ALGEBRAIC AND EUCLIDEAN LATTICES: OPTIMAL LATTICE REDUCTION AND BEYOND

PAUL KIRCHNER, THOMAS ESPITAU AND PIERRE-ALAIN FOUQUE

Abstract. We introduce a framework generalizing lattice reduction algorithms to module lattices in order to practically and efficiently solve the γ-Hermite Module-SVP problem over arbitrary cyclotomic fields. The core idea is to exploit the structure of the subfields for designing a doubly-recursive strategy of reduction: both recursive in the rank of the module and in the field we are working in. Besides, we demonstrate how to leverage the inherent symplectic geometry existing in the tower of fields to provide a significant speed-up of the reduction for rank two modules. The recursive strategy over the rank can also be applied to the reduction of Euclidean lattices, and we can perform a reduction in asymptotically almost the same time as matrix multiplication. As a byproduct of the design of these fast reductions, we also generalize to all cyclotomic fields and provide speedups for many previous number theoretical algorithms.

Quantitatively, we show that a module of rank 2 over a cyclotomic field of degree n can be heuristically reduced within approximation factor $2^{\tilde{\mathrm{O}}(n)}$ in time $\tilde{\mathrm{O}}\big(n^2 B\big)$, where B is the bitlength of the entries. For B large enough, this complexity shrinks to $\tilde{\mathrm{O}}\big(n^{\log_2 3} B\big)$. This last result is particularly striking as it goes below the estimate of $n^2 B$ swaps given by the classical analysis of the LLL algorithm using the so-called potential.

Finally, all this framework is fully parallelizable, and we provide a full implementation. We apply it to break multilinear cryptographic candidates on concrete proposed parameters. We were able to reduce matrices of dimension 4096 with 6675-bit integers in 4 days, which is more than a million times faster than previous state-of-the-art implementations. Eventually, we demonstrate a quasicubic time for the Gentry-Szydlo algorithm which finds a generator given the relative norm and a basis of an ideal. This algorithm is important in cryptanalysis and requires efficient ideal multiplications and lattice reductions; as such we can practically use it in dimension 1024.

This work has been supported in part by the European Union H2020 Programme under grant agreement number ERC-669891 and Prometheus Project-780701.

1. Introduction

Lattice-based cryptography increasingly uses ideal and module lattices for efficiency reasons as the NTRU cryptosystem since 1996. This achieves quasilinear key size, encryption/decryption and signature time complexities instead of quadratic. Consequently, it is of utmost importance to reduce such lattices very efficiently. Peikert in [41] asked the following question: *For worst-case problems on ideal lattices, especially in cyclotomic rings, are there (possibly quantum) algorithms that substantially outperform the known ones for general lattices? If so, do these attacks also extend to work against the ring-SIS and ring-LWE problems themselves ?* So far, there is no result in this direction and the security parameters are chosen so that these lattices are as hard to reduce as random lattices.

The classical way of reducing algebraic lattices starts by *descending* the algebraic lattice over the integers $\mathbf{Z}$. This corresponds to forgetting the algebraic structure of the module and running a reduction algorithm on it. But the image over $\mathbf{Z}$ of a rank d algebraic lattice is of rank $d \times n$, where n is the degree of field inside which we are working initially. Hence, even in the case where the lattice is of small rank, the reduction can be very costly as the actual dimension over $\mathbf{Z}$ might be large. This process is forgetful of the algebraic specificities of the base ring. But these properties translate into symmetries over modules, as they are very structured. Consequently, the above-mentioned reduction cannot take these symmetries into account. Thus, it is natural to wonder if it is possible to *exploit* the algebraic structure of the fields to speed up the reduction.

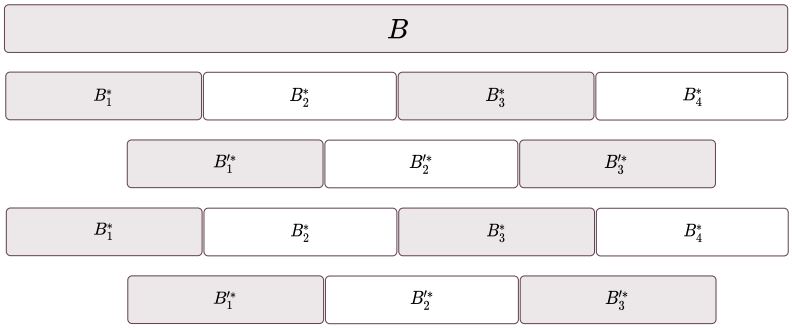
In this paper, we present several optimal and heuristic algorithms for LLL-reducing lattices defined over $\mathbf{Z}$ and more generally over module lattices defined over cyclotomic fields [30]. In the special case of rank-2 module, which is the case in the cryptanalysis of the NTRU cryptosystem [21], we describe more specific algorithms. One of them takes into account the symplectic structure of these lattices. Since recent advanced cryptographic constructions such as multilinear maps [2] and fully homomorphic encryption schemes [49, 10] increasingly use lattices with high dimension and very large numbers, our goal is to give *very efficient* and *parallel* algorithms to reduce them. Consequently, we depart from the current research line of proved worst-case lattice reductions to present heuristic algorithms with high performance. However, the introduced heuristics are practically verified and a large part of the algorithms is proven.

1.1. Technical framework.

We introduce a framework of techniques to provide fast polynomial-time algorithms for reducing algebraic lattices defined over cyclotomic fields. The core design principles of our reductions are:

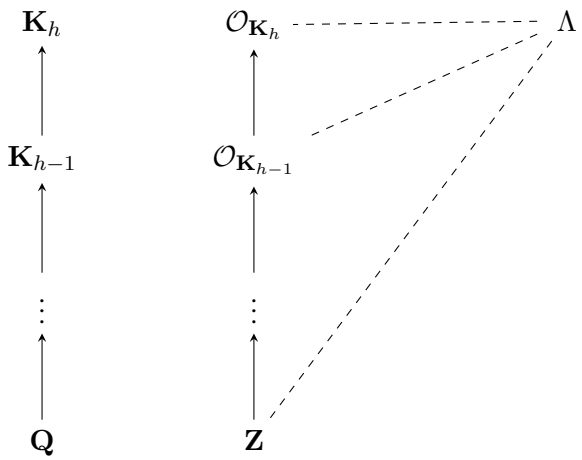
A recursive strategy on the rank: The reduction of a rank d lattice is performed recursively on large blocks. Instead of relying on a local (LLL-like) strategy consisting in choosing the first (or an arbitrary) block for which some progress can be made, we systematically perform the reduction of the blocks. This global process is somewhat similar to the ironing out strategies of BKZ-like reductions or to the fast variant of LLL of Neumaier and

Stehlé [39], where successive passes of local reductions are made on the *whole* basis to gradually improve its reduceness. However, we differ from the iterative design à la BKZ as we shift the blocks between odd and even steps to mix all basis vectors as in the early parallelized versions of LLL of Villard [51]. A generic instance of two successive passes of our strategy is given in the following:



The basis B is here sundered in four chunks B_1, B_2, B_3, B_4 of length $|B|/4$. The reduction process will start by reducing (possibly at the same time) the first chunk $B_1^* = B_1$, the projection B_2^* of the second one orthogonally to B_1, the projection B_3^* of the third one orthogonally of $B_1 \| B_2$ and so on. When this pass is over, the same process starts again, but this time on shifted blocks (i.e. the first block B_1' starts with the vector $|B|/8$ and is of length $|B|/4$). Hence, the rank of the lattices which are called recursively decreases until we reach rank 2 lattices, where we can use a fast reduction like Schönhage's algorithm [46].

A recursive strategy on the degree of the field: Suppose that we are given a tower of number fields $\mathbf{K}_0 \subset \mathbf{K}_1 \subset \ldots \subset \mathbf{K}_h$. Let Λ be an algebraic lattice defined over the ring of integers of the upper field $\mathbf{K}_h$. We can look at Λ as an algebraic lattice defined over the field right under, that is $\mathbf{K}_{h-1}$.



Such an identification is possible at the cost of increasing the rank of the lattice: the rank of Λ seen over $\mathbf{K}_{h-1}$ is exactly $[\mathbf{K}_h : \mathbf{K}_{h-1}]$ times its rank over $\mathbf{K}_h$. Then we make use of the recursive design over the rank, introduced above, to reduce this problem into numerous instances of reduction of rank two lattices over $\mathbf{K}_{h-1}$. Each of them can be seen over $\mathbf{K}_{h-2}$, inviting us to pursue this descent until we get to the bottom of the tower and are now reducing lattices over $\mathbf{Z}$, that is, Euclidean lattices.

A generic use of symplectic structures in number fields: A Euclidean space is a vector space endowed with a positive definite symmetric bilinear form acting on it. Replacing this form by an antisymmetric one yields the notion of *symplectic space*. Lattices embedded in symplectic spaces have additional symmetries that can be exploited to (roughly) halve the cost of the reduction. We prove that we can define a recursive symplectic structure over a tower of number fields. As a consequence we can halve the running time of the reduction at *each* level of the recursion tree, yielding significant asymptotic speedups on the overall reduction.

A (controlled) low precision reduction: We use approximations instead of exact computations, which corresponds to reducing the projected sublattices with only the most significant bits of their basis. A careful analysis of the precision required to ensure a global reduction gains a factor up to d depending on the condition number of the initial basis, where d is the rank of the lattice we want to reduce. Furthermore, we can show that the precision needed will significantly decrease during *some* recursive calls, up to a factor of d once again.

A fast and generic algorithmic for the log-unit lattice: During the reduction of an algebraic lattice, we need to balance the size of the Archimedean embeddings of elements to avoid a blow-up of the precision used. This can be done by carefully multiplying the considered quantities by units of the field, yielding a decoding problem in the so-called *log-unit lattice* of cyclotomic fields. We generalize the work of Cramer, Ducas, Peikert, and

Regev [11], which proved two different results. The first is that, given a point, we can find a unit nearby with prime-power cyclotomics[1]. The second one is that, given a log-unit lattice point plus some large subgaussian noise, we can find the lattice point in polynomial time. We prove that these results can be achieved within quasilinear running time, and for any cyclotomic field.

1.2. **Results and practical considerations.** We now discuss the practical implication of the techniques above-mentioned. Using the recursion on the rank with the low precision technique yields a fast heuristic reduction algorithm for Euclidean lattices. More precisely we prove that for a Euclidean lattice given by a matrix M of dimension d with entries in $\mathbf{Z}$ of bitsize at most B, with *condition number* bounded by 2^B, our reduction algorithm finds a lattice vector v such that $\|v\| \leq 2^{\frac{d}{2}} |\det M|^{1/d}$ (that is the $2^{\frac{d}{2}}$-Hermite SVP) in time:

$$\mathrm{O}\left(\frac{d^\omega}{(\omega-2)^2} \cdot \frac{B}{\log B} + d^2 B \log B\right),$$

where ω is the exponent of matrix multiplication. We give in appendix D a reduction from lattice reduction to modular linear algebra which suggests that this complexity is almost optimal. We also show that for the ubiquitous "knapsack-like" matrices, we can further reduce by a factor of d the complexity.

Combining the recursion over the degree of the number fields yields a reduction algorithm for module lattices over cyclotomic fields. Over a cyclotomic field of degree n and sufficiently smooth conductor, we can reduce a rank two module represented as a 2×2 matrix M whose number of bits in the input coefficients is uniformly bounded by $B > n$, in time

$$\tilde{\mathrm{O}}\big(n^2 B\big).$$

The first column of the reduced matrix has its coefficients uniformly bounded by $2^{\tilde{\mathrm{O}}(n)} (\operatorname{vol} M)^{\frac{1}{2n}}$. Using the symplectic technique gives the fastest heuristic reduction algorithm over cyclotomic fields, achieving the same approximation factor of $2^{\tilde{\mathrm{O}}(n)}$ in time:

$$\tilde{\mathrm{O}}\left(n^{2+\frac{\log(1/2+1/2q)}{\log q}} B\right) + n^{\mathrm{O}(\log\log n)}$$

where q is a prime, and the conductor is a power of q.

A note on the approximation factor. It is noticeable that the approximation factor increases quickly with the height of the tower. If we can perform a reduction over a number field above $\mathbf{Q}$ directly, then there is no need to descend to a $\mathbf{Z}$-basis and we can instead stop at this intermediate level. Actually, the larger the ring is, the more efficient the whole routine is. It is well-known that it is possible to come up with a direct reduction algorithm for an algebraic lattice when the underlying ring of integer is norm-Euclidean, as first mentioned by Napias [37]. The

[1]This was later extended by Wesolowski [53] to all cyclotomics, however the running time is still superquadratic.

reduction algorithm over such a ring $\mathcal{O}_{\mathbf{K}}$ can be done exactly as for the classical LLL algorithm, by replacing the norm over $\mathbf{Q}$ by the algebraic norm over $\mathbf{K}$. Hence a natural choice would be $\mathbf{Z}[x]/(x^n+1)$ with $n \leq 8$ as these rings are proved to be norm-Euclidean. We explain in section 7 how we can in fact deal with the larger ring $\mathbf{Z}[x]/(x^{16}+1)$ even though it is not norm-Euclidean. In several applications, it is interesting to decrease the approximation factor. Our technique is, at the lowest level of recursion, and when the number of bits is low, to use a LLL-type algorithm. Each time the reduction is finished, we descend the matrix to a lower level where the approximation factor is lower.

1.2.1. *Practical impact in cryptography.* We test our algorithm on a large instance coming from multilinear map candidates based on ideal lattices proposed in [2] where $q \approx 2^{6675}$ and $N = 2^{16}$. We solve this instance over the smaller field $n = 2^{11}$ in 13 core-days. If we compare this computation with the previous large computation with fplll, Albrecht *et al.* were able to compute with $n = 2^8$, $q \approx 2^{240}$ in 120 hours. As the complexity of their code is about $n^4 \log(q)^2$ we can estimate our improvement factor to 4 million.

As a byproduct of our reduction we were also able to drastically enhance the Gentry-Szydlo algorithm [16]. The key in this algorithm is to quicken the ideal arithmetic. Instead of the classical $\mathbf{Z}$-basis representation, we choose to represent ideals with a small family of elements over the order of a subfield of $\mathbf{K}$. Then, one can represent the product of two ideals using the family of all products of generators. However, this leads to a blow-up in the size of the family. A reasonable approach is then to sample a bit more than $[\mathbf{L} : \mathbf{K}]$ random elements in the product so that with overwhelming probability the ideal generated by these elements is the product ideal itself. It then suffices to reduce the corresponding module with the fast reduction process to go back to a representation with few generators.

An important piece is then the reduction of an ideal itself. Our practical approach is here to reduce a square matrix of dimension $[\mathbf{L} : \mathbf{K}]$, and every two rounds to add a new random element with a small Gram-Schmidt in the ideal at the last position. We show in section 8.2.1 that the overall complexity is $\tilde{\mathrm{O}}(n^3)$, while the previous implementation was in $\mathrm{O}(n^6)$. The running time of the first practical implementation published [4] in dimension 256 was 20 hours while we were able to do it in 30 minutes. Assuming it is proportional to n^6 leads to an estimate of 10 years for $n = 1024$ while we were able to compute it in 103 hours.

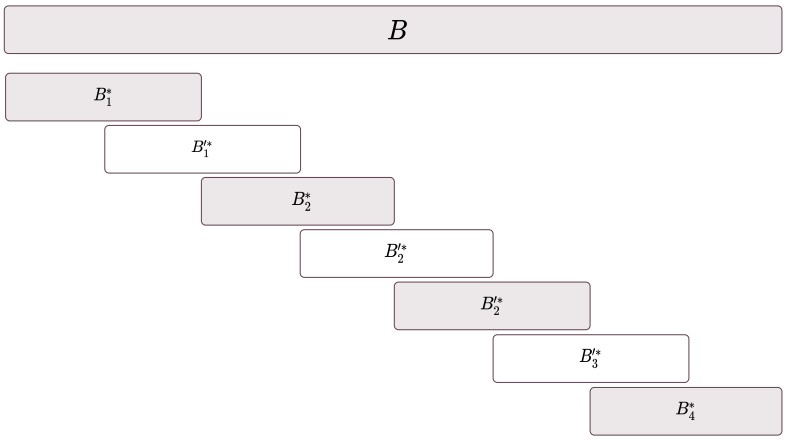
1.3. **Related Work.** Recently some independent line of research started to tackle the problem of reduction of algebraic lattices [31, 36]. These works actually provide polynomial time reduction from γ-module-SVP (or γ-Hermite-SVP) in small rank to the same problem in arbitrary rank. However, an implementation would rely on an actual oracle for this problem, yielding algorithms whose running time would be exponential in the degree of the field. We emphasize here that while our techniques rely on *many* heuristics, the resulting algorithms are implemented and enable a fast reduction of high-dimensional lattices.

The fastest (theoretical) asymptotic variant of the LLL reduction is the recursive strategy of Neumaier and Stehlé [39], whose running time is

$$d^4 B^{1+o(1)}$$

for an integer lattice of rank d with coefficients of bitsize smaller than B.
In all applications of LLL known to the authors, the condition number of a matrix is barely larger than the matrix entries; however, we underscore that it can be much larger. Also, even though both their and our algorithms are not proven to return an LLL-reduced basis, but a basis starting with a short vector; in practice the basis returned is in fact LLL-reduced.

We give an example of a round in Neumaier-Stehlé's algorithm:



The main difference is that their round prevent any parallelism.

It is well-known since the work of Schnorr [44] that we can provably use approximations in the computations to reduce the needed bitsize. However, previous papers were limited to the "well-conditioned" LLL-reduced part of the matrix, which prevented the use of fast matrix multiplications. In contrast, we give a framework able to work with approximations, such that the the number of bits needed is within a small constant factor of the optimal. This, in turn, enables a reduction in the precision used on the partially-reduced intermediary bases.

1.4. **Organization of the paper.** In the next section, we present the mathematical objects we need in the paper and the LLL algorithm. In section 3 we present the algorithm which reduces rank 2 modules, whose complexity is analyzed in section 4. In section 5 we show how to efficiently reduce high-rank modules, and its impact on the reduction of knapsack-like bases. Then in section 6, we explain how to use the symplectic structure to obtain an even faster reduction of rank 2 modules. We describe tricks for a faster implementation in section 7 and detail applications and compare with a previous implementation in section 8.

