conclude that $m_j$ is squarefree and $\Psi(m_j) \leq N_j$. Now let $R := \sum_{j=1}^{s} R_j$ and $\{\lambda_1, \ldots, \lambda_k\}$ be a basis of $R$. For $1 \leq t \leq k$, we set $\omega_t := \zeta^{\lambda_t}$ and $m' := \text{ord}(\omega_1, \ldots, \omega_k)$. Let $m'' := \text{lcm}(m_1, \ldots, m_s)$. We prove that
\[ m'' = m'. \]
The least positive integer \( m \) for which

\[
\omega_{j,1}^m = \cdots = \omega_{j,k_j}^m = 1
\]

is \( m = m_j \). The above identity also holds for \( m = m' \), so \( m_j | m' \), and hence, \( m'' | m' \).

Conversely, let \( t \) be an integer between 1 and \( k \). Since \( \lambda_t = \sum_{j=1}^s R_j \), there exist integers \( a_{j,l}^{(t)} \), where \( 1 \leq j \leq s \) and \( 1 \leq l \leq k_j \), such that

\[
\lambda_t = \sum_{j=1}^l \sum_{l=1}^{k_j} a_{j,l}^{(t)} \lambda_{j,l}. 
\]

Thus, we have

\[
\omega_{m''}^t = \zeta^{m''} \lambda_t
\]

\[
= \prod_{j=1}^s \prod_{l=1}^{k_j} \omega_{m''}^{a_{j,l}^{(t)}} \lambda_{j,l}
\]

\[
= \prod_{j=1}^s \prod_{l=1}^{k_j} \omega_{m''}^{a_{j,l}^{(t)}}
\]

\[
= \prod_{j=1}^s \prod_{l=1}^{k_j} 1
\]

and

\[
m' | m''.
\]

As in the proof of Theorem 2.2, the integer \( m' \) is squarefree and satisfies \( \Psi(m') \leq N - 2(s - 1) \).

If the equations in (14) all hold for some \( \zeta \in \mu_n^\infty \), then \( F(\zeta) = 0 \) for every \( \zeta \in \mu_n^\infty \) satisfying

\[
\omega_t = \zeta^{\lambda_t} \quad \text{for } 1 \leq t \leq k.
\]

Thus we have

\[
F(\zeta) = 0 \quad \forall \zeta \in B(L, (\omega_1, \ldots, \omega_k)),
\]

where \( L \) denotes the matrix of size \( k \times n \) whose rows are the vectors \( \lambda_1, \ldots, \lambda_k \). We recall that \( B(L, (\omega_1, \ldots, \omega_k)) \) was defined in (9) as

\[
B(L, (\omega_1, \ldots, \omega_k)) = \{ \zeta \in \mathbb{G}_m^n \mid \zeta^L = (\omega_1, \ldots, \omega_k) \}.
\]

Let us summarize what we have just proved: if \( F(\zeta) = 0 \) for some \( \zeta \in \mu_n^\infty \), then there exist \( L \in M_{n,k}(\mathbb{Z}) \) and \( \omega \in \mu_k^\infty \), where \( k \leq n \), such that

\[
\zeta \in B(L, \omega) \quad \text{and} \quad B(L, \omega) \subset V(F).
\]

Furthermore, the matrix \( L \) corresponds to a partition of the support of \( F \) and can be computed from it. On the other hand,

\[
\omega \in Q_{k,N} = \{ \zeta \in \mu_k^\infty : \Psi(\text{ord}(\zeta)) \leq N \},
\]

which has cardinality bounded by \( \exp(3(k + 1)\sqrt{N \log N}) \) by Lemma 2.6. Thus the algorithm will test the equalities (14) for all possible partitions of the support of \( F \) and for all possible choices of points in \( Q_{k,N} \). We can do this as follows:
Algorithm 2

Input: A polynomial \( F(X) = \sum_{i=1}^{N} a_i X^{\alpha_i} \) given by its lacunary encoding.
Output: A list \( S_F \) of torsion cosets satisfying Theorem 1.1