The appendix A is dedicated to fast approximate algorithms, as well as bounding the precision needed. Appendix B explains how to round efficiently with respect to the cyclotomic units. Appendix C indicates ways to obtain a symplectic

structure with all number fields. Finally, appendix D reduces lattice reduction to modular linear algebra.

Contents

2. Background

We describe the mathematical definitions and lattice reduction algorithm. For algebraic number theory results, a comprehensive reference can be found in [38].

2.1. Notations and conventions.

The bold capitals $\mathbf{Z}$, $\mathbf{Q}$, $\mathbf{R}$ refer as usual to the ring of integers and respectively the field of rational and real. Given a real number x, its integral rounding denoted by $\lfloor x \rceil$ returns its closest. Its fractional part is the excess beyond that number's integer part and denoted by $\{x\}$.

These operators are extended to operate on vectors and matrices by point-wise composition. The complex conjugation of $z \in \mathbf{C}$ is denoted by the usual bar $\bar{z}$. The logarithm functions are used as $\log$ for the binary logarithm and $\ln$ for the natural one.

We say that an integer $n \in \mathbf{Z}$ is **log-smooth** if all the prime factors of n are bounded by $\log(n)$.

Matrix and norms. For a field $\mathbf{K}$, let us denote by $\mathbf{K}^{d\times d}$ the space of square matrices of size d over $\mathbf{K}$, $\mathrm{GL}_d(\mathbf{K})$ its group of invertibles. Denote classically the elementary matrices by $T_{i,j}(\lambda)$ and $D_i(\lambda)$ for respectively the transvection (or shear mapping) and the dilatation of parameter λ.

We extend the definition of the product for any pair of matrices (A, B): for every matrix C with compatible size with A and B, we set: $(A, B)\cdot C = (AC, BC)$.

For a vector v (resp. matrix A), we denote by $\|v\|_\infty$ (resp. $\|A\|_{\max}$) its absolute (resp. max) norm, that is the maximum of the absolute value of its coefficients.

We adopt the following conventions for submatrix extraction: for any matrix $M = (m_{i,j}) \in \mathbf{K}^{n\times n}$ and $1 \le a < b \le n, 1 \le c < d \le n$, define the extracted submatrix

$$M[a : b, c : d] = (m_{i,j})_{a\le i\le b, c\le j\le d},$$

while M_i refers to the ith column of M.

Computational setting. We use the standard model in algorithmic theory, i.e. the word-RAM with unit cost and logarithmic size register (see for instance [34,

Section 2.2] for a comprehensive reference). The number of bits in the register is w.

For a non-negative integer d, we set $\omega(d)$ to be the exponent of matrix multiplication of $d \times d$ matrices. If the dimension d is clear from context we might omit it and write simply $\mathrm{O}(d^\omega)$ for this complexity. We can assume that this exponent is not too close to 2, in particular $\omega(d) > 2 + 1/\log(d)$, so that complexities with terms in $(\omega - 2)^{-1}$ make sense. Also, we assume that ω is non-increasing. Note the conflict with Landau's notations.

2.2. **Background on Algebraic Number Theory. Number fields.** A number field $\mathbf{K}$ is an algebraic extension of $\mathbf{Q}$ such that:

$$\mathbf{K} \cong \mathbf{Q}[X]/(P) = \mathbf{Q}(\alpha),$$

where P is a monic irreducible polynomial of degree n over $\mathbf{Z}$ and α is the image of X in the quotient. For a number field $\mathbf{L}$ containing $\mathbf{K}$ denote by $[\mathbf{L} : \mathbf{K}]$ the dimension of $\mathbf{L}$ seen as a $\mathbf{K}$-vector space. This integer is called the relative degree of $\mathbf{L}$ to $\mathbf{K}$. Any element γ of $\mathbf{K}$ has a minimal polynomial, i.e. the unique monic polynomial of least degree among all polynomials of $\mathbf{Q}[X]$ vanishing at γ. An *algebraic integer* has its minimal polynomial in $\mathbf{Z}[X]$. The set of all integers in $\mathbf{K}$ forms a ring, called the *ring of integers* or *maximal order* of $\mathbf{K}$, $\mathcal{O}_\mathbf{K}$.

Let $(\alpha_1, \ldots, \alpha_n) \in \mathbf{C}^n$ be the distinct complex roots of P. Then, there are n distinct embeddings, field homomorphisms, of $\mathbf{K}$ in $\mathbf{C}$. We define the i-th embedding $\sigma_i : \mathbf{K} \to \mathbf{C}$ as the morphism mapping α to α_i. We distinguish embeddings induced by real roots, *real embeddings* from embeddings coming from complex roots, *complex embeddings*. Assume that P has r_1 real roots and r_2 complex roots, $n = r_1 + r_2$. This leads to the Archimedean *embedding* σ:

$$\begin{array}{rccl} \sigma : & \mathbf{K} & \longrightarrow & \mathbf{R}^{r_1} \times \mathbf{C}^{r_2} \\ & x & \longmapsto & (\sigma_1(x), \ldots, \sigma_{r_1}(x), \sigma_{r_1+1}(x), \ldots \sigma_{r_1+r_2}(x)). \end{array}$$

This embedding can be used to define a Hermitian symmetric bilinear form on $\mathbf{K}$, which is positive definite and endows $\mathbf{K}$ with a natural Hermitian structure:

$$\langle a, b \rangle_\sigma = \sum_{i=1}^{n} \sigma_i(a)\overline{\sigma_i(b)}.$$

Modules and Ideals. Let fix R be a ring with multiplicative identity 1_R. A R-module $\mathcal{M}$ consists of an abelian group $(\mathcal{M}, +)$ and a composition law $\cdot : R \times \mathcal{M} \to \mathcal{M}$ which is bilinear and associative. Suppose $\mathcal{M}$ is a R-module and $\mathcal{N}$ is a subgroup of $\mathcal{M}$. Then $\mathcal{N}$ is a R-submodule if, for any v in $\mathcal{N}$ and any r in R, the product $r \cdot v$ is in $\mathcal{N}$. A R-module $\mathcal{M}$ is said to be *free* if it is isomorphic to R^d for some positive integer d. Consequently, there exists a set of elements $v_1, \ldots, v_d \in \mathcal{M}$ so that every element in $\mathcal{M}$ can be uniquely written as an R-linear combination of the v_i's. Such a family is called a basis of the module.

An ideal of $\mathcal{O}_\mathbf{K}$ is as an $\mathcal{O}_\mathbf{K}$-submodule of $\mathcal{O}_\mathbf{K}$. Every ideal $\mathfrak{a}$ in number fields are finitely generated modules that is it can be described by a finite family of generators i.e. expressed as $\alpha_1\mathcal{O}_\mathbf{K} + \cdots + \alpha_k\mathcal{O}_\mathbf{K}$, for some integer k with the (α_i)

belongings to $\mathcal{O}_{\mathbf{K}}$. Since the ring $\mathcal{O}_{\mathbf{K}}$ is Dedekind, any ideal can be generated by two elements. The product of two ideals $\mathfrak{a}$ and $\mathfrak{b}$ is defined as follows

$$\mathfrak{ab} := \{a_1 v_1 + \cdots + a_m v_m \mid a_i \in \mathfrak{a} \text{ and } v_i \in \mathfrak{b}, i \in \{1, \ldots, n\};\ \text{for } m \in \mathbf{N}\},$$

i.e., the product is the ideal generated by all products ab with $a \in \mathfrak{a}$ and $b \in \mathfrak{b}$.

Trace and norm in K. Let $\mathbf{K} \subset \mathbf{L}$ a number field extension and $n = [\mathbf{L} : \mathbf{K}]$. Let $\sigma_i^{\mathbf{K}} : \mathbf{L} \to \mathbf{C}$ the n field embeddings fixing $\mathbf{K}$. For any element $\alpha \in \mathbf{L}$ define its (relative) algebraic norm $\mathcal{N}_{\mathbf{L}/\mathbf{K}}(\alpha)$ to be the determinant of the $\mathbf{K}$-linear map $x \mapsto x\alpha$. One can describe this norm using the $\sigma_i^{\mathbf{K}}$ embeddings as: $\mathcal{N}_{\mathbf{L}/\mathbf{K}}(\alpha) = \prod_{1 \le i \le n} \sigma_i^{\mathbf{K}}(\alpha)$, showing in particular that the relative norm is multiplicative. Similarly define its (relative) trace $\mathrm{tr}_{\mathbf{L}/\mathbf{K}}(\alpha)$ to be the trace of the $\mathbf{K}$-linear map $x \mapsto x\alpha$. This trace is described using the $\sigma_i^{\mathbf{K}}$ embeddings as: $\mathrm{tr}_{\mathbf{L}/\mathbf{K}}(\alpha) = \sum_{1 \le i \le n} \sigma_i^{\mathbf{K}}(\alpha)$, showing in particular that the relative trace is additive. It is clear from these definitions that the for any $\alpha \in \mathbf{L}$, its relative trace and norm are elements of $\mathbf{K}$. Remark that by definition of the Archimedean structure of $\mathbf{K}$, we have $\langle a, b \rangle_\sigma = \mathrm{tr}_{\mathbf{K}/\mathbf{Q}}\left(a\bar{b}\right)$ for any elements $a, b \in \mathbf{K}$. We define the (relative) canonical norm of an element over $\mathbf{K}$ to be

$$\|\alpha\|_{\mathbf{L}/\mathbf{K}} = \left(\mathrm{tr}_{\mathbf{L}/\mathbf{K}}(\alpha\overline{\alpha})\right)^{\frac{1}{2}}.$$

We easily derive a relation between the algebraic norm of an integer and its canonical norm, based on the inequality of arithmetic and geometric means.

Lemma 1 (Inequality between relative arithmetic and geometric norms)**.** *Let* $\mathbf{Q} \subset \mathbf{K} \subset \mathbf{L}$ *a tower of number field. For every* $\alpha \in \mathbf{L}$*:*

$$|\mathcal{N}_{\mathbf{L}/\mathbf{Q}}(\alpha)| \le \left(\frac{\mathcal{N}_{\mathbf{K}/\mathbf{Q}}(\|\alpha\|_{\mathbf{L}/\mathbf{K}})}{\sqrt{[\mathbf{L} : \mathbf{K}]}}\right)^{[\mathbf{L}:\mathbf{K}]}.$$

2.3. **Cyclotomic fields and Modules over $\mathbf{Z}[\zeta_f]$.** We denote by $\Phi_f \in \mathbf{Z}[X]$ the f-th cyclotomic polynomial, that is the unique monic polynomial whose roots $\zeta_f^k = \exp(2ik\pi/f)$ with $\gcd(k, f) = 1$ are the f-th primitive roots of the unity. The f-th cyclotomic polynomial can be written as: $\Phi_f = \prod_{k \in \mathbf{Z}_f^\times} (X - \zeta_f^k)$ and the cyclotomic field $\mathbf{Q}(\zeta_f)$ is obtained by adjoining a primitive root ζ_f to the rational numbers. As such, $\mathbf{Q}(\zeta_f)$ is isomorphic to the field $\mathbf{Q}[X]/(\Phi_f)$. Its degree over $\mathbf{Q}$ is $\deg(\Phi_f) = \varphi(f)$, the Euler totient of f. In this specific class of number fields, the ring of integers is precisely $\mathbf{Z}[X]/(\Phi_f) \cong \mathbf{Z}[\zeta_f]$ (see [38, Proposition 10.2]).

Canonical Hermitian structure. Let $\mathcal{M}$ be a free module of rank d over the cyclotomic ring of integers $\mathbf{Z}[\zeta_f]$. It is isomorphic to $\bigoplus_{i=1}^d \alpha_i \mathbf{Z}[\zeta_f]$, for some linearly independent vectors $\alpha_i \in \mathbf{Q}(\zeta_f)^d$. The Hermitian structure of $\mathbf{Q}(\zeta_f)^d$ naturally lifts to $\mathcal{M}$ as defined to $\langle \alpha_i, \alpha_j \rangle = \sum_{t=1}^d \mathrm{tr}\left(\alpha_i^{(t)}\overline{\alpha_j^{(t)}}\right)$ on the basis elements and extended by linearity. We denote by $\|\cdot\|$ the corresponding norm. More generically we also use this notation to denote the associated induced norm on endomorphisms (or matrices) over this $\mathbf{Q}(\zeta_f)^d$.

Relative structure of ring of integers in a tower. Let $\mathbf{K} \subseteq \mathbf{L}$ be a subfield of $\mathbf{L}$ of index n. Then $\mathcal{O}_\mathbf{K}$ is a subring of $\mathcal{O}_\mathbf{L}$, so that $\mathcal{O}_\mathbf{L}$ is a module over $\mathcal{O}_\mathbf{K}$. In whole generality, it is not necessarily free over $\mathcal{O}_\mathbf{K}$, but by the Steinitz theorem it is isomorphic to $\mathcal{O}_\mathbf{K}^{n-1} \oplus \mathfrak{a}$ for a fractional ideal $\mathfrak{a}$ of $\mathbf{K}$ (see for instance [6, Theorem 7.23]). Nonetheless, in our case, we only consider the case where both $\mathbf{K}$ and $\mathbf{L}$ are *both* cyclotomic fields. In this precise situation, $\mathcal{O}_\mathbf{L}$ is a free $\mathcal{O}_\mathbf{K}$ module of rank n over $\mathcal{O}_\mathbf{L}$. Henceforth, the module $\mathcal{M}$ can itself be viewed as a free module over $\mathcal{O}_\mathbf{K}$ of rank dn. Indeed, consider $(\xi_1, \ldots, \xi_n)$ a basis of $\mathcal{O}_\mathbf{K}$ over $\mathcal{O}_\mathbf{L}$ and $(v_1, \ldots, v_d)$ a basis of $\mathcal{M}$ over $\mathcal{O}_\mathbf{K}$. For any $1 \leq i \leq d$, each coefficient of the vector v_i decomposes uniquely in the basis (ξ_j). Grouping the corresponding coefficients accordingly yields a decomposition

$$v_i = v_i^{(1)}\xi_1 + \cdots + v_i^{(d)}\xi_n,$$

where $v_i^{(j)} \in \mathcal{O}_\mathbf{L}^{dn}$. The family $(v_i^{(j)}\xi_j)_{\substack{1\leq i\leq d,\\ 1\leq j\leq n}}$ is a basis of $\mathcal{M}$ viewed as $\mathcal{O}_\mathbf{K}$-module.

Unit rounding in cyclotomic fields. The group of units of a number field is the group of invertible elements of its ring of integers. Giving the complete description of the units of a generic number field is a computationally hard problem in algorithmic number theory. It is possible to describe a subgroup of finite index of the unit group, called the *cyclotomic units*. This subgroup contains all the units that are products of elements[2] of the form $\zeta_f^i - 1$ for any $1 \leq i \leq f$.

As these units are dense, structured and explicit we can use them to round an element. The following theorem is a fast variant of [11, Theorem 6.3], and is fully proved in appendix B.

Theorem 1. *Let $\mathbf{K}$ be the cyclotomic field of conductor f. There is a quasi-linear randomized algorithm that given any element in $x \in (\mathbf{R}\otimes\mathbf{K})^\times$ finds a unit $u \in \mathcal{O}_\mathbf{K}^\times$ such that for any field embedding $\sigma : \mathbf{K} \to \mathbf{C}$ we have*

$$\sigma\left(xu^{-1}\right) = 2^{O\left(\sqrt{f \log f}\right)} \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(x)^{\frac{1}{\varphi(f)}}.$$

Remark 1. *Recall that $\frac{f}{\varphi(f)} = O(\log\log f)$, then denoting by $n = \varphi(n)$ the dimension of $\mathbf{K}$, we then shall use the bound*

$$2^{O\left(\sqrt{n \log n \log\log n}\right)} \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(x)^{\frac{1}{n}},$$

in the result of theorem 1.

We call **Unit** the corresponding program.

2.4. **Lattice.**

Definition 1 (Lattice)**.** *A* lattice Λ *is a finitely generated free $\mathbf{Z}$-module, endowed with a Euclidean norm on $\|.\|$ on the rational vector space $\Lambda \otimes_\mathbf{Z} \mathbf{Q}$.*

[2]One should notice that $\zeta_f^i - 1$ is not a unit for f a prime-power.

We may omit to write down the norm to refer to a lattice Λ when there is no ambiguity. By definition of a finitely-generated free module, there exists a finite family $(v_1, \ldots, v_d) \in \Lambda^d$ such that $\Lambda = \bigoplus_{i=1}^{d} v_i \mathbf{Z}$, called a *basis* of Λ. Every basis has the same number of elements called the rank of the lattice.

Two different bases of the same lattice Λ are related by a unimodular transformation, which is a linear transformation represented by an element of $\mathrm{GL}_d(\mathbf{Z})$, set of $d \times d$ integer-valued matrices of determinant ± 1. Thus, algorithms acting on lattice bases can be seen as sequences of unimodular transformations. Among these procedures, reduction algorithms are of the utmost importance. They aim at congenial classes of bases, proving that for any lattice, one can efficiently find *quasi-orthogonal* bases with controlled norm vectors. The *volume* of a lattice is defined to be the square root of the Gram-matrix of any basis, that is:

$$\operatorname{vol} \Lambda = \sqrt{\det(\langle v_i, v_j \rangle)_{i,j}}$$

Orthogonalization of vectors in Hermitian space.

Let $\mathcal{S} = (v_1, \ldots, v_d)$ a family of linearly independent vectors of a space E. The orthogonal complement $\mathcal{S}^\perp$ is the subspace $\{x \in E \mid \forall i,\ \langle x, v_i \rangle = 0\}$. Denote by π_i the orthogonal projection on $(v_1, \ldots, v_{i-1})^\perp$, with the convention that $\pi_1 = \mathrm{Id}$. The Gram-Schmidt orthogonalization process (GSO) is an algorithmic method for orthogonalizing $\mathcal{S}$ while preserving the increasing chain of subspaces $(\bigoplus_{j=1}^{i} v_j \mathbf{R})_i$. It constructs the orthogonal set $\mathcal{S}^* = (\pi_1(v_1), \ldots, \pi_d(v_d))$. For notational simplicity we refer generically to the orthogonalized vectors of such family by v_i^* for $\pi_i(v_i)$. The computation of $\mathcal{S}^*$ can be done inductively as follows: for all $1 \leq i \leq d$,

$$v_i^* = v_i - \sum_{j=1}^{i-1} \frac{\langle v_i, v_j^* \rangle}{\langle v_j^*, v_j^* \rangle} v_j.$$

Collect the family $\mathcal{S}$ in a matrix S; the Gram-Schmidt transformation corresponds to the QR decomposition of S. Namely we have $S = QR$ for an orthogonal matrix Q and an upper triangular matrix R, where $R_{i,j} = \frac{\langle v_i, v_j^* \rangle}{\|v_j^*\|}$ and $Q = \left[\frac{v_1^*}{\|v_1^*\|}, \ldots, \frac{v_d^*}{\|v_d^*\|}\right]$

The volume of the parallelepiped spanned by the vectors of S can be computed from the Gram-Schmidt vectors S^* as: $\mathrm{vol}(S) = \prod_{i=1}^{d} \|v_i^*\|$.

Size-reduction of a family of vectors. Let Λ be a rank d lattice given by a basis $(v_1, \ldots, v_d)$, we might want to use the Gram-Schmidt process. However since the quotients $\frac{\langle v_i, v_j^* \rangle}{\langle v_j^*, v_j^* \rangle}$ are not integral in general, the vectors v_i^* may not lie in Λ. However, we can approximate the result of this process by taking a rounding to a nearest integer. This process is called *Size-reduction* and corresponds to the simple

iterative algorithm, where v_j^* refers to the current value of v_j:

for $i = 2$ **to** n **do**
  for $j = i - 1$ **to** 1 **do**
    $v_i \leftarrow v_i - \left\lceil \frac{\langle v_i, v_j^* \rangle}{\langle v_j^*, v_j^* \rangle} \right\rfloor v_j$
  end
end

2.5. **The LLL reduction algorithm.** Lenstra, Lenstra, and Lovász [32] proposed a notion called *LLL-reduction* and a polynomial time algorithm that computes an LLL-reduced basis from an arbitrary basis of the same lattice. Their reduction notion is formally defined as follows:

Definition 2 (LLL reduction)**.** *A basis $\mathcal{B}$ of a lattice is said to be δ-LLL-reduced for certain parameters $1/4 < \delta \leq 1$, if the following two conditions are satisfied:*

$$\forall i < j, \quad |\langle v_j, v_i^* \rangle| \leq \frac{1}{2}\|v_i^*\|^2 \quad \textit{(Size-Reduction condition)}$$

$$\forall i, \quad \delta\|v_i^*\|^2 \leq \|v_{i+1}^*\|^2 + \frac{\langle v_{i+1}, v_i^* \rangle^2}{\|v_i^*\|^2} \quad \textit{(Lovász condition).}$$

To find a basis satisfying these conditions, it suffices to iteratively modify the current basis at any point where one of these conditions is violated. This yields the simplest version of the LLL algorithm as described in algorithm 1. The method can be extended to lattices described by a generating family rather than by a basis [42].

Algorithm 1 — Textbook LLL reduction

Input: Initial basis $B = (b_1, \ldots, b_d)$
Result: A δ-LLL-reduced basis

$k \leftarrow 1$
while $k < d$ **do**
  Compute the R part of the QR-decomposition of B
  for $j = k - 1$ **downto** 1 **do**
    $b_k \leftarrow b_k - \lceil R_{k,j} \rfloor \cdot b_j$
    $R_k \leftarrow R_k - \lceil R_{k,j} \rfloor \cdot R_j$
  end for
  if $\delta\|(R_{k,k}, 0)\|^2 \leq \|(R_{k+1,k}, R_{k+1,k+1})\|^2$ **then**
    $k \leftarrow k + 1$
  else
    Swap b_k and b_{k+1}
    $k \leftarrow \max(k - 1, 1)$
end while
return $(b_1, \ldots, b_d)$

Decrease of the potential and complexity. The algorithm can only terminate when the current lattice basis is LLL-reduced. Moreover, as shown in [32], it terminates in polynomial time when $\delta < 1$. Indeed, consider the (square of the) product of the volumes of the flag associated with the basis: $\prod_{i=1}^{d} \|v_i^*\|^{2(n-i+1)}$, which is often called its *potential*. This value decreases by a factor at least δ^{-1} in each exchange step and is left unchanged by other operations. Indeed:

- The flag is not modified by any operation other than swaps.
- A swap between v_k and v_{k-1} only changes the sublattice spanned by the $k-1$ first vectors. The corresponding volume $\prod_{i=1}^{k-1} \|v_i^*\|^2$ decreases by a factor at least δ^{-1} and so does the potential.

Since the total number of iterations can be bounded by twice the number of swaps plus the dimension of the lattice, this suffices to conclude that it is bounded by $\mathrm{O}\big(d^2 B\big)$ where B is a bound on the size of the coefficients of the matrix of the initial basis. As the cost of a loop iteration is of $\mathrm{O}\big(d^2\big)$ arithmetic operations on *rational* coefficients of length at most $\mathrm{O}(dB)$, the total cost in term of arithmetic operations is loosely bounded by $\mathrm{O}\big(d^6 B^3\big)$.

Reduceness of LLL-reduced bases and approximation factor. Let Λ be a rank d lattice and $v_1, \ldots, v_d$ a δ-LLL reduced basis of Λ. The length of vectors and orthogonality defect of this basis is related to the reduction parameter δ:

Proposition 1. *Let $1/4 < \delta < 1$ be an admissible LLL parameter. Let $(v_1, \ldots, v_d)$ a δ-LLL reduced basis of rank-d lattice $(\Lambda, \langle \cdot, \cdot \rangle)$. Then for any $1 \leq k \leq d$:*

$$\mathrm{vol}(v_1, \ldots, v_k) \leq (\delta - 1/4)^{-\frac{(d-k)k}{4}} \, \mathrm{vol}(\Lambda)^{\frac{k}{d}}.$$

2.6. **$\mathcal{O}_{\mathbf{K}}$-lattices.** We now generalize the notion of Euclidean lattice to the higher-degree context. As a lattice is a finitely generated free $\mathbf{Z}$-module Λ endowed with a Euclidean structure on its real ambient space $\Lambda \otimes_{\mathbf{Z}} \mathbf{R}$. To extend this definition we want to replace the base-ring $\mathbf{Z}$ by the ring of integer $\mathcal{O}_{\mathbf{K}}$ of a number field $\mathbf{K}$. In the present context we will keep the freeness condition of the module, even if this setting is slightly too restrictive in general[3].

Definition 3 ($\mathcal{O}_{\mathbf{K}}$-lattice)**.** *Let $\mathbf{K}$ be a cyclotomic number field. An $\mathcal{O}_{\mathbf{K}}$-lattice—or algebraic lattice over $\mathcal{O}_{\mathbf{K}}$—is a free $\mathcal{O}_{\mathbf{K}}$-module Λ endowed with a $\mathbf{K} \otimes \mathbf{R}$-linear positive definite self-adjoint[4] form on the ambient vector space $\Lambda \otimes_{\mathcal{O}_{\mathbf{K}}} \mathbf{R}$.*

As for Euclidean lattices without loss of generality we can only look at the case where the inner product is the one derived from the polarization of the canonical norm introduced in section 2.2. As for the Euclidean case we now study the orthogonalization process in such space and devise the equivalent notion of the volume of an algebraic lattice.

[3]Indeed in a tower of field $\mathbf{Q} \subseteq \mathbf{K} \subseteq \mathbf{L}$, the module $\mathcal{O}_{\mathbf{L}}$ seen over the Dedekind domain $\mathcal{O}_{\mathbf{K}}$ is not necessarily free. Hence using as definition for such a generalized lattice Λ to be a free $\mathcal{O}_{\mathbf{L}}$-module would forbid Λ to be a lattice over $\mathcal{O}_{\mathbf{K}}$. Relaxing the freeness into projectiveness is however sufficient as $\mathcal{O}_{\mathbf{L}}$ is always a projective $\mathcal{O}_{\mathbf{K}}$-module.

[4]The definition of such a form is done in the usual manner: it is represented by a matrix A such that $A = A^*$ for $*$ being the composition of the transposition and conjugation operator of $\mathbf{K}_{\mathbf{R}}$.

Taking the basis $(m_1, \ldots, m_d)$ of $\mathcal{M}$, one can construct an orthogonal family $(m_1^*, \ldots, m_d^*)$ such that the flag of subspaces $(\oplus_{i=1}^k b_i \mathbf{K})_{1 \leq k \leq d}$ is preserved. This routine is exactly the same as for Euclidean lattices and is given in algorithm 2, **Orthogonalize**. We present it here in its matrix form, which generalizes the so-called QR-decomposition.

Algorithm 2 — Orthogonalize

Input : Basis $M \in \mathcal{O}_{\mathbf{K}_h}^{d \times d}$ of an $\mathcal{O}_{\mathbf{K}_h}-$module $\mathcal{M}$
Output : R part of the QR-decomposition of M

for $j = 1$ **to** d **do**
  $Q_j \leftarrow M_j - \sum_{i=1}^{j-1} \frac{\langle M_j, Q_i \rangle}{\langle Q_i, Q_i \rangle} Q_i$
end for
return $R = \left(\frac{\langle Q_i, M_j \rangle}{\|Q_i\|} \right)_{1 \leq i < j \leq d}$

The volume of S can be computed from the Gram-Schmidt vectors collected in the matrix R as: $\mathrm{vol}(\mathcal{M}) = \mathcal{N}_{\mathbf{K}/\mathbf{Q}}\left(\prod_{i=1}^d R_{i,i} \right)$.

3. Reduction of low-rank $\mathcal{O}_{\mathbf{K}}$-modules in cyclotomic fields

Let h be a non-negative integer. In the following of this section we fix a tower of log-smooth conductor cyclotomic fields $\mathbf{K}_h^{\uparrow} = (\mathbf{Q} = \mathbf{K}_0 \subset \mathbf{K}_1 \subset \cdots \subset \mathbf{K}_h)$ and denote by $1 = n_0 < n_1 < \cdots < n_h$ their respective degrees over $\mathbf{Q}$. Then we consider a free module $\mathcal{M}$ of rank d over the upper field $\mathbf{K}_h$, which is represented by a basis $(m_1, \ldots, m_d)$ given as the columns of a matrix $M \in \mathcal{O}_{\mathbf{K}_h}^{d \times d}$. For notational simplicity, in this section, we shall denote by $\langle a, b \rangle$ the $\mathcal{O}_{\mathbf{L}}$-module $a\mathcal{O}_{\mathbf{L}} \oplus b\mathcal{O}_{\mathbf{L}}$.

3.1. In-depth description of the algorithm.

3.1.1. *Outer iteration.* To reduce the module $\mathcal{M}$ we adopt an iterative strategy to progressively modify the basis: for ρ steps a reduction pass over the current basis is performed, ρ being a parameter whose value is computed to optimize the complexity of the whole algorithm while still ensuring the reduceness of the basis; we defer the precise computation of this constant to section 4. As in the LLL algorithm a size-reduction operation is conducted to control the size of the coefficients of the basis and ensure that the running time of the reduction is polynomial. Note that for number fields this subroutine needs to be adapted to deal with units of $\mathcal{O}_{\mathbf{K}_h}$ when rounding. The specificities of this size-reduction are the matter of section 3.1.5.

3.1.2. *Step reduction subroutine.* We now take a look at the step reduction pass, once the size-reduction has occurred. As observed in section 2.5, the textbook LLL algorithm epitomizes a natural idea: make the reduction process boiling down to the treatment of rank two modules and more precisely to iteratively reduce *orthogonally projected* rank two modules. We are using the same paradigm here and

this step reduction pass over the current basis is a sequence of reduction of projected rank 2 $\mathcal{O}_{\mathbf{K}_h}$−modules. However on the contrary to the LLL algorithm we do not proceed progressively along the basis, but reduce $\lfloor d/2 \rfloor$ independent rank 2 modules at each step. This design enables an efficient parallel implementation which reduces submodules simultaneously, in the same way that the classical LLL algorithm can be parallelized [51, 19].

Formally, given the basis of $\mathcal{M}$ collected in the matrix M, let us denote by r_j the vector $(R_{j,j}, R_{j+1,j} = 0)$, and r'_j the vector $(R_{j+1,j}, R_{j+1,j+1})$ where R is the R-part of the QR-decomposition of M. The module $\mathcal{R}_i$ encodes exactly the projection of $\mathcal{M}_i = \langle m_{i-1}, m_i \rangle$ over the orthogonal space to the first $i-1$ vectors $(m_1, \ldots, m_{i-1})$. In order to recursively call the reduction algorithm on $\mathcal{R}_i$ we need to *descend* it to the subfield $\mathbf{K}_{h-1}$.

3.1.3. *Interlude: descending to cyclotomic subfields.* Remark now that since $\mathbf{K}_h$ is a cyclotomic extension of the cyclotomic field $\mathbf{K}_{h-1}$, there exists a root of unity ξ such that

$$\mathcal{O}_{\mathbf{K}_h} = \mathcal{O}_{\mathbf{K}_{h-1}} \oplus \xi \mathcal{O}_{\mathbf{K}_{h-1}} \oplus \cdots \oplus \xi^{q_h - 1} \mathcal{O}_{\mathbf{K}_{h-1}}.$$

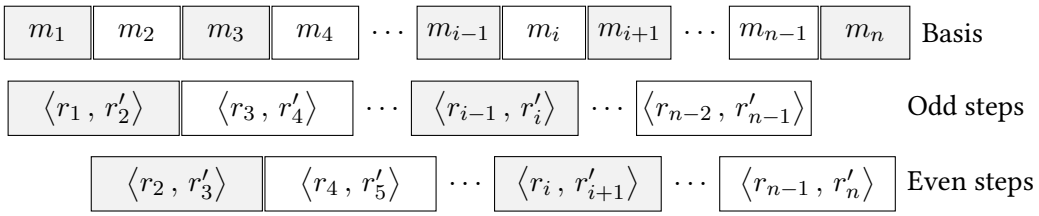
for $q_h = n_h / n_{h-1}$ being the relative degree of $\mathbf{K}_h$ over $\mathbf{K}_{h-1}$. As a consequence, the module $\mathcal{R}_i$ decomposes over $\mathcal{O}_{\mathbf{K}_{h-1}}$ as:

$$\begin{aligned}
\mathcal{R}_i &= r_i \mathcal{O}_{\mathbf{K}_h} \oplus r'_{i+1} \mathcal{O}_{\mathbf{K}_h} \\
&= r_i \mathcal{O}_{\mathbf{K}_{h-1}} \oplus \xi r_i \mathcal{O}_{\mathbf{K}_{h-1}} \oplus \cdots \oplus \xi^{q_h - 1} r_i \mathcal{O}_{\mathbf{K}_{h-1}} \oplus \\
&\quad r'_{i+1} \mathcal{O}_{\mathbf{K}_{h-1}} \oplus \xi r'_{i+1} \mathcal{O}_{\mathbf{K}_{h-1}} \oplus \cdots \oplus \xi^{q_h - 1} r'_{i+1} \mathcal{O}_{\mathbf{K}_{h-1}},
\end{aligned}$$

yielding a basis of $\mathcal{R}_i$ viewed as a free $\mathcal{O}_{\mathbf{K}_{h-1}}$-module of rank $2 \times q_h$. This module can then recursively reduced, this time over a tower of height $h-1$. This conversion from an $\mathcal{O}_{\mathbf{K}_h}$-module to an $\mathcal{O}_{\mathbf{K}_{h-1}}$ module is referred as the function **Descend**. Conversely, any vector $u \in \mathcal{O}_{\mathbf{K}_{h-1}}^{2q_h}$ can be seen with this decomposition as a vector of $\mathcal{O}_{\mathbf{K}_h}^2$ by grouping the coefficients as $\left(\sum_{i=1}^{q_h} u[i]\xi^i, \sum_{i=1}^{q_h} u[q_h + 1 + i]\xi^i\right)$. We denote by **Ascend** this conversion.

3.1.4. *Back on the step reduction.* As mentioned in section 3.1.2, we start by reducing—with a recursive call after descending—all the modules $\mathcal{R}_{2i} = \langle r_{2i-1}, r'_{2i} \rangle$ for $1 \leq i \leq \lfloor d/2 \rfloor$, so that each of these reductions yields a small element of the submodule $\mathcal{M}_{2i} = \langle m_{2i-1}, m_{2i} \rangle$; which is then *completed*[5] in a basis of $\mathcal{M}_{2i}$. But on the contrary of the classical LLL reduction, this sequence of pairwise independent reductions does not make interact the elements m_{2i} and m_{2i+1}, in the sense that no reduction of the module projected from $\langle m_{2i}, m_{2i+1} \rangle$ is performed. To do so, we then perform the same sequence of pairwise reductions but with all indices shifted by 1: we reduce the planes $\langle r_{2i}, r'_{2i+1} \rangle$ for each $1 \leq i \leq \lfloor d/2 \rfloor$, as depicted in the following diagram:

[5]The precise definition of this completion and lifting is given in section 3.1.7.



3.1.5. *Unit-size-reduction for $\mathcal{O}_{\mathbf{K}_h}$-modules.* As mentioned in section 3.1.1 in order to adapt the size-reduction process to the module setting, one needs to adjust the rounding function. When $\mathbf{K}_h = \mathbf{Q}$, the rounding boils down to finding the closest element in $\mathcal{O}_{\mathbf{K}} = \mathbf{Z}$, which is encompassed by the round function $\lceil \cdot \rfloor$. In the higher-dimensional context, we need to approximate any element of $\mathbf{K}_h$ by *a* close element of $\mathcal{O}_{\mathbf{K}_h}$.

Note that finding *the* closest integral element is not efficiently doable. The naive approach to this problem consists in reducing the problem to the resolution of the closest integer problem in the Euclidean lattice of rank n_h given by $\mathcal{O}_{\mathbf{K}_h}$ under the Archimedean embedding. However, up to our knowledge, no exponential speedup exists using its particular structure compared to sieving or enumeration in this lattice.

Nonetheless, finding a target *close enough* to the target suffices for our application. As such we simply define the rounding of an element $\alpha \in \mathbf{K}_h$ as the integral rounding on each of its coefficients when represented in the power base of $\mathbf{K}_h$.

We add here an important and necessary modification: before the actual size-reduction occurred, we compute a unit u using theorem 1 close to $R_{i,i}$. This routine is denoted by **Unit**. The vector M_i is then divided by u. While not changing the algebraic norms of the elements, this technicality forces the Archimedean embeddings of the coefficients to be balanced and helps the reduced matrix to be well-conditioned. This avoids a blow-up of the precision required during the computation. This modified size-reduction is fully described in algorithm 3, **Size-Reduce**.