1. \( S_F \leftarrow \emptyset \).
2. For each partition \( \{ J_1, \ldots, J_s \} \) of \( \{ 1, \ldots, N \} \) such that \( \#J_j \geq 2 \) for all \( 1 \leq j \leq s \), do:
   (a) Compute for \( 1 \leq j \leq s \), a basis \( \{ \lambda_{j,1}, \ldots, \lambda_{j,k_j} \} \) of \( R_j \) and a Laurent polynomial \( G_j \) satisfying the condition (13).
   (b) Compute a basis \( \{ \lambda_1, \ldots, \lambda_k \} \) of \( R \) and compute, for \( 1 \leq j \leq s \) and for \( 1 \leq h \leq k_j \) the integers \( \delta_{j,h,t} \) such that \( \lambda_{j,h} = \sum_{t=1}^{k} \delta_{j,h,t} \lambda_t \).
   (c) For each squarefree integer \( m' \) such that \( \Psi(m') \leq N - 2(s - 1) \) and for each \( (\omega_1, \ldots, \omega_k) \in \mu_{\infty}^k \) of order \( m' \) do:
      (i) Compute for \( 1 \leq j \leq s \) and for \( 1 \leq h \leq k_j \), \( \omega_{j,h} := \prod_{t=1}^{k} \omega_t^{\delta_{j,h,t}} \) and set \( m_j := \text{ord}(\omega_{j,1}, \ldots, \omega_{j,k_j}) \).
      (ii) If for all \( 1 \leq j \leq s \), we have \( G_j(\omega_{j,1}, \ldots, \omega_{j,k_j}) = 0 \), then do \( S_F \leftarrow S_F \cup \{ (L, (\omega_1, \ldots, \omega_k)) \} \), where \( L \) is the matrix of size \( k \times n \) whose rows are the vectors \( \lambda_1, \ldots, \lambda_k \).
3. Output \( S_F \) and terminate.

The previous discussion guarantees the correctness of Algorithm 2 and it only remains to evaluate its running time.

At step (2.a) we first have to compute, for \( 1 \leq j \leq s \), the differences \( \alpha_{j,j} - \alpha_{j,1} \). This requires \( O(Nn \log(d)) \) bit operations. Then we have to compute a basis \( \{ \lambda_{j,1}, \ldots, \lambda_{j,k_j} \} \) of the subgroup \( R_j \) of \( \mathbb{Z}^n \). This can be done by computing the Hermite normal form of the matrix whose rows consist of the vectors \( \alpha_{j,1} - \alpha_{j,1} \).

This requires \( O(N^4n^2M(\log(d)) \log(d)) \) bit operations by Lemma 3.2. In order to compute the Laurent polynomials \( G_j \), we then have to compute the coordinates of the vectors \( \alpha_{j,1} - \alpha_{j,1} \) in the basis \( \{ \lambda_{j,1}, \ldots, \lambda_{j,k_j} \} \). Thus we have to invert a linear system of size \( n \times n \) with coefficients bounded by \( 2d \). This can be done in \( O(n^3M(\log(d)) \log(d)) \) bit operations. Finally, we can bound the total number of bit operations performed at this step by
\[
O \left( N^5n^3M(\log(d)) \log(d) \right).
\]

At step (2.b), we compute the Hermite normal form of the matrix whose rows are the vectors \( \lambda_{j,h} \), where \( 1 \leq j \leq s \) and \( 1 \leq h \leq k_j \). We obtain a basis \( \{ \lambda_1, \ldots, \lambda_k \} \) of \( R \). Finally, we compute the coordinates of the vectors \( \lambda_{j,h} \) in this basis. This step again requires
\[
O \left( N^5n^3M(\log(d)) \log(d) \right)
\]
bit operations.

At step (2.c), each number \( \omega_i \) is represented by the pair of integers \( (\rho_i, m') \) such that
\[
\omega_i = \exp \left( \frac{2i\rho_i \pi}{m'} \right).
\]

At step (2.c.i), each number \( \omega_{j,h} \) is represented by the pair of integers
\[
(d_{j,h}, m') := \left( \sum_{t=1}^{k} \delta_{j,h,t}\rho_i \mod m', m' \right).
\]
such that
\[ \omega_{j,h} = \exp \left( \frac{2id_{j,h} \pi}{m'} \right). \]

This computation requires
\[ O \left( nM (\log(d)) M \left( \exp \left( 3\sqrt{N \log N} \right) \right) \right) = O \left( nN^2 M (\log(d)) \right) \]
bit operations, since \( m' \leq \exp \left( 3\sqrt{N \log N} \right) \) according to Lemma 2.6. Then we can compute for \( 1 \leq j \leq s \), the integer
\[ m_j := \text{lcm} \left( \frac{m'}{\gcd(m', d_{j,h})} \right). \]

These computations require \( O(nN^2) \) bit operations. Finally, this step requires
\[ O(nN^2 M (\log(d))) \]
bit operations.