Algorithm 3 — Size-Reduce

Input : R-factor of the QR-decomposition of $M \in \mathcal{O}_{\mathbf{K}_h}^{d\times d}$
Output : A unimodular transformation U representing the size-reduced basis obtained from M.

$U \leftarrow \mathrm{Id}_{d,d}$
for $i = 1$ **to** d **do**
  $D \leftarrow D_i(\mathbf{Unit}(R_{i,i}))$ // D_i is a dilation matrix
  $(U, R) \leftarrow (U, R) \cdot D^{-1}$
  for $j = i - 1$ **downto** 1 **do**
    $\sum_{\ell=0}^{n-1} r_\ell X^\ell \leftarrow R_{i,j}$ // Extraction as a polynomial
    $\mu \leftarrow \sum_{\ell=0}^{n-1} \lfloor r_\ell \rceil X^\ell$ // Approximate rounding of $R_{i,j}$ in $\mathcal{O}_{\mathbf{K}_h}$
    $(U, R) \leftarrow (U, R) \cdot T_{i,j}(-\mu)$ // $T_{i,j}$ is a shear matrix
  end for
end for
return U

3.1.6. *Reduction of the leaves.* As the recursive calls descend along the tower of number fields, the bottom of the recursion tree requires reducing $\mathcal{O}_{\mathbf{K}_0} (= \mathcal{O}_{\mathbf{Q}} = \mathbf{Z})$-modules, that is Euclidean lattices. As a consequence, the step reduction performs calls to a reduction oracle for plane Euclidean lattices. For the sake of efficiency we adapt Schönhage's algorithm [46] to reduce these lattices, which is faster than the traditional Gauss' reduction. This algorithm is an extension to the bidimensional case of the half-GCD algorithm, in the same way, that Gauss' algorithm can be seen as a bidimensional generalization of the classical GCD computation.

The original algorithm of Schönhage only deals with the reduction of binary quadratic forms, but can be straightforwardly adapted to reduce rank 2 Euclidean lattices, and to return the corresponding unimodular transformation matrix. In all of the following, we denote by **Schonhage** this modified procedure.

3.1.7. *The lifting phase.* As explained in section 3.1.2, we recursively call the reduction procedure to reduce the descent of projected modules of rank 2 of the form $\mathcal{R}_i = \langle r_i, r'_{i+1} \rangle$, over $\mathbf{K}_{h-1}$, yielding a unimodular transformation $U' \in \mathcal{O}_{\mathbf{K}_{h-1}}^{2q_h \times 2q_h}$ where q_h is the relative degree of $\mathbf{K}_h$ over $\mathbf{K}_{h-1}$.

From U', we can find random short elements in the module by computing a small linear combination of the first columns. Applying **Ascend**, we deduce some short $x = m_i a + m_{i+1} b$. But then to replace m_i by x in the current basis, we need to complete this vector into a basis (x, y) of $\mathcal{M}_i$ over $\mathcal{O}_{\mathbf{K}_h}$. Doing so boils down to complete a vector of $\mathcal{O}_{\mathbf{K}_h}^2$ into a unimodular transformation. Indeed, suppose that such a vector y is found and denote by (a, b) and (v, u) the respective coordinates of x and y in the basis (m_i, m_{i+1}). By preservation of the volume we have without

loss of generality:

$$1 = \det\begin{pmatrix} a & v \\ b & u \end{pmatrix} = au - bv.$$

Therefore finding the element y to complete x reduces to solving the Bézout equation in the unknown u and v

$$au - bv = 1 \tag{1}$$

over the ring $\mathcal{O}_{\mathbf{K}_h}$. Since this ring is in general not Euclidean we can not apply directly the Euclidean algorithm to solve this equation as an instance of the extended GCD problem. However, we can use the algebraic structure of the tower $\mathbf{K}_h^{\uparrow}$ to recursively reduce the problem to the rational integers. This *generalized* Euclidean algorithm works as follows:

If $\mathbf{K}_h = \mathbf{Q}$: then the problem is an instance of extended GCD search, which can be solved efficiently by the binary-GCD algorithm.

If the tower $\mathbf{K}_h^{\uparrow}$ is not trivial: we make use of the structure of $\mathbf{K}_h^{\uparrow}$ and first descend the problem to the subfield $\mathbf{K}_{h-1}$ by computing the relative norm $\mathcal{N}_{\mathbf{K}_h/\mathbf{K}_{h-1}}$ of the elements a and b; then by recursively calling the algorithm on these elements $\mathcal{N}_{\mathbf{K}_h/\mathbf{K}_{h-1}}(a)$ and $\mathcal{N}_{\mathbf{K}_h/\mathbf{K}_{h-1}}(b)$, we get two algebraic integers μ and ν of $\mathcal{O}_{\mathbf{K}_{h-1}}$ fulfilling the equation:

$$\mu\mathcal{N}_{\mathbf{K}_h/\mathbf{K}_{h-1}}(a) - \nu\mathcal{N}_{\mathbf{K}_h/\mathbf{K}_{h-1}}(b) = 1. \tag{2}$$

But then remark that for any element $\alpha \in \mathcal{O}_{\mathbf{K}_h}$ we have, using the comatrix formula and the definition of the norm as a determinant that: $\mathcal{N}_{\mathbf{K}_h/\mathbf{K}_{h-1}}(\alpha) \in \alpha\mathcal{O}_{\mathbf{K}_h}$, so that $\alpha^{-1}\mathcal{N}_{\mathbf{K}_h/\mathbf{K}_{h-1}}(\alpha) \in \mathcal{O}_{\mathbf{K}_h}$. Then, from eq. ((2)):

$$a \cdot \underbrace{\mu\, a^{-1}\mathcal{N}_{\mathbf{K}_h/\mathbf{K}_{h-1}}(a)}_{:=u\in\mathcal{O}_{\mathbf{K}_h}} - b \cdot \underbrace{\nu\, b^{-1}\mathcal{N}_{\mathbf{K}_h/\mathbf{K}_{h-1}}(b)}_{:=v\in\mathcal{O}_{\mathbf{K}_h}} = 1,$$

as desired.

Reduction of the size of solutions: The elements u, v found by the algorithm are not necessarily the smallest possible elements satisfying eq. ((1)). To avoid a blow-up in the size of the coefficients lifted, we do need to control the size of the solution at each step. Since the function **Size-Reduce** preserves the determinant by construction and reduces the norm of the coefficients, we can use it to reduce the bitsize of u, v to (roughly) the bitsize of a and b.

The translation of this method in pseudocode is given in algorithm 4, **G-Euclide**.

Algorithm 4 — G-Euclide, Lift

Function *G-Euclide*:
  Input : Tower of number fields $\mathbf{K}_h^{\uparrow}$, $a, b \in \mathbf{K}_h$.
  Output : $u, v \in \mathbf{K}_h$, such that $au + bv = 1$
  if $\mathbf{K}_h = \mathbf{Q}$ **then return** $\mathbf{ExGcd}(a, b)$
  $\mu, \nu \leftarrow \mathbf{G\text{-}Euclide}\left(\mathbf{K}_{h-1}^{\uparrow}, \mathcal{N}_{\mathbf{K}_h/\mathbf{K}_{h-1}}(a), \mathcal{N}_{\mathbf{K}_h/\mathbf{K}_{h-1}}(b)\right)$
  $\mu', \nu' \leftarrow \mu\, a^{-1}\mathcal{N}_{\mathbf{K}_h/\mathbf{K}_{h-1}}(a), \nu\, b^{-1}\mathcal{N}_{\mathbf{K}_h/\mathbf{K}_{h-1}}(b)$
  $W \leftarrow \begin{pmatrix} a & \nu' \\ b & \mu' \end{pmatrix}$
  $V \leftarrow \mathbf{Size\text{-}Reduce}(\mathbf{Orthogonalize}(W))$
  return $W \cdot V[2]$

Function *Lift*:
  Input : Tower of number fields $\mathbf{K}_h^{\uparrow}$, unimodular matrix $U' \in \mathcal{O}_{\mathbf{K}_{h-1}}^{2q_h}$
  Output : Unimodular matrix $U \in \mathcal{O}_{\mathbf{K}_h}^{2\times 2}$
  $a, b \leftarrow \mathbf{Ascend}(\mathbf{K}_h, U[1])$
  $\mu, \nu \leftarrow \mathbf{G\text{-}Euclide}\left(\mathbf{K}_{h-1}^{\uparrow}, a, b\right)$
  $U \leftarrow \begin{pmatrix} a & \nu \\ b & \mu \end{pmatrix}$
  return U

The number of bits needed to represent the relative norms does not depend on the subfield, and the size-reduction forces the output vector to have the same bitsize as the input one. This remark is the crux of the quasilinearity of the **G-Euclide**, as stated in lemma 4.

Remark that the algorithm needs $\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(a)$ to be prime with $\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(b)$. We assume that we can always find quickly such a, b with a short x. This will lead to heuristic 1, and the validity of this assumption is discussed in section 7.3.

3.2. **Wrapping-up.** The full outline of the reduction is given in algorithm 5 and a schematic overview of the recursive steps is provided in the diagram of fig. 1.

Algorithm 5 — Reduce

Input : Tower of cyclotomic fields $\mathbf{K}_h^{\uparrow}$, Basis $M \in \mathcal{O}_{\mathbf{K}_h}^{d \times d}$ of the $\mathcal{O}_{\mathbf{K}_h}-$module $\mathcal{M}$

Output : A unimodular transformation $U \in \mathcal{O}_{\mathbf{K}_h}^{d \times d}$ representing a reduced basis of $\mathcal{M}$.

if $d = 2$ **and** $\mathbf{K}_h = \mathbf{Q}$ **then return Schonhage**(M)
for $i = 1$ **to** ρ **do**
  $R \leftarrow$ **Orthogonalize**(M)
  $U_i \leftarrow$ **Size-Reduce**(R)
  $(M, R) \leftarrow (M, R) \cdot U_i$
  for $j = 1 + (i \mod 2)$ **to** d **by step of** *2* **do**
    if $\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R_{j,j}) \leq 2^{2(1+\varepsilon)\alpha n_h^2} \mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R_{j+1,j+1})$ **then**
      $M' \leftarrow$ **Descend**$(\mathbf{K}_{h-1}^{\uparrow}, R[j : j+1, j : j+1])$
      $U' \leftarrow$ **Reduce**$(\mathbf{K}_{h-1}^{\uparrow}, M')$
      $(U_i, M) \leftarrow (U_i, M) \cdot$ **Lift**(U')
    end if
  end for
end for
return $\prod_{i=1}^{\rho} U_i$

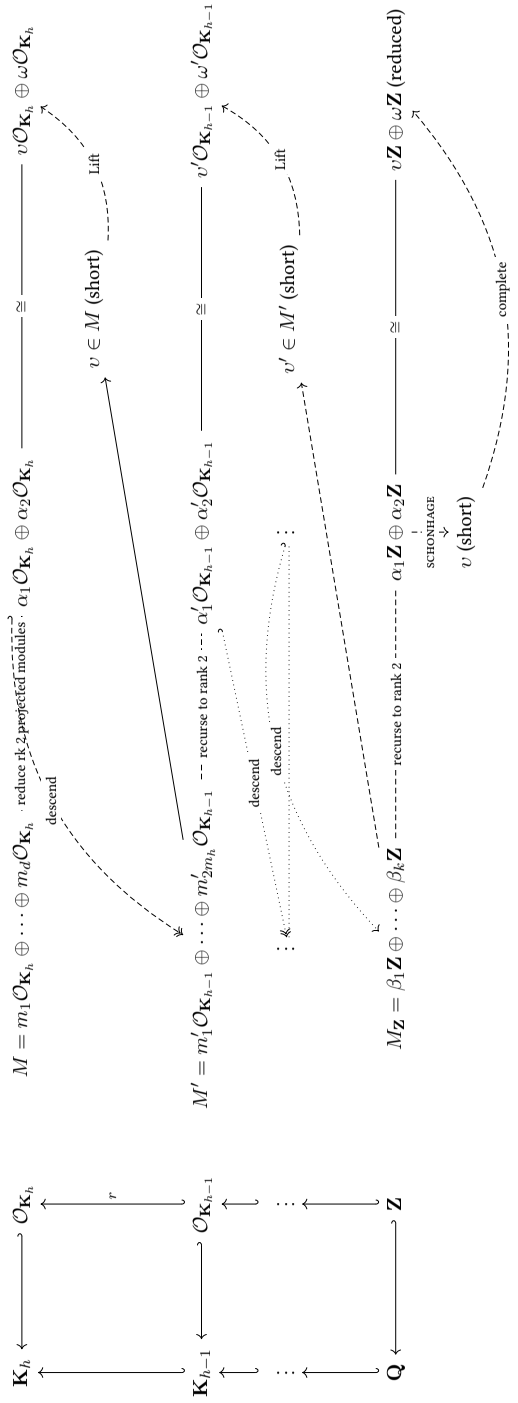


FIGURE 1. Schematic view of the recursive call of reductions.

4. Complexity analysis

In this section, we devise the complexity of the algorithm 5 and of its approximation factor. More formally we prove the following theorem:

Theorem 2. *Let f be a log-smooth integer. The complexity of the algorithm **Reduce** on rank two modules over $\mathbf{K} = \mathbf{Q}[x]/\Phi_f(x)$, represented as a matrix M whose number of bits in the input coefficients is uniformly bounded by $B > n$, is heuristically a $\tilde{O}(n^2 B)$ with $n = \varphi(f)$. The first column of the reduced matrix has its coefficients uniformly bounded by $2^{\tilde{O}(n)}(\mathrm{vol}\, M)^{\frac{1}{2n}}$.*

4.1. **Setting.** Let $h > 0$ be a non-negative integer. In the following of this section we fix a tower of cyclotomic fields $\mathbf{K}_h^{\uparrow} = (\mathbf{Q} = \mathbf{K}_0 \subset \mathbf{K}_1 \subset \cdots \subset \mathbf{K}_h)$ with log-smooth conductors and denote by $1 = n_0 < n_1 < \cdots < n_h$ their respective degrees over $\mathbf{Q}$. We consider a free module $\mathcal{M}$ of rank d over the upper field $\mathbf{K}_h$, given by one of its basis, which is represented as a matrix $M \in \mathcal{O}_{\mathbf{K}_h}^{d\times d}$. In all of the following, for any matrix A with coefficients in $\mathbf{K}_h$ we denote by $\|A\|$ the 2-norm for matrices.

We aim at studying the behavior of the reduction process given in algorithm 5 on the module $\mathcal{M}$; as such we denote generically by $X^{(\tau)}$ the value taken by any variable X appearing in the algorithm at *the beginning* of the step $i = \tau$, for $1 \leq \tau \leq \rho + 1$. For instance $R^{(1)}$ denotes the R-part of the orthogonalization of M and $M^{(\rho+1)}$ represents the reduced basis at the end of the algorithm.

Since the implementation of the algorithm is done using floating-point arithmetic, we need to set a precision which is sufficient to handle the internal values during the computation. To do so we set:

$$p = \log \frac{\max_{\sigma:\mathbf{K}_h\to\mathbf{C}, R_{i,i}\in R} \sigma(R_{i,i})}{\min_{\sigma:\mathbf{K}_h\to\mathbf{C}, R_{i,i}\in R} \sigma(R_{i,i})},$$

where the σ runs over the possible field embeddings and the $R_{i,i}$ are the diagonal values of the R part of the QR-decomposition of the input matrix of the reduction procedure. We will prove as a byproduct of the complexity analysis that taking a precision of $\mathrm{O}(p)$ suffices.

For technical reasons which will appear in the subsequent proofs, we introduce a constant $\alpha > 0$ which will be optimized at the end of our analysis. It essentially encodes the approximation factor of the reduction. Eventually, we set the variable ε to be equal to $1/2$. This apparently odd choice allows us to state our theorems with sufficient generality to reuse them in the enhanced proof of the reduction algorithm with symplectic symmetries, as detailed in section 6, with a different value.

The whole set of notations used in the analysis is recalled in table 1.

4.2. **Overview of the proof.** Before going into the details of the proof, we lay its blueprint. We start by estimating the approximation factor of the reduction and deduce a bound in $\mathrm{O}(d^2 \log p)$ on the number of rounds ρ required to achieve the reduction the module $\mathcal{M}$, where p is the precision needed to handle the full

h	Height of the tower
n_h	Absolute height $[\mathbf{K}_h : \mathbf{Q}]$
p	bound on the precision used by the reduction
ε	1/2
i	Current outmost loop number ($1 \leq i \leq \rho$) iteration
α	Constant to be optimized

TABLE 1. Notations used in the complexity analysis. p is of course set to be larger than the bitsize of the input matrix.

computation. We then prove that the limiting factor for the precision is to be sufficiently large to represent the shortest Archimedean embedding of the norm of the Gram-Schmidt orthogonalization of the initial basis. We then devise a bound by looking at the sum of all the bit sizes used in the recursive calls and concludes on the complexity. The critical part of the proof is to use the potential to show that dividing the degrees by $\frac{d}{2}$ leads to a multiplication by a factor at most in $\mathrm{O}\big(d^2\big)$ of the sum of all the precisions in the recursive calls, instead of the obvious $\mathrm{O}\big(d^3 \log p\big)$.

4.3. **A bound on the number of rounds and the approximation factor of the reduction.** We define here a set of tools to study the approximation factor of the reduction, by approximating it by an iterative linear operator on the family of volumes of the submodules $\mathcal{M}_i = m_1\mathbf{Z} \oplus \cdots \oplus m_i\mathbf{Z}$ for $1 \leq i \leq d$. This method is quite similar to the one used by Hanrot *et al.* in [18] to analyze the BKZ algorithm by studying a dynamical system.

To ease the computation of the number of rounds, we can without loss of generality, scale the input matrix and suppose that:

$$\operatorname{vol} \mathcal{M} = |\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(\det M)|^{\frac{1}{d}} = 2^{-(d+1)(1+\varepsilon)\alpha n_h^2}.$$

We only do so for this subsection.

4.3.1. *Potential and volumes of flags.* A *global* measure of reduceness of a Euclidean lattice is its potential. An $\mathcal{O}_{\mathbf{K}_h}$-analog of this constant can be defined in a similar manner by using the algebraic norm to replace the Euclidean norm over $\mathbf{R}^n$.

Definition 4 (Potential)**.** *Let $(m_1, \ldots, m_d)$ be a basis of the module $\mathcal{M}$ given as the columns of a matrix $M \in \mathcal{O}_{\mathbf{K}_h}^{d \times d}$, and let R be the R-part of its QR-decomposition.*

Its log-potential *is defined as:*

$$\Pi(M) = \sum_{i=1}^{d} \log \operatorname{vol} \mathcal{M}_i = \sum_{i=1}^{d} (d-i) \log \mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R_{i,i}).$$

As in the Euclidean case, a *local* tool to analyze the evolution of a basis $(m_1, \ldots, m_d)$ of a lattice Λ, through a reduction, is the *profile* of the volumes associated with the flag of a basis, namely the family:

$$\operatorname{vol}(\mathcal{M}_1), \ldots, \operatorname{vol}(\mathcal{M}_i), \ldots, \operatorname{vol} \Lambda.$$

As for the potential, we define the profile of the flag in a similar way with the algebraic norm on $\mathbf{K}_h$, but for technical reasons, we quadratically twist it with the constant $\alpha > 0$.

Definition 5 (Flag profile)**.** *Let $(m_1, \ldots, m_d)$ be a basis of the module $\mathcal{M}$ given as the columns of a matrix $M \in \mathcal{O}_{\mathbf{K}_h}^{d \times d}$, and let R be the R-part of its QR-decomposition. Its* profile *is the vector $\mu(M) \in \mathbf{R}^d$ defined by:*

$$\mu(M)_j = \sum_{k=1}^{j} \big(\log \mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R_{k,k}) + 2k(1+\varepsilon)\alpha n_h^2 \big), \qquad \text{for } 1 \leq j \leq d.$$

The following lemma gives an estimate of the norm of the profile in terms of the parameters of the algorithm and of the input bitsize.

Lemma 2. *With the same notations as in definition 5, we have:*

$$\|\mu(M)\|_2 \leq (2+\varepsilon)\alpha d^2 n_h p$$

Proof. We have $|\mathcal{N}_{\mathbf{K}/\mathbf{Q}}(\det(M))| \leq 1$ so for each i, and each embedding σ, we have that $|\sigma(R_{i,i})| \leq 2^p$. Now we compute:

$$\begin{aligned} \frac{\|\mu(M)\|_2^2}{d} &\leq \max_{j=1,\ldots,d-1} \left\{ \sum_{k \leq j} \big(\log \mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R_{k,k}) + 2k(1+\varepsilon)\alpha n_h^2 \big) \right\} \\ &\leq d n_h p + d(d-1)(1+\varepsilon)\alpha n_h^2 \end{aligned}$$

which implies the result. ∎

4.3.2. *A family of step operators.* To study the reduction steps, we define the following linear operators

$$\delta_j : \left| \begin{array}{ccc} \mathbf{R}^d & \longrightarrow & \mathbf{R}^d \\ v & \longmapsto & (w_\ell)_\ell = \begin{cases} \frac{v_{j-1}+v_{j+1}}{2} & \text{if } \ell = j \\ v_j & \text{if } \ell = j+1 \\ v_\ell & \text{else} \end{cases} \end{array} \right. , \tag{3}$$

for each $1 \leq j \leq d$. These operators provide an upper bound on the profile of a basis after a reduction at index j. To encode the behavior of a full round of reduction we define the operators:

$$\Delta_o = \prod_{i=1 \,|\, i \text{ odd}} \delta_i, \quad \text{and} \quad \Delta_e = \prod_{i=2 \,|\, i \text{ even}} \delta_i,$$

to define inductively the sequence:

$$\mu^{(1)} = \mu(M^{(1)})$$
$$\mu^{(i)} = \Delta_o\left(\mu^{(i-1)}\right) \quad \text{if } i = 0 \pmod 2 \quad \text{else} \quad \Delta_e\left(\mu^{(i-1)}\right)$$

Remark 2. *By the constraint we set on the volume of $\mathcal{M}$ to be equal to $2^{-d(d+1)(1+\varepsilon)\alpha n_h^2}$, we have for all $1 \leq i \leq \rho$, that $\mu_d^{(i)} = 0$.*

Proposition 2 (Exponential decay of $\|\mu^{(i)}\|_2$)**.** *For all odd i, we have,*

$$\left|\mu_1^{(i)}\right| \leq \mathrm{e}^{-\frac{\pi^2(i-1)}{2d^2}} \|\mu^{(1)}\|_2$$

and

$$\|\mu^{(i+1)}\|_2 \leq 2\mathrm{e}^{-\frac{\pi^2(i-1)}{2d^2}} \|\mu^{(1)}\|_2.$$

Proof. Note that $\Delta_o \circ \Delta_e$ depends only on the odd coordinates, so let Δ be its restriction to them in the domain and codomain. Remark that for all $1 \leq k \leq \lceil \frac{d-1}{2} \rceil$ the vector

$$\left(\sin\left(\frac{(2j-1)k\pi}{2\lfloor d/2 \rfloor}\right)\right)_j$$

is an eigenvector of Δ of associated eigenvalue $\cos\left(\frac{k\pi}{2\lfloor d/2 \rfloor}\right)^2$. A direct computation ensures that the eigenvectors are orthogonal. Since $2\lfloor d/2 \rfloor \leq d$, we use the trivial bound $\left|\cos\left(\frac{k\pi}{d}\right)\right| \leq \cos\left(\frac{\pi}{d}\right)$ in addition to the convexity bound

$$\ln(\cos(\pi/d)) < -\frac{\pi^2}{2d^2}$$

to obtain:

$$\sum_{k=1 \text{ odd}} \left(\mu^{(i)}\right)_k^2 \leq \mathrm{e}^{-\frac{\pi^2(i-1)}{2d^2}} \|\mu^{(1)}\|_2^2.$$

This implies the first statement and

$$\sum_{k=2 \text{ even}} \left(\mu^{(i+1)}\right)_k^2 \leq \sum_{k=1 \text{ odd}} \left(\mu^{(i)}\right)_k^2$$

implies the second. ∎

Remark 3 (A "physical" interpretation of Δ)**.** *The operator Δ introduced in the proof of proposition 2 acts as a* discretized *Laplacian operator on the discrete space indexed by $\{1, \ldots, d\}$, for a metric where two consecutive integers are at distance 1. Then, the action of Δ through the iterations $1 \leq i \leq \rho$ are reminiscent of the diffusion property of the solution of the* heat equation *($\frac{\partial u}{\partial t} = \alpha \Delta u$), whose characteristic time is quadratic in the diameter of the space.*

4.3.3. *A computational heuristic.* We now relate the behavior of the sequences of μ to the values taken by $R^{(i)}$. In order to do so, we introduce a computational heuristic on the behavior of the **Lift** function, asserting that the lifting phase does not blow up the size of the reduced vectors.

Heuristic 1 (Size of lifting)**.** *For any $1 \leq i \leq \rho$ and any $1 \leq j \leq d$ where a call to **Lift** happened:*

$$\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i+1)}\right) \leq \min\left(2^{\alpha n_h^2}\sqrt{\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}R_{j+1,j+1}^{(i)}\right)}, \mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}\right)\right).$$

A discussion on the validity of this heuristic is done in section 7.3. However, we recall that we *do not* perform a local reduction if the following condition is fulfilled, up to the approximation error due to the representation at finite precision[6]:

$$\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R_{j,j}^{(i)}) \leq \min\left(2^{(1+\varepsilon)\alpha n_h^2}\sqrt{\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}R_{j+1,j+1}^{(i)}\right)}, \mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}\right)\right).$$

From heuristic 1 we can show by a direct induction on i that the sequence of $\mu^{(i)}$ is an over-approximation of the flag profile at step i. More precisely we have:

Lemma 3. *Under heuristic 1, for any $1 \leq i \leq \rho$:*

$$\mu\left(M^{(i)}\right) \leq \mu^{(i)},$$

where the comparison on vectors is taken coefficient-wise.

4.3.4. *A bound on the approximation factor and number of rounds.* We can now conclude this paragraph by giving a quasiquadratic bound on the number of rounds:

Theorem 3. *Assuming that ρ is even and $\rho > \frac{2d^2}{\pi^2}\ln((2+\epsilon)\alpha d^2 n_h p)$, we have that*

$$\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R_{1,1}^{(\rho+1)}) \leq 2^{(d-1)(1+\varepsilon)\alpha n_h^2+1}|\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(\det M)|^{\frac{1}{d}}.$$

Proof. By taking the exponential of both sides of the inequality of lemma 3, we have:

$$\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R_{1,1}^{(\rho+1)}) \leq 2^{\mu_1^{(\rho+1)}-2(1+\varepsilon)\alpha}.$$

Recall that we forced $\left|\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(\det M)\right|^{\frac{1}{d}} = 2^{-(d+1)\alpha n_h^2(1+\varepsilon)}$, so that:

$$\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R_{1,1}^{(\rho+1)}) \leq 2^{(d-1)(1+\varepsilon)\alpha n_h^2+\mu_1^{(\rho+1)}}\left|\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(\det M)\right|^{\frac{1}{d}}.$$

[6]More precisely, if the precision used when performing this testing is p, then if we are certain that

$$\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R_{j,j}^{(i)}) \geq \min\left(2^{(1+\varepsilon)\alpha n_h^2}\sqrt{\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}R_{j+1,j+1}^{(i)}\right)}, \mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}\right)\right),$$

no local reduction is called, else we have

$$\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R_{j,j}^{(i)}) \geq \min\left(2^{(1+\varepsilon)\alpha n_h^2}\sqrt{\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}R_{j+1,j+1}^{(i)}\right)}, \mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}\left(R_{j,j}^{(i)}\right)\right)(1-2^{-\Omega(p)})$$

and a recursive local reduction is called, the multiplicative error term coming from the approximation error committed by the approximation of the values $R_{*,*}$ at precision p.

By proposition 2, we know that $\mu_1^{(\rho+1)} \leq e^{-\frac{\pi^2\rho}{2d^2}} \|\mu^{(1)}\|_2$. Since we have:

$$\begin{aligned}\ln|\mu_1^{(\rho+1)}| &\leq \ln\|\mu^{(1)}\|_2 - \frac{\rho\pi^2}{2d^2} \\ &\leq \ln((2+\epsilon)\alpha d^2 n_h p) - \frac{\rho\pi^2}{2d^2} \leq \quad 0,\end{aligned}$$

using lemma 2 and the hypothesis on ρ together with the fact that $d > 1$. All in all $|\mu_1^{(\rho)}| \leq 1$ and which entails the desired inequality. ■

With mild assumptions on the relative size of the parameters α, n_h, d and p we have the following rewriting of theorem 3.

Corollary 1. *Suppose that $\alpha = \log^{O(1)}(n_h)$ and that $p > n_h + d$, then taking $\rho = O\big(d^2\log(p)\big)$ is sufficient to reduce the module $\mathcal{M}$ and such that the algebraic norm of the first vector is bounded by a*

$$2^{\tilde{O}(dn_h^2)}|\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(\det M)|^{\frac{1}{d}}.$$

Remark 4. *If the caller makes a similar heuristic with a α', then we need $\alpha' > \alpha \cdot 2(1+\epsilon)\frac{d-1}{d}$ and any such value is plausible for large n_h.*

4.4. **Time complexity of the toplevel reduction.** Now that we have an estimate of the number of rounds, we can aim at bounding the complexity of each round, without counting the recursive calls, in a first time. To do so we will look independently at each of the part of a round, namely at the complexity of **Orthogonalize**, **Reduce** and **Lift**. Since the lifting algorithm performs a size-reduction, we first give a fine-grained look at the **Size-Reduce** function.

4.4.1. *Complexity and quality of **Size-Reduce**.* The quantitative behavior of the **Size-Reduce** procedure is encoded by the following theorem, given in all generality for arbitrary matrices over a cyclotomic field.

Theorem 4. *Let A be a matrix of dimension d whose coefficients lie in the cyclotomic field $\mathbf{K} = \mathbf{Q}[\zeta_f]$, and $n = \varphi(f)$. We are given a non-negative integer $p > 0$, where $\|A\|, \|A^{-1}\| \leq 2^p$ and such that $\sqrt{n\log n\log\log n} + d\log n < p$. By calling the algorithm **Orthogonalize** and **Size-Reduce**, we can find in time*

$$O\left(d^2 np\left(1 + \frac{d}{\log p}\right)\right)$$

an integral triangular matrix $U \in \left(\mathcal{O}_{\mathbf{K}}^{\times}\right)^{n\times n}$, such that $\|U\| \leq 2^{O(p)}$, and a matrix $R + E$, such that $\|E\| \leq 2^{-p}$, with R being the R-factor of the QR decomposition of AU and

$$\kappa(AU) \leq \left(\frac{\max_i \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}{\min_i \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}\right)^{\frac{1}{n}} 2^{O\left(\sqrt{n\log n\log\log n} + d\log n\right)},$$

for $\kappa(X) = \|X\|\|X^{-1}\|$ being the condition number of X.

Proof. See appendix A. ■

Corollary 2. *Suppose that:*

$$\|M^{(0)}\|, \|{M^{(0)}}^{-1}\| \leq 2^p \qquad \textit{and} \qquad d\log n_h + \sqrt{n_h \log n_h \log\log n_h} < p.$$

Then, we have the following bound on the condition number of $M^{(i)}$, valid for any loop index $1 \leq i \leq \rho$:

$$\kappa\Big(M^{(i)}\Big) \leq 2^{2p+O\left(\sqrt{n_h \log n_h \log\log n_h}+d\log n_h\right)},$$

and the call of the procedure ***Size-Reduce*** *at this i-th round has complexity*

$$O\left(d^2 n_h p\left(1+\frac{d}{\log p}\right)\right)$$

and requires a $O(p)$ of precision

Proof. We first remark that for any $1 \leq j \leq d$, the map $i \mapsto \max_j \mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}\Big(R^{(i)}_{j,j}\Big)$ is non-increasing, and therefore that $i \mapsto \min_j \mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}\Big(R^{(i)}_{j,j}\Big)$ is non-decreasing.

Now, theorem 1 implies that the Archimedean embeddings are balanced so that we have for all i:

$$\frac{\max_{\sigma:\mathbf{K}_h\to\mathbf{C}, R^{(i)}_{j,j}\in R^{(i)}}\left|\sigma\left(R^{(i)}_{j,j}\right)\right|}{\min_{\sigma:\mathbf{K}_h\to\mathbf{C}, R^{(i)}_{j,j}\in R^{(i)}}\left|\sigma\left(R^{(i)}_{j,j}\right)\right|} \leq 2^{2p+\mathrm{O}\left(\sqrt{n_h \log n_h \log\log n_h}\right)},$$

and so that

$$\frac{\max_j \mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R_{j,j})}{\min_i \mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R_{j,j})} = 2^{n_h\left(2p+\mathrm{O}\left(\sqrt{n_h \log n_h \log\log n_h}\right)\right)}.$$

Therefore, by combining this bound with the result of theorem 4, after the call to **Size-Reduce**, the condition number of $M^{(i)}$ is bounded by

$$2^{2p+\mathrm{O}\left(\sqrt{n_h \log n_h \log\log n_h}+d\log n_h\right)}$$

and the computation requires a $\mathrm{O}(p)$ bits of precision, with error bounded by 2^{-p}. ∎

4.4.2. *Complexity of the* ***Lift*** *procedure.* With the bounds given by theorem 4 we are now able to bound the complexity of the lift procedure described in algorithm 4.

Lemma 4 (Quasilinearity of **Lift**)**.** *Let $\mathbf{K}$ be the cyclotomic field of conductor $f > 0$, of dimension $n = \varphi(f)$. Denote by r the largest prime factor of f. Let $a, b \in \mathcal{O}_\mathbf{K}$ and suppose that:*

$$\gcd(\mathcal{N}_{\mathbf{K}/\mathbf{Q}}(a), \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(b)) = 1 \qquad \textit{and} \qquad \|a\| + \|b\| \leq 2^p.$$

Then, the time complexity of the algorithm ***G-Euclide*** *on the inuput (a, b) is a*

$$O(r\log(r) np \log p)$$

for $p \geq \sqrt{n \log n \log\log n}$. *Consequently, it is quasilinear for* $r \leq \log n$. *The output* (u, v) *verify:*

$$au + bv = 1 \qquad \textit{and} \qquad \|u\| + \|v\| \leq 2^{p+O\left(\sqrt{n \log n \log\log n}\right)}.$$

Proof. We use a tower of number fields[7] $\mathbf{L}_h^{\uparrow}$, where $\mathbf{L}_i = \mathbf{Q}[x]/\Phi_{f_i}(x)$ and $f_i/f_{i+1} \leq r$. By trivial induction and multiplicativity of the relative norm map, we know that the input of the recursive call at level i, that is, in $\mathbf{L}_i$ is $\mathcal{N}_{\mathbf{L}_h/\mathbf{L}_i}(a), \mathcal{N}_{\mathbf{L}_h/\mathbf{L}_i}(b)$. As such, with p_i being the number of bits of the coefficients of the input at level i of the recursion, we have $n_i p_i = \mathrm{O}(n_h p)$. Since computing the automorphisms corresponds to permutation of evaluation of a polynomial, each norm can be computed in time $\mathrm{O}(r \log(r) n_i p_i)$ using a product tree [35].

Now, we have by induction that $1 = \det W = \det V$. With R being the R-part of the QR-decomposition of V we have at any level i in the tower $\mathbf{L}_h^{\uparrow}$:

$$\|R_{2,2}\| = \|1/R_{1,1}\| \leq 2^{\mathrm{O}\left(\sqrt{n_i \log n_i \log\log n_i}\right)},$$

so that the size-reduction implies that

$$\begin{aligned}\|M\| &\leq \mathcal{N}_{\mathbf{L}_i/\mathbf{Q}}(R_{1,1})^{\frac{1}{n_i}} 2^{\mathrm{O}\left(\sqrt{n_i \log n_i \log\log n_i}\right)} \\ &= \left(n_h \|a\| + n_h \|b\|\right)^{\frac{n_h}{n_i}} 2^{\mathrm{O}\left(\sqrt{n_i \log n_i \log\log n_i}\right)}.\end{aligned}$$

Hence, the output coefficients are also stored using $\mathrm{O}(n_h p/n_i)$ bits. The complexity when $n_0 = 1$, i.e. the **ExGcd** base case, is classically in $\mathrm{O}(p_0 \log p_0)$. Summing along all complexities gives:

$$\mathrm{O}\left(n_h p \log(n_h p) + \sum_{i=1}^{h} r \log(r) n_i p\right) = \mathrm{O}(n_h p \log p + r \log(r) n_h p \log n_h)$$

which simplifies to a $\mathrm{O}(r \log(r) n p \log p)$. ∎

4.4.3. *Complexity of the top-level.* Now that we have analyzed the complexity and the output quality of each "atomic" parts, we can examine the complexity of the top-level of the algorithm **Reduce**—that is to say its complexity without counting the recursive calls.