At step (2.c.ii), we have to test, for \( 1 \leq j \leq s \) the equality
\[ G_j(\omega_{j,1}, \ldots, \omega_{j,k_j}) = 0. \]

This is equivalent to testing, for \( 1 \leq j \leq s \) the divisibility condition
\[ \Phi_{m_j}(X) \mid G_j(X^{d_{j,1}}, \ldots, X^{d_{j,k_j}}), \]
which can be checked by the method used in the step (2.c.ii) of Algorithm 1 in
\[ O \left( N^4 (\log N)^2 2^{3\sqrt{N \log N}} (h + M (\log(d))) \right) \]
bit operations.

Moreover, since there are at most \( N! \) partitions to consider at step (2) according to Lemma 2.5 and since by Lemma 2.6 there are at most
\[ \exp \left( 3(n + 1) \sqrt{N \log N} \right) \]
torsion points to consider at step (2.c), the total number of bit operations performed by the algorithm is:
\[ O \left( N^{nN} (M (\log(d)) \log \log(d) + h) \right). \]

This terminates the proof of Theorem 1.1 in the case of a hypersurface.

4. Computing the Torsion Cosets of a Variety

We finally prove Theorem 1.1 in the general case. In the following, we fix \( F_1, \ldots, F_k \in \mathbb{Z}[X_1, \ldots, X_n] \) and we would like to compute a representation of the torsion cosets in the variety:
\[ V := \{ \bar{x} \in \mathbb{C}_m^n : F_1(\bar{x}) = \ldots = F_k(\bar{x}) \}. \]

We will explain how we can modify Algorithm 2 of the last section to treat the case of several polynomials. We remark that the same modifications can be applied to Algorithm 1 to compute the common cyclotomic factors of several univariate polynomials.

Let \( \zeta \in V \cap \mu_\infty \) and \( N_i \) be the number of nonzero terms in \( F_i \), for \( 1 \leq i \leq k \) and also let \( N := \max(N_1, \ldots, N_k) \). For each integer \( i \) between 1 and \( k \), we can reduce the equality \( F_i(\zeta) = 0 \) to minimal sums as we have done in previous sections, by considering a partition of the set \( \{1, \ldots, N_i\} \). Once performed this step for every
integer \( i \), we are reduced to step (2) of Algorithm 2 and all other steps of Algorithm 2 can be kept.

The number of partitions to consider is bounded now by:

\[
N_1! \times \cdots \times N_k! \leq (N!)^k.
\]

The number of torsion points which appear at step (2.c) is bounded now by:

\[
\exp \left( 3(n+1)\sqrt{kN \log kN} \right).
\]

Finally, the total number of bit operations executed by the algorithm is:

\[
O \left( N^{nkN} (M(\log(d)) \log(d) + h) \right).
\]

This concludes the proof of Theorem 1.1. \( \Box \)

5. An upper bound for the number of torsion cosets

We prove Corollary 1.2. Let \( V \) be a variety defined by \( k \) polynomials, each having at most \( N \) nonzero terms. Our algorithm outputs a representation of \( V_{\text{tors}} \) as the union of torsion cosets. We remark that each loop of Algorithm 2 outputs at most one torsion coset. Thus the total number \( t \) of torsion cosets included in \( V \) is bounded by the number of loops performed by the algorithm. Every loop corresponds to a partition of these polynomials and to a choice of a point in \( \mu^\infty_n \) of squarefree order \( m \) satisfying \( \Psi(m) \leq N \). As we have seen before, there are at most \( (N!)^k \) partitions and the number of points in \( \mu^\infty_n \) to consider is bounded by \( \exp \left( 3(n+1)\sqrt{kN \log kN} \right) \). Thus

\[
t \leq (N!)^k \exp \left( 3(n+1)\sqrt{kN \log kN} \right),
\]

which is the desired upper bound. \( \Box \)

Acknowledgements

I would like to thank Francesco Amoroso and Martín Sombra for their precious help and many suggestions. I also express my gratitude to Denis Simon who suggested to me the example in Section 2.3.

References