Proposition 3. *Suppose that the following conditions are fulfilled:*

$$\min_{\sigma:\mathbf{K}_h \to \mathbf{C}, R_{i,i}^{(1)} \in R^{(1)}} \left|\sigma(R_{i,i}^{(1)})\right| \geq 2^{-p}, \quad \alpha = \log^{O(1)}(n_h)$$

$$d \log n_h + \sqrt{n_h \log n_h \log\log n_h} < p.$$

Then, the complexity at the top-level of the algorithm is a $O\left(d^5 n_h p \log p\right)$.

Proof. **Base case:** $\mathbf{K}_h = \mathbf{Q}$**:** This is a consequence of the analysis of Schönhage's fast reduction [46].

[7] Note that this tower is not same as the one used in the whole reduction process. The two towers are indeed constructed independently to optimize the global running time.

General case: Using corollary 1, the number of rounds is $\rho = \mathrm{O}(d^2 \log p)$. By lemma 4 the complexity of **Lift** is quasilinear. Thus, the complexity of each round is dominated by the computation of the QR decomposition and the size-reduction. By theorem 4, this complexity is a $\mathrm{O}(d^3 n_h p/\log p + d^2 n_h p)$, yielding a global complexity of $\mathrm{O}(d^5 n_h p + d^4 n_h p \log p) = \mathrm{O}(d^5 n_h p \log p)$. ■

4.4.4. *Bounding the precision at each level.* We now bound the precision used in the recursive calls at the top-level of the **Reduce** algorithm:

Lemma 5. *The sum of all bit sizes used in the recursive calls at the top-level is $O(d^2 p)$, when subjected to the conditions:*

$$\min_{\sigma:\mathbf{K}_h \to \mathbf{C}, R^{(1)}_{i,i} \in R^{(1)}} \left|\sigma(R^{(1)}_{i,i})\right| \geq 2^{-p} \qquad d \log n_h + \sqrt{n_h \log n_h \log\log n_h} < p.$$

Proof. Recall that the potential of the basis is defined as

$$\Pi = \sum_{j=1}^{d} (d-j) \log(\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R_{j,j})),$$

which is in $\mathrm{O}(n_h d^2 p)$ by assumption on p. Let $1 \leq j \leq d$, then the reduction algorithm is about to perform a local reduction of the projected sublattice (r_j, r'_{j+1}), as presented in section 3.1.4, two cases can occur:

- Either $\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R^{(i)}_{j,j}) \leq \min\left(2^{\alpha n_h^2}\sqrt{\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}\left(R^{(i)}_{j,j} R^{(i)}_{j+1,j+1}\right)}, \mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}\left(R^{(i)}_{j,j}\right)\right)$, and as mentioned in section 4.3.3 the local reduction is not performed. We can consider that we use here a zero precision call.
- Either a local reduction is actually performed and by the result of appendix A.3.1, we can use a precision in $\mathrm{O}(p_{i,j})$ with:

  $$p_i = \log\left(\frac{\max_k \sigma_k(R^{(i)}_{j,j})}{\min_k \sigma_k(R^{(i)}_{j+1,j+1})}\right)$$

  to represent the projected lattice. Let now set

  $$L = \frac{\log(\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R^{(i)}_{j,j}/R^{(i)}_{j+1,j+1}))}{n_h}.$$

  The precision $p_{i,j}$ is, thanks to the unit rounding theorem 1 a

  $$\mathrm{O}\left(L + \sqrt{n_h \log n_h \log\log n_h}\right) = \mathrm{O}(L),$$

  by hypothesis. The reduction of this truncated matrix yields a unimodular transformation, represented with precision $\mathrm{O}(p_{i,j})$, which when applied to the actual basis matrix implies that Π decreases by a term at least:

  $$\delta_{i,j} = n_h \left[\frac{L}{2} - \alpha n_h\right] - 2^{-\Omega(p)}$$

by heuristic 1 and theorem 11. Let us bound the ratio $p_{i,j}/\delta_{i,j}$:

$$\frac{p_i}{\delta_i} = \frac{L + \mathrm{O}\big(\sqrt{n_h \log n_h \log\log n_h}\big)}{\big(\frac{L}{2} - \alpha n_h\big) n_h - 2^{-\Omega(p_{i,j})}} = \frac{1 + \mathrm{O}\Big(\frac{\sqrt{n_h \log n_h \log\log n_h}}{L}\Big)}{\frac{n_h}{2} - \frac{\alpha n_h^2}{L} - \frac{2^{-\Omega(p_{i,j})}}{2L}}.$$

Now recall that $\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R^{(i)}_{j,j}) \geq 2^{2(1+\varepsilon)\alpha n_h^2} \mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R^{(i)}_{j+1,j+1})(1-2^{-\Omega(p_{i,j})})$, the multiplicative error term coming from the precision at which the values of the $R^{(i)}_{j,j}$ and $R^{(i)}_{j+1,j+1}$ are approximated at runtime. Thus, we have:

$$\sqrt{n_h \log n_h \log\log n_h}/L = \mathrm{O}\left(\sqrt{\frac{\log n_h \log\log n_h}{n_h}}\right),$$

and

$$\alpha n_h^2/L \leq \frac{n_h}{2(1+\varepsilon)}.$$

As such we have:

$$\frac{p_{i,j}}{\delta_{i,j}} \leq \frac{1 + \mathrm{O}\Big(\sqrt{\frac{\log n_h \log\log n_h}{n_h}}\Big)}{\frac{n_h \varepsilon}{1+\varepsilon} + o(1)}.$$

But then, $\delta_{i,j} = \Omega(n_h \varepsilon p_{i,j})$.

The potential is always a sum of non-negative terms, so $\sum_{i,j} \delta_{i,j} \leq \Pi$. The sum of the precision for the calls can thus be bounded by $\mathrm{O}\Big(\frac{\varepsilon}{(1+\epsilon)} \frac{\Pi}{n_h}\Big) = \mathrm{O}\big(d^2 p\big)$, since $\varepsilon = \frac{1}{2}$, which concludes the proof. ∎

Eventually we can prove the general complexity of the algorithm:

Proof of theorem 2. The first step of the proof consists in selecting a suitable tower of subfields, for which the relative degrees are chosen to optimize the complexity of the whole reduction. We choose a tower of cyclotomic subfields $\mathbf{K}_h^{\uparrow} = (\mathbf{Q} = \mathbf{K}_0 \subset \mathbf{K}_1 \subset \cdots \subset \mathbf{K}_h)$ with $[\mathbf{K}_i : \mathbf{Q}] = n_i$ and $n_{i+1}/n_i = r_i$ which satisfies $r_i/n_{i+1}^{1/5} \in [1; \log f]$, so that $h = \mathrm{O}(\log\log n)$. This always exists as f is log-smooth. We can set $\alpha_i = 4^{h-i+1}$ to satisfy the conditions of lemma 5 while making heuristic 1 practically possible. By definition of the value set for p we have $p = \mathrm{O}(B)$. And it of course satisfies the requirements of proposition 3. Note that by the choices of local precision made in the proof lemma 5, a simple induction shows that at each level of the recursion the local precision fulfills the condition of lemma 5, by the exact choice of the $p_{i,j}$'s. A by product of this induction asserts that the sum of the precision used in *all* the recursive calls needed to reduce a projected lattice at level i is a

$$\mathrm{O}\left(p \prod_{j=1}^{i-1} \mathrm{O}\big(r_j^2\big)\right) = 2^{\mathrm{O}(i)} B \left(\frac{n}{n_i}\right)^2.$$

Then, since by proposition 3 the complexity of the top-level call at level i is a $\mathrm{O}\big(r_i^5 n_i p \log(p)\big) = \mathrm{O}\big(r_i^5 n_i B \log(B)\big)$. Hence the total complexity at level i is

$r_i^5/m_i \cdot n^2 B \log(Bn) 2^{O(i)} = n^2 B \log(B) \log^{\mathrm{O}(1)} n$. Summing over all the levels retrieves the announced result. ■

An important point is that all recursive calls can be computed in parallel, and as most of the complexity is in the leaves, this leads to an important practical speed-up. We conjecture that when the number of processors is at most $n/\log^{\mathrm{O}(1)} n$, the speed-up is linear.

5. A fast reduction algorithm for high-rank lattices

While the previous reduction was tailored to reduce small (typically rank 2) rank lattices over cyclotomic fields, we now turn to the reduction of high rank lattices. It runs roughly in a constant number of matrix multiplications. It can also be used in the previous algorithm at each step to reduce the hidden logarithmic powers; but is of course interesting on its own for reducing rational lattices.

A bottleneck with algorithm 5 is that each round needs a matrix multiplication, and there are at least d^2 rounds. However, one can notice that each round only make local modifications. As a result, we propose to use a small number D of blocks, typically 4 or 8 suffices, and a round will (recursively) reduce consecutive pairs of dimension d/D. The resulting number of rounds is again $\mathrm{O}\big(D^2 \log B\big)$, giving a top-level complexity of $\mathrm{O}\big(D^2\big)$ (equivalent) multiplications. The corresponding algorithm is given in algorithm 6. In addition, the naive **Size-Reduce** procedure is replaced by a variant of Seysen reduction, which is detailed in appendix A.4. The complexity analysis is exactly the same as in the previous section, with the flag profile defined with respect to the volume of the blocks instead of simply the vectors, that is:

$$\mu(M)_j = \sum_{k=1}^{jd/D} \big(\log |\mathcal{N}_{\mathbf{K}/\mathbf{Q}}(R_{k,k})| + 2k(1+\varepsilon)\alpha n_h^2\big), \qquad \text{for } 1 \le j \le d.$$

We describe the algorithm with respect to an oracle **Oracle** which computes the base case. One can either use **Schonhage**, the algorithms in the previous or current section, or a recursive call.

Algorithm 6 — Reduce

Input : Basis $M \in \mathcal{O}_{\mathbf{K}}^{d\times d}$ of the $\mathcal{O}_{\mathbf{K}}-$module $\mathcal{M}$
Output : A unimodular transformation $U \in \mathcal{O}_{\mathbf{K}}^{d\times d}$ representing a reduced basis of $\mathcal{M}$.

if $d = 2$ **then return Oracle**(M)
for $i = 1$ **to** ρ **do**
  $R \leftarrow$ **Orthogonalize**(M)
  $U_i \leftarrow$ **Seysen-Size-Reduce**(R)
  $(M, R) \leftarrow (M, R) \cdot U_i$
  for $j = 1 + (i \mod 2)$ **to** d **by step of** $2d/D$ **do**
    $V_1 \leftarrow \operatorname{vol} R[j : j + d/D - 1, j : j + d/D - 1]$
    $V_2 \leftarrow \operatorname{vol} R[j + d/D : j + 2d/D - 1, j + d/D : j + 2d/D - 1]$
    if $V_1 \leq 2^{2(1+\varepsilon)\alpha n_h^2 d/D} V_2$ **then**
      $U' \leftarrow$ **Reduce**$(R[j : j + 2d/D - 1, j : j + 2d/D - 1])$
      $(U_i, M) \leftarrow (U_i, M) \cdot \mathrm{Diag}(\mathrm{Id}_j, U', \mathrm{Id}_{2d-j-2})$
    end if
  end for
end for
return $\prod_{i=1}^{\rho} U_i$ // The product is computed from the end

The analysis by Neumaier-Stehlé [39] only bounded the number of rounds, and as a result the complexity is $d^3 B^{1+o(1)}$. One can remark that even the simple algorithm uses $\Omega(d^3 \log B)$ local reductions, so that significantly decreasing their complexity can only come from a reduced precision in this local operation.

We, on the other hand, make the following heuristic:

Heuristic 2. *At any point in the recursion, when reducing a lattice of rank d, if we use a precision of $p \geq (1+\epsilon)\alpha d n$ then we decrease the potential Π by $\Omega(d^2 p)$.*

It is justified by the fact that the $\mathcal{N}_{\mathbf{K}/\mathbf{Q}}(R_{i,i})$ usually decrease roughly exponentially in i both in the input and the output matrices.

We need one last heuristic, which removes a $\log B$ factor:

Heuristic 3. *The number of bits needed decreases exponentially quickly, at the same speed as the μ vector.*

Indeed, a standard assumption for random lattices is that the upper-bound in heuristic 1 is in fact an approximation. As a result, we expect that lemma 3 holds with the vectors replaced by their forward differences, which implies this heuristic. The same property also implies the previous heuristic, as the forward difference of the eigenvector corresponding to the largest eigenvalue is a cosine.

Theorem 5. *Let A be a matrix of dimension d with entries in $\mathbf{K}$, with $\kappa(A) \leq 2^B$ such that $B \geq \sqrt{n \log n \log\log n} + \log n \log d$, n being the degree of $\mathbf{K}$ over $\mathbf{Q}$. Given A and an oracle which obeys heuristic 1, our reduction algorithm finds an*

integer vector x with

$$\|Ax\| \leq 2^{2(1+\epsilon+o(1))\alpha dn} \operatorname{vol}^{1/nd} A,$$

with α and ϵ defined as in the heuristic 2. Further, the sum of the precision used in the oracle calls is $O\big(d^2 p\big)$ and the heuristic *running time is*

$$O\left(\frac{d^\omega}{(\omega-2)^2} n \cdot B/\log B + d^2 nB \log^2 d\right)$$

for any constant ϵ.

Proof. Let r_i be the rank of the matrix at the i-th recursive level (one is the top). We use $w = \lfloor \log(B) \rfloor$. Then, using our heuristic on the potential, the sum of the precision p used in this level is $\mathrm{O}\big((d/r_i)^2 B\big)$. Using the complexity results presented in appendix A.4 each call with precision $p \geq \log B$ has a running-time of

$$\mathrm{O}\left((r_{i+1}/r_i)^2 \left(\frac{r_i^\omega}{\omega-2} n \cdot p/\log B + r_i^2 np \log r_i\right)\right)$$

using heuristic 3 on the exponential decrease of the precision used. Thus, the complexity of the i-th level is

$$\mathrm{O}\left((r_{i+1}/r_i)^2 \left(d^2 r_i^{\omega-2} \frac{n}{\omega-2} \cdot B/\log B + d^2 nB \log r_i\right)\right).$$

If $p < \log B$, then $r_i n = \mathrm{O}(\log B)$ and the cost is bounded by

$$\mathrm{O}\left((r_{i+1}/r_i)^2 \left(\frac{r_i^\omega}{\omega-2} n + r_i^2 np \log r_i\right)\right)$$

which in total is at most

$$\mathrm{O}\left((r_{i+1}/r_i)^2 \left(d^2 r_i^{\omega-2} \frac{n}{\omega-2} \cdot \frac{B}{r_i n} + d^2 nB \log r_i\right)\right).$$

As $r_i^{\omega(r_i)-3}/(\omega(r_i)-2)$ is bounded, this is always negligible.

One possible instantiation is r_i/r_{i+1} bounded. We then get $\sum_i d^2 r_i^{\omega(r_i)-2} = \mathrm{O}\Big(\frac{d^\omega}{\omega-2}\Big)$.

This gives an algorithm which finds a transition matrix such that the first block has a low volume:

$$\prod_{i=1}^{r_1} |\mathcal{N}_{\mathbf{K}/\mathbf{Q}}(R_{i,i})|^{1/r_1} \leq 2^{2(1+\epsilon)\alpha(d-r_1)n^2} \operatorname{vol}^{1/d} A \tag{4}$$

One then recurses on the first block, which corresponds to taking the product of a family of formulas of the same shape as eq. ((4)) for which the $(d-r_1)$ is replaced by a $(r_i - r_{i+1})$. The results derives directly from a telescopic summation over the exponents. This recursion is done for a fraction of the global complexity. ■

We emphasize that *in practice*, the entire basis is reduced at the end of the algorithm.

If we instantiate on rational lattices, this gives:

Corollary 3. *Let A be a matrix of dimension d with entries in $\mathbf{Z}$, with $\kappa(A) \leq 2^B$ such that $B \geq d$. Given A and an oracle which obeys heuristic 1, our reduction algorithm finds an integer vector x with*

$$\|Ax\| \leq 2^{d/2} |\det A|^{1/d}.$$

Further, the heuristic *running time is*

$$O\left(\frac{d^\omega}{(\omega-2)^2} \cdot B/\log B + d^2 B \log B\right).$$

This is, up to the $1/(\omega-2)$ factor on the first term the complexity of QR-decomposition, so the algorithm is essentially optimal.

Almost always, the first term is dominant and one can use $r_i/r_{i+1} = (d/r_i)^{1/3}$. The number of levels is then only $\mathrm{O}(\log\log d)$, and the larger r_i/r_{i+1} makes the heuristics more plausible.

Once the matrix is LLL reduced, we can reduce it further with a BKZ algorithm. We can use the same recursive structure, but when the dimension is less than $\beta\log(\beta)$, we use a BKZ reduction. The total number of calls is $\mathrm{O}\big(d^3\log d\big)$ [18], and we also have an approximation factor of $\beta^{\mathrm{O}(d/\beta)}$. Hence, we can use $\beta = \Theta(\log(Bd^{\omega-3}))$, which for ω not too small is $\Omega(\log(d))$, without increasing the running time. This implies that we can remove a $\log d/\log\log d$ factor when solving vectorial knapsacks, such as the ones for polynomial factoring [50].

We can instantiate this algorithm on roughly triangular matrices, and show that for random ones, one can get a (heuristic) significant speed-up. These matrices are widespread, as it corresponds to "knapsack" problems or searching integer relations[8]. In particular, one can quickly search a putative minimal polynomial.

Theorem 6. *Let A be a "random" matrix of dimension with d columns, $O(d)$ rows and entries in $\mathcal{O}_\mathbf{K} = \mathbf{Z}[x]/\phi_f(x)$. We define $B \geq d^2 n$ such that*

$$\|A\| + \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(\mathrm{vol}\, C) \leq 2^B$$

for all matrices C whose columns are a subset of A. For R the R-factor of the QR-decomposition of A, we also assume that $\|R^{-1}\| \leq 2^{B/d}$ and $\|R_{i,j}\| \leq 2^{B/i}$ for all i, j. We also require that $A_{i,j} = 0$ for $i \geq O(j)$ with a uniform constant. We can find an integer vector x with

$$\|Ax\| \leq 2^{d\tilde{O}(n)} \mathrm{vol}^{1/nd} A.$$

The heuristic complexity is

$$O\left(\frac{d^{\omega-1}}{(\omega-2)^2} n \cdot B/\log B + dnB\log^2 d\right) + d\tilde{O}\big(n^2 B\big)$$

Proof. The algorithm consists in reducing the first $k = 2^i$ columns of A for successive powers of two until d. The result is stored in A_i. The volume of A_i is

[8]While PSLQ [13] also solves this problem on *real RAM machines*, this model is an extremely poor approximation of computers [45]. See [13, Section 2] for what can go wrong, e.g.

bounded by 2^B so heuristically we expect, and will assume that

$$\|A_i\|, \|A_i\|^{-1} = 2^{d\tilde{\mathrm{O}}(n)+\mathrm{O}\left(B/2^i\right)}$$

for all i. We also store $Q_i R_i$, the QR-decomposition of A_i, and R_i^{-1}. We now explain how to compute A_{i+1}. Let x be a column of A which is not in the span of A_i. In order to reduce its bit size, we replace it by $x - A_j \lfloor R_j^{-1} \overline{Q_j}^t x \rceil$ for increasing j. This reduces the size of the projection of x orthogonally to A_j to $2^{d\tilde{\mathrm{O}}(n)+\mathrm{O}\left(B/2^j\right)}$; and by assumption on the input matrix, this is also true for the part orthogonal to A_j. At the end of this process, the length of x is therefore at most $2^{d\tilde{\mathrm{O}}(n)+\mathrm{O}\left(B/2^i\right)}$. For efficiency, this reduction is computed on all d vectors at the same time. Now we concatenate to A_i all the reductions of the needed vectors, and use our lattice reduction algorithm on the R-factor of the QR-decomposition of this matrix.

We now show that this matrix is well-conditioned. This matrix is written as

$$\begin{pmatrix} R_i & W \\ 0 & Z \end{pmatrix}.$$

We remark that the reduction process did not change Z, so that $\|Z^{-1}\| \leq \|R^{-1}\| \leq 2^{B/d}$. The inverse is

$$\begin{pmatrix} R_i^{-1} & -R_i^{-1} W Z^{-1} \\ 0 & Z^{-1} \end{pmatrix}$$

so that its condition number is bounded by $2^{d\tilde{\mathrm{O}}(n)+\mathrm{O}\left(B/2^i\right)}$.

The lattice reduction calls cost in total

$$\mathrm{O}\left(\sum_{i=1}^{\log d} \frac{2^{\omega(2^i)i}}{(\omega-2)^2} \frac{B}{2^i \log B} + 2^{2i} nBi/2^i\right)$$

and the cost of the oracles are bounded using the fact that $\Pi = \mathrm{O}(dnB)$:

$$d\tilde{\mathrm{O}}\left(n^2 B\right).$$

The pre-reduction computed by the algorithm has a running time of:

$$\mathrm{O}\left(\sum_{i=1}^{\log d} \frac{2^{\omega(2^i)i}}{\omega-2} \cdot \frac{d}{2^i} \frac{B}{2^i \log B} + d2^i nB/2^i\right).$$

Summing these complexities leads to the announced result. ∎

One can check than knapsack matrices, or Hermite Normal Form matrices with decreasing round pivots verify the assumptions with a small B.

6. Symplectic reduction

6.1. **On symplectic spaces and symplectic groups.** In the following, we very briefly introduce the linear theory of symplectic geometry and establish all along this presentation the parallel between the Euclidean and Symplectic geometries.

6.1.1. *Definitions.* A *symplectic space* is a finite dimensional vector space E endowed it with an antisymmetric bilinear form $J : E \times E \to E$. We can define a natural orthogonality relation between vectors $x, y \in E$ as being $J(x, y) = 0$. The linear transformations of E letting the symplectic structure J invariant is a group, called the J-symplectic group (or symplectic group if the context makes J clear). This group plays a similar role to the *orthogonal group* for Euclidean spaces.

6.1.2. *Darboux bases.* However on the contrary to Euclidean spaces, a symplectic space does not possess an orthogonal basis, but instead a basis $e_1, \ldots, e_d, f_1, \ldots, f_d$, so that for any indices $i < j$ we have $J(e_i, e_j) = 0, J(f_i, f_j) = 0, J(e_i, f_j) = 0$ and $J(e_i, f_i) > 0$. It implies in particular that any symplectic space has even dimension. We demonstrated in section 2.4 that it is easy to transform any basis of a Euclidean space in an orthogonal basis. This iterative construction is easily adapted to the symplectic case.

6.1.3. *Symplectic lattice, size reduction.* We can now easily adapt the definition of a lattice to the symplectic setting:

Definition 6. *A* symplectic lattice Λ *is a finitely generated free* $\mathbf{Z}$*-module, endowed with a symplectic form* J *on the rational vector space* $\Lambda \otimes_{\mathbf{Z}} \mathbf{Q}$.

As mentioned in section 3.1.5, an important tool to reduce lattices is the *size-reduction* procedure, which can be viewed as a discretization of the Gram-Schmidt orthogonalization. It aims at reducing the size and the condition number of the lattice basis. When dealing with symplectic symmetries, we can also discretize the process to obtain a basis which is close to a Darboux basis.

As we generalized the lattice formalism to $\mathcal{O}_{\mathbf{K}}$-modules in number fields, we can generalize straightforwardly the notions of symplectic lattices to the algebraic context. Using the work presented in section 3, we aim at providing a fast reduction algorithm for $\mathcal{O}_{\mathbf{K}}$-modules using these symplectic considerations.

6.1.4. *Towards an improved algorithmic size-reduction.* The specificities of the symplectic symmetry and of the evoked symplectic size-reduction enable a faster algorithm.

Indeed, we will demonstrate that a local reduction within the first half of the matrix can be applied directly to the second half. This almost divides by two the overall complexity *at each descent*.

In the rest of this section, we generalize the work of Gama, Howgrave-Graham and Nguyen [14] on the use of symplectic symmetries lattices within the reduction process. In particular, we show that such techniques can be used for all towers of number fields, and instead of an overall constant factor improvement, we can gain a constant factor at each floor of the tower and then cumulate them. Lattice reduction algorithms hinge on the two following facts:

Size reduction: We can control the bit size without changing the Gram-Schmidt norms.

Local reduction: Any two consecutive Gram-Schmidt norms can be made similar.

We therefore have to show that these two parts can be done while preserving the symplectic property.

6.2. **J-Symplectic group and compatibility with extensions.** In all the following we fix an *arbitrary* tower of number fields

$$\mathbf{K}_h^{\uparrow} = (\mathbf{Q} = \mathbf{K}_0 \subset \mathbf{K}_1 \subset \cdots \subset \mathbf{K}_h).$$

For any $1 \leq i \leq h$ we denote by d_h the relative degree of $\mathbf{K}_h$ over $\mathbf{K}_{h-1}$. On any of these number fields, we can define a simple symplectic form, which derives from the determinant form:

Definition 7. *Let $\mathbf{K}$ be a field, and set J to be an antisymmetric bilinear form on $\mathbf{K}^2$. A matrix $M \in \mathbf{K}^{2\times 2}$ is said to be* J-symplectic *(or simply symplectic if there is no ambiguity on J) if it lets the form J invariant, that is if $J \circ M = J$.*

Let us instantiate this definition in one of the fields of the tower $\mathbf{K}_h^{\uparrow}$ on the 2×2-determinant form. Let J_h be the antisymmetric bilinear form on $\mathbf{K}_h^2$ which is given as the determinant of 2×2 matrices in $\mathbf{K}_h$, i.e.

$$J_h\left(\begin{pmatrix} x_0 \\ x_1 \end{pmatrix}, \begin{pmatrix} y_0 \\ y_1 \end{pmatrix}\right) = x_0 y_1 - x_1 y_0.$$

Remark 5. *In the presented case, M is J_h-symplectic iff* $\det M = 1$.

Notice that we can always scale a basis so that this condition is verified.

We descend the form J_h to $\mathbf{K}_{h-1}$ by composition with a non-trivial linear form $\mathbf{K}_h \to \mathbf{K}_{h-1}$, for instance by using the relative trace, that is $J'_h = \mathrm{tr}_{\mathbf{K}_h/\mathbf{K}_{h-1}} \circ J_h$. We then extend the definition of symplectism to $\mathbf{K}_{h-1}^{2d_h}$ by stating that a $2d_h \times 2d_h$ matrix M' is symplectic if it preserves the J'_h form, that is if $J'_h \circ M' = J'_h$. This construction is tailored to be compatible with the descent of a matrix to $\mathbf{K}_{h-1}$ in the following sense:

Lemma 6. *Let M be a 2×2 matrix over $\mathbf{K}_h$ which is J_h-symplectic, then its descent $M' \in \mathbf{K}_{h-1}^{2d_h \times 2d_h}$ is J'_h-symplectic.*

6.3. **Towards module transformations compatible with J-symplectism.** Before exposing the transformation matrices in our size-reduction process of symplectic lattices, we give an insight on these techniques coming from the Iwasawa decomposition of Lie groups.

6.3.1. *On the Iwasawa decomposition.* The *Iwasawa decomposition* is a factorization of any semisimple Lie group in three components, which generalizes the decomposition of $\mathrm{GL}(n, \mathbf{R})$ in the product KAN where $K = O(n, \mathbf{R})$ is the orthogonal group, A is the group of diagonal matrices with positive coefficients and N is the unipotent group consisting of upper triangular matrices with 1s on the diagonal. This decomposition of $\mathrm{GL}(n, \mathbf{R})$ arises directly from the Gram-Schmidt decomposition of any real matrix and extracting the diagonal of its R part. The

$J-$symplectic group defined here is a semisimple Lie group and thus is subject to Iwasawa decomposition. We aim at using an *effective* version of the Iwasawa decomposition. In order to compute effectively such a decomposition, we need to find a generating set of *elementary* transformations over bases, which generalizes the operators of transvections and swaps in the general linear case.

We start by treating a simpler case: the Kummer-like extensions. The general case is covered in appendix C.

6.3.2. *A simple case: Kummer-like extensions* $\mathbf{K}[X]/(X^{d_h}+a)$. We define R_{d_h} as the reverse diagonal of 1 in a square matrix of dimension d_h.

In this section, we use the notation A^s as a shorthand for $R_{d_h}A^TR_{d_h}$, which corresponds to the reflection across the antidiagonal, that is exchanging the coefficients $A_{i,j}$ with A_{d_h+1-i,d_h+1-j}. We proceed here by adapting the work of Sawyer [43]. Suppose that the defining polynomial of $\mathbf{K}_h/\mathbf{K}_{h-1}$ is $X^{d_h}+a$. Recall that J_h is the 2×2-determinant form over $\mathbf{K}_h^2$. We can compose it by the linear form

$$\left|\begin{array}{rcl}\mathbf{K}_h\cong\mathbf{K}_{h-1}[X]/(X^{d_h}+a) & \longrightarrow & \mathbf{K}_{h-1}\\ y & \longmapsto & \mathrm{tr}_{\mathbf{K}_h/\mathbf{K}_{h-1}}(\frac{Xy}{d_ha})\end{array}\right. ,$$

to construct the matrix J'_h, which now becomes

$$J'_h=\begin{pmatrix}0 & R_{d_h}\\ -R_{d_h} & 0\end{pmatrix}$$

in the power basis. In this particular setting we retrieve the instantiation of [14]. In particular:

Lemma 7. *Fix a basis of the symplectic space where the matrix corresponding to* J'_h *is* $\begin{pmatrix}0 & R_{d_h}\\ -R_{d_h} & 0\end{pmatrix}$. *Then, for any* M *a* J'_h*-symplectic matrix and* QR *its QR decomposition, both* Q *and* R *are* J'_h*-symplectic.*

Proof. Direct from the explicit Iwasawa decomposition given by [43]. ■

Lemma 8 (Elementary J'_h-symplectic matrices)**.**

- *For any* $A\in GL(d_h,\mathbf{K}_h)$,
$$\begin{pmatrix}A & 0\\ 0 & A^{-s}\end{pmatrix}$$
is J'_h*-symplectic.*
- *For any* $A\in GL(2,\mathbf{K}_h)$ *with* $\det A=1$ *the block matrix*
$$\begin{pmatrix}Id_{d_h-1} & 0 & 0\\ 0 & A & 0\\ 0 & 0 & Id_{d_h-1}\end{pmatrix}$$
is J'_h *symplectic.*

Proof. By direct computation. ■

We now turn to the shape of triangular J'_h symplectic matrices.

Lemma 9. *Block triangular symplectic matrices are exactly the matrices of the form*

$$\begin{pmatrix} A & AU \\ 0 & A^{-s} \end{pmatrix}$$

where $U = U^s$.

Proof. Let $M = \begin{pmatrix} A & U \\ 0 & B \end{pmatrix}$ a block triangular matrix. By lemma 8, the action of the block diagonal matrices $\begin{pmatrix} A & 0 \\ 0 & A^{-s} \end{pmatrix}$ by left multiplication preserves the J'_h-symplectic group, so that without loss of generality we can suppose that A is the identity matrix. Identifying the blocks of $M^T J'_h M = J'_h$ yields two relations:

- $R_{d_h} B = R_{d_h}$, entailing $B = \mathrm{Id}_{d_h}$,
- $B^T R_{d_h} U - U^T R_{d_h} B = 0$, so that $R_{d_h} U = U^T R_{d_h}$, and as such $U = U^s$.

■

6.3.3. *Size-reduction of a J'_h-symplectic matrix.* A direct consequence of lemma 8 is that the local reductions occurring during the reduction, that is swaps and transvections can preserve the J'_h-symplectism by using the corresponding previous constructions.

Consider X a J'_h-symplectic matrix, we want to efficiently *size-reduce* X using the symmetries existing by symplectism. Let first take the R part of the QR-decomposition of X and make appear the factors A and U as in lemma 9.

Then we can focus on the left-upper matrix A and size-reducing it into a matrix A'. Each elementary operations performed is also symmetrically performed on A^s to retrieve $(A')^s$. Eventually the size reduction is completed by dealing with the upper-right block, which is done by performing a global multiplication by

$$\begin{pmatrix} \mathrm{Id}_{d_h} & -\lfloor U \rceil \\ 0 & \mathrm{Id}_{d_h} \end{pmatrix}.$$

The corresponding algorithm is given in algorithm 7, and uses the "classical" **Size-Reduce** procedure as a subroutine. The recursive reduction algorithm using the symplectic structure is then the exact same algorithm as algorithm 5, where the size-reduction call of line 4 is replaced by **Symplectic-Size-Reduce**.

Algorithm 7 — Symplectic-Size-Reduce

Input : R-factor of the QR decomposition of a J'_h-symplectic matrix $M \in \mathcal{O}^{d\times d}_{\mathbf{K}_h}$
Output : A J'_h-symplectic unimodular transformation U representing the size-reduced basis obtained from M.

1 Set A, U such that $\begin{pmatrix} A & AU \\ 0 & A^{-s} \end{pmatrix} = R$
2 $V \leftarrow$ **Size-Reduce**(A)
3 **return** $\begin{pmatrix} V & -V\lfloor U \rceil \\ 0 & V^{-s} \end{pmatrix}$

The size reduction property on A' implies that both A' and A'^{-1} are small, and therefore it is easy to check that the same is true for the now reduced R' and of course for the corresponding size reduction of the matrix X itself.

This approach admits several algorithmic optimizations:

- Only the first half of the matrix R is actually needed to perform the computation since we can retrieve the other parts. Indeed, with the equation $QR = X$, R is upper triangular and it only depends on the first half of Q.
- Further, we compute only the part above the antidiagonal of AU. This is actually enough to compute the part above the antidiagonal of $A^{-1}(AU)$, which is persymmetric.
- An interesting implication is that since we need to compute only half of the QR decomposition, we need (roughly) only half the precision.

6.4. **Improved complexity.** We analyze the algorithm of the previous section with the size-reduction of section 6.3.3. Using lemma 8, we can use the transition matrix found after a reduction in the first half of the matrix to directly reduce the second half of the matrix. This means that in symplectic reduction, we have recursive calls only for the first d_h steps of the tour. These are the only modifications in our algorithm. It is clear that, during the entire algorithm, the matrix R is symplectic.

The notation used in this section are the same as in section 4, with the notable exception that we may use here a large ε—recall that it was fixed to $1/2$ in all of section 4. We also assume that $\alpha > \sqrt{\log n_h \log\log n_h}$ for the sake of simplicity. We use here the modified potential where we consider only the first half of the matrix:

$$\Pi = \sum_{i=1}^{d_h} (d_h + 1 - i) \log \mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R_{i,i}).$$

To complete the proof we need an *experimentally validated heuristic* on the repartition of the potential during the reduction.

Heuristic 4. *The potential* Π *is, at the end of* **Reduce**, *always larger than the potential of an orthogonal matrix with the same volume.*

Remark 6. *This heuristic hinges on the fact the sequence of $\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R_{i,i})$ is non-increasing, which is always the case in practice for random lattices.*

We now give a better bound on the increase in bit sizes, which is a refinement of lemma 5. The proof is done in the exact same manner.

Lemma 10. *Suppose the input matrix M is a descent of a 2×2 triangular matrix $\begin{pmatrix} u & v \\ 0 & w \end{pmatrix}$, where the diagonal elements have been balanced in the sense of theorem 1. Under heuristic 4, the sum of all bit sizes used in the recursive calls at the top-level is at most*

$$pd_h^2\left(1+\frac{1}{\varepsilon}\right)\left(\frac{1}{2}+\frac{1}{d_h}+O\left(\sqrt{\frac{\log n_h \log\log n_h}{n_h}}\right)\right),$$

with

$$p = \log \frac{\max_{\sigma:\mathbf{K}_h\to\mathbf{C}, R_{i,i}\in R} \sigma(R_{i,i})}{\min_{\sigma:\mathbf{K}_h\to\mathbf{C}, R_{i,i}\in R} \sigma(R_{i,i})} \geq n_h d_h,$$

where the σ runs over the possible field embeddings and the $R_{i,i}$ are the diagonal values of the R part of the QR-decomposition of M.

Proof. Without loss of generality, up to scaling, we can assume that

$$\mathcal{N}_{\mathbf{K}_{h+1}/\mathbf{Q}}(u)\mathcal{N}_{\mathbf{K}_{h+1}/\mathbf{Q}}(w) = \mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}\left(\prod_i R_{i,i}\right) = 1.$$

Therefore, with our choice of p, we have at the beginning

$$\|R_{i,i}\| \leq \|u\| \in 2^{p/2+\mathrm{O}\left(\sqrt{n_h d_h \log(n_h d_h)\log\log(n_h d_h)}\right)}.$$

Thus we have :

$$\begin{aligned}\Pi =& \frac{n_h d_h(d_h+1)}{4}\left(p+\mathrm{O}\left(\sqrt{n_h d_h \log(n_h d_h)\log\log(n_h d_h)}\right)\right)\\ =& \frac{n_h d_h(d_h+1)}{4} p\left(1+\mathrm{O}\left(\sqrt{\frac{\log n_h \log\log n_h}{n_h}}\right)\right),\end{aligned}$$

since by hypothesis, $p > n_h d_h$. And then by, heuristic 4, we have $\Pi \geq 0$ at the end of the calls. When performing local reductions, as in the proof of lemma 5, two cases can occur:

- Either $\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R^{(i)}_{j,j}) \leq 2^{2(1+\varepsilon)\alpha n_h^2}\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R^{(i)}_{j+1,j+1})$, and as mentioned in section 4.3.3 the local reduction is not performed, so that we can consider that we use here a zero precision call.
- Either a local reduction is actually performed and by the result of appendix A.3.1, we can use a precision in $\mathrm{O}(p_{i,j})$ with:

$$p_{i,j} = \log\left(\frac{\max_k \sigma_k(R^{(i)}_{j,j})}{\min_k \sigma_k(R^{(i)}_{j+1,j+1})}\right),$$

Let now set

$$L = \frac{\log(\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R^{(i)}_{j,j}/R^{(i)}_{j+1,j+1}))}{n_h}.$$

The value $p_{i,j}$ is, thanks to the unit rounding theorem 1 a

$$L + \mathrm{O}\Big(\sqrt{n_h \log n_h \log\log n_h}\Big),$$

by hypothesis. The reduction of this truncated matrix yields a unimodular transformation, represented with precision $\mathrm{O}(p_{i,j})$, which when applied to the actual basis matrix implies that Π decreases by a term at least:

$$\delta_{i,j} = n_h \left[\frac{L}{2} - \alpha n_h\right] - 2^{-\Omega(p)}$$

by heuristic 1 and theorem 11. Let us bound the ratio $p_{i,j}/\delta_{i,j}$:

$$\frac{p_i}{\delta_i} = \frac{L + \mathrm{O}\big(\sqrt{n_h \log n_h \log\log n_h}\big)}{\big(\frac{L}{2} - \alpha n_h\big) n_h - 2^{-\Omega(p)}} = \frac{1 + \frac{\mathrm{O}\left(\sqrt{n_h \log n_h \log\log n_h}\right)}{L}}{\frac{n_h}{2} - \frac{\alpha n_h^2}{L} - \frac{2^{-\Omega(p)}}{2L}}.$$

Now recall that $\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R^{(i)}_{j,j}) \geq 2^{2(1+\varepsilon)\alpha n_h^2} \mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R^{(i)}_{j+1,j+1})(1-2^{-\Omega(p)})$, the multiplicative error term coming from the precision at which the values of the $R^{(i)}_{j,j}$ and $R^{(i)}_{j+1,j+1}$ are approximated at runtime. Thus we have:

$$\sqrt{n_h \log n_h \log\log n_h}/L = \mathrm{O}\left(\sqrt{\frac{\log n_h \log\log n_h}{n_h}}\right),$$

and

$$\alpha n_h^2 / L \leq \frac{n_h}{2(1+\varepsilon)}.$$

As such we have:

$$\frac{p_{i,j}}{\delta_{i,j}} \leq \frac{1 + \mathrm{O}\left(\sqrt{\frac{\log n_h \log\log n_h}{n_h}}\right)}{\frac{n_h \varepsilon}{1+\varepsilon} + \mathrm{O}(1/n_h)}.$$

The sum of precisions is therefore multiplied by

$$d_h^2 \left(1 + \frac{1}{\varepsilon}\right)\left(\frac{1}{2} + \frac{1}{2d_h} + \mathrm{O}\left(\sqrt{\frac{\log n_h \log\log n_h}{n_h}}\right)\right),$$

which finishes the proof. ∎

We can now collect all the calls at each level to compute the global complexity, for refining theorem 2:

Theorem 7. *Select an integer f a power of $q = O(\log f)$ and let $n = \varphi(f)$. The complexity for reducing matrices M of dimension two over $\mathbf{L} = \mathbf{Q}[x]/\Phi_f(x)$ with B the number of bits in the input coefficients is heuristically*

$$\tilde{O}\left(n^{2+\frac{\log((1/2+1/2q)(1+1/\varepsilon))}{\log q}} B\right)$$

and the first column of the reduced matrix has coefficients bounded by

$$\exp\left(O\left(n^{1+\frac{\log((1+\varepsilon)\frac{2q-1}{q})}{\log q}}\right)\right)\left|\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(\det M)\right|^{\frac{1}{2n}}.$$

Proof. The proof is now exactly the same as for theorem 2. We select a tower of cyclotomic subfields $\mathbf{K}_h^{\uparrow}$ with $\mathbf{K}_0 = \mathbf{Q}$, $[\mathbf{K}_i : \mathbf{Q}] = n_i$, $n_{i+1}/n_i = d_i = q$ for $i < h$ and $\mathbf{K}_h = \mathbf{L}$ with $h = \log f / \log q$. According to remark 2, we can take

$$\alpha_i = \mathrm{O}\left(n_i\left((1+\varepsilon)\frac{2q-1}{q}\right)^i\right)$$

and all our previous assumptions are fulfilled.

The complexity at the level i is $\mathrm{O}\big(q^5 n_i p \log(Bn)\big)$ for precision p but the sum on the precision over all calls is a:

$$\mathrm{O}\left(B\prod_{j>i}\left(1+\frac{1}{\varepsilon}\right)\left(\frac{1}{2}+\frac{1}{2q}+\mathrm{O}\left(\sqrt{\frac{\log n_i \log\log n_i}{n_i}}\right)\right)d_j^2\right),$$

which simplifies in

$$\mathrm{O}\left(B\left(\frac{n}{n_i}\right)^2\left(\frac{(1+\frac{1}{\varepsilon})(q+1)}{2q}\right)^{h-i}\right).$$

Summing over all i gives the result. ∎

Selecting $\varepsilon = \log n$, and running the algorithm of section 3 on the output of the reduction analyzed in section 4 gives:

Corollary 4. *Select an integer f a power of $q = O(\log f)$ and let $n = \varphi(f)$. The complexity for reducing matrices M of dimension two over $\mathbf{L} = \mathbf{Q}[x]/\Phi_f(x)$ with B the number of bits in the input coefficients is heuristically*

$$\tilde{O}\left(n^{2+\frac{\log(1/2+1/2q)}{\log q}}B\right) + n^{O(\log\log n)}$$

and the first column of the reduced matrix has coefficients bounded by

$$2^{\tilde{O}(n)}\left|\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(\det M)\right|^{\frac{1}{2n}}.$$

Clearly, for $B = n^{\omega(1)}$, we can choose $\varepsilon = \omega(1)$ and get a running time of

$$n^{2+\frac{\log(1/2+1/2q)}{\log q}+o(1)}B.$$

We insist on the fact that for $B = n^x$, the above proof does not give an optimal running time. This running time is given in fig. 2. One can improve on the upper-bound by using a stronger (yet credible) heuristic on Π, having only one reduction on each round where $\mathcal{N}_{\mathbf{K}_h/\mathbf{Q}}(R_{i,i}/R_{i+1,i+1})$ is maximized, an adaptive ε and two different q used. Clearly, this algorithm can also be parallelized, but the maximum number of processor used is less than before.

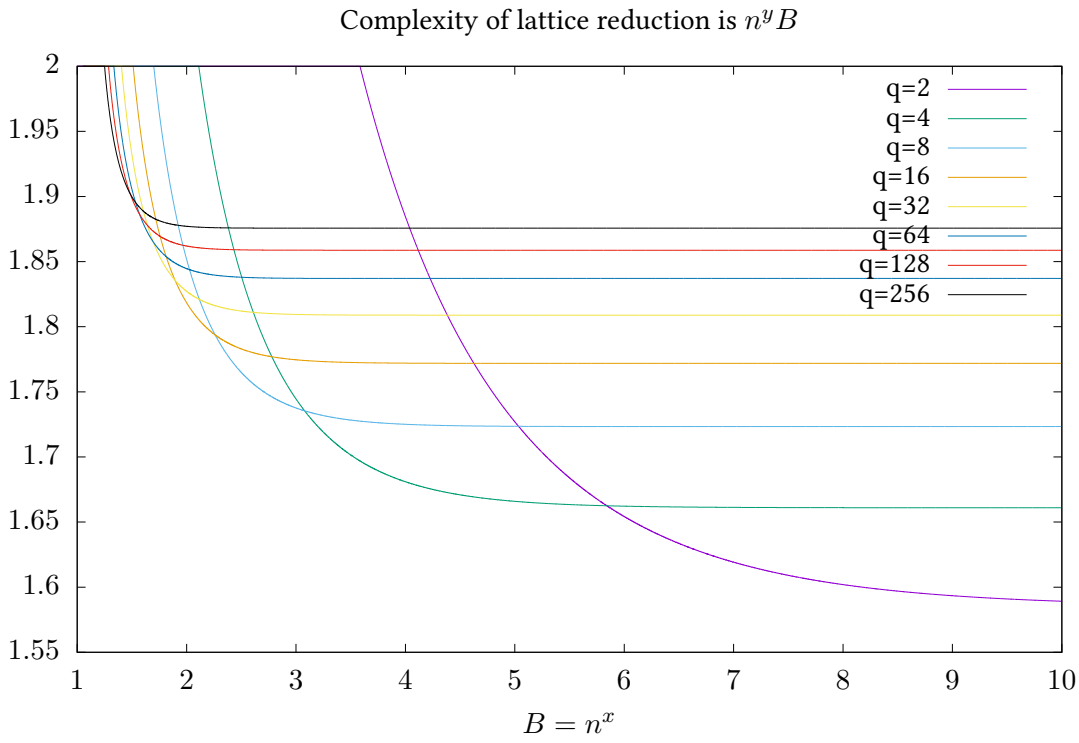


FIGURE 2. Upper bound on the complexity of symplectic reduction

7. Optimizations and Implementation

The algorithms detailed in section 3, section 5 and section 6 have been implemented and tested. This section details various optimizations and implementation choices, as well as gives an experimental assessment on the heuristics used in the complexity proofs.

The first algorithm of section 5 was used in the rational case to solve NTRU instances in a previous paper by Kirchner and Fouque [27], and found to perform as expected.

7.1. On the choice of the base case.

Let $h > 0$ be a non-negative integer. The setting of the reduction is a tower of power-of-two cyclotomic fields $\mathbf{K}_h^{\uparrow} = (\mathbf{Q} = \mathbf{K}_0 \subset \mathbf{K}_1 \subset \cdots \subset \mathbf{K}_h)$.

7.1.1. *Stopping the reduction before hitting* $\mathbf{Z}$.

As stated in theorem 2, the approximation factor increases quickly with the height of the tower. However, if we know how to perform a reduction over a number field above $\mathbf{Q}$, say $\mathbf{K}_1$ for instance, directly, then there is no need to reduce up to getting a $\mathbf{Z}$-module and we instead stop at this level. Actually, the largest the ring, the better the approximation factor becomes and the more efficient is the whole routine. It is well-known

TABLE 2. Lattice reduction with root factor α in dimension d over $\mathbf{Z}$ gives an element of Λ of norm around $\alpha^{d/2}\,\mathrm{vol}(\Lambda)^{1/d}$. After k steps in the Euclidean algorithm with norm factor β, the norm of the elements is roughly divided by β^k. Both are for random inputs.

Dimension	Root factor	Norm factor
1	1.031	4.6
2	1.036	7.1
4	1.037	17
8	1.049	26
16	1.11	24

that it is possible to come up with a *direct* reduction algorithm for an algebraic lattice when the underlying ring of integer is *norm-Euclidean*, as first mentioned by Napias in [37]. The reduction algorithm over such a ring $\mathcal{O}_{\mathbf{K}}$ can be done exactly as for the classical LLL algorithm, by replacing the norm over $\mathbf{Q}$ by the algebraic norm over $\mathbf{K}$. Hence a natural choice would be $\mathbf{Z}[x]/(x^n+1)$ with $n \leq 8$ as these rings are proved to be norm-Euclidean.

7.1.2. *The ring* $\mathbf{Z}[x]/(x^{16}+1)$. However, it turns out that while $\mathbf{K} = \mathbf{Z}[x]/(x^{16}+1)$ is not norm-Euclidean, we can still use this as our base case. As such, we need to slightly change the algorithm in case of failure of the standard algorithm. Given a, b, we use the *randomized* unit rounding of $\sqrt{\{\mu\}}$ computed by theorem 1 with $\mu = a/b$, which gives a unit u such that $u^2\{\mu\}$ is round. We accept the change if

$$\mathcal{N}_{\mathbf{K}/\mathbf{Q}}(a - b(\lfloor \mu \rceil + \lfloor u\{\mu\} \rceil u^{-1})) < \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(a)$$

and restart up to a hundred times if it fails.

This algorithm restarts on average 0.7 times and fails every 50000 times. On failure, one can for example use a more complicated approach; but as long as the number of bits is not gigantic, we can simply stop there since the other reductions around the two Gram-Schmidt norms will randomize everything and the algorithm can smoothly continue. The terms a, b tend to slowly accumulate a unit contribution when $n \geq 4$, and it is therefore needed to rebalance them using randomized rounding. For $n = 16$, this happens on average every 50 times.

7.1.3. *Comparison between the base fields.* We give in the table 2 the properties of the various possible base cases between the dimension 1 over $\mathbf{Q}$—that is $\mathbf{Q}$ itself—and 16, as described above.

Remark 7. *We need the base case to be (relatively) fast in our implementation. We followed the standard divide-and-conquer strategy: we first reduce the input matrix with half the precision, apply the transition matrix, and reduce the rest with about half the precision.*

7.2. **Decreasing the approximation factor.** In several applications, it is interesting to decrease the approximation factor. Our technique is, at the lowest level of recursion, and when the number of bits is low, to use a LLL-type algorithm. Each time the reduction is finished, we descend the matrix to a lower level where the approximation factor is lower.

Remark that the unit rounding is, at least theoretically, mandatory. In particular, a swap when the basis is not reduced with the definition in [25] may not lead to a reduction in potential so that the proof of [25, Theorem 3] is incorrect. We also point out that without a bound on the unit contributions, we have no polynomial bound on the number of bits used in their algorithm 3.

From a practical point of view, this does not seem to be a problem. If this is the case, our algorithm can be used every time we have a reasonable tower of number fields.

7.3. **Lifting a reduction.** One might expect that, as soon as the ideal generated by all the $\mathcal{N}_{\mathbf{L}/\mathbf{K}}(a_i)$ and $\mathcal{N}_{\mathbf{L}/\mathbf{K}}(b_i)$ is $\mathcal{O}_{\mathbf{K}}$, that for most of the small $x \in \mathcal{O}_{\mathbf{L}}$, we would have

$$\mathcal{N}_{\mathbf{L}/\mathbf{K}}(\langle a, x\rangle)\mathcal{O}_{\mathbf{K}} + \mathcal{N}_{\mathbf{L}/\mathbf{K}}(\langle b, x\rangle)\mathcal{O}_{\mathbf{K}} = \mathcal{O}_{\mathbf{K}}.$$

There is, however, a profusion of counterexamples to this and the algorithm often stumbles on them. This implies that the lift of a short vector can actually be quite large, depending on the norm of the ideal generated by the elements $\mathcal{N}_{\mathbf{L}/\mathbf{K}}(\langle a, x\rangle)$ and $\mathcal{N}_{\mathbf{L}/\mathbf{K}}(\langle b, x\rangle)$. A solution which practically works is to increase the number of short vectors we consider in the lifting phase: instead of lifting one vector, we lift multiple of them. As such, the lift step never causes problem when we are reducing a random lattice. In our experiments with random lattices, the average number of lifted vectors is around 1.5.

When the lattice is not random, for example with a short planted element, it sometimes completely fails: at each round in the algorithm, the lift will return a long vector even if the recursive reduction found plenty of short ones. While this may not be a problem for some applications – finding a short vector in a NTRU lattice implies an ability to decrypt – it is an important one for others. Our proposed solution to this difficulty is to use a pseudo-basis instead of a basis. Indeed, it is a standard fact that the first element can be lifted into a unimodular pseudo-basis [8, Corollary 1.3.5]. Of course, we need to have a fast ideal arithmetic and to keep the ideals of small norm, which is neither easy nor fast and will be the subject of a future work.

7.4. **Other details.** The program was written in the interpreted language `Pari/GP` [3]. It uses the native functions for multiplying field elements, which is not at all optimal, and even more so when we multiply matrices. Only the recursive calls were parallelized, and not the Gram-Schmidt orthogonalization nor the size reduction, which limits the speed-up we can achieve in this way. We used the Householder method for the QR decomposition. The symplectic optimization

was used at each step, and was not found to change the quality of the reduction[9]. We did not use the algorithm of section 5 inside the recursion of section 3. We chose a number of rounds of d^2 for all but the first level.

8. Applications

8.1. **Attacks on multilinear maps.** In 2013, a construction for cryptographic multilinear maps was announced [15] with a heuristic security claim. An implementation of an optimization of the scheme was later published [2]; however some of its uses, in particular involving an encoding of zero, were broken [23]. Subsequently, subfield attacks showed that the previous choice of parameters was unsafe [1, 7, 27], but these attacks were only asymptotical due to the extremely large dimension and length of the integers involved.

The improved scheme [2] gives encoding of the form $u_i = e_i/z \bmod q$ where $\|e_i\|$ is around

$$28\mathrm{e}N^4 \log(N)^{3/2} \sqrt{\pi \log(8N)}$$

in the ring $\mathbf{Z}[x]/(x^N + 1)$ with N a power of two. The attack, attributed to Galbraith, consists in computing $u_1/u_2 = e_1/e_2$ and recovering short vectors in

$$\begin{pmatrix} q & u_1/u_2 \\ 0 & \mathrm{Id}_N \end{pmatrix}$$

which is obviously solving a NTRU-like problem.

The present work revisits the results of the attacks presented in [27]: many instances can be broken even with a high approximation factor. A simple instance is with $N = 2^{16}$ and $q \approx 2^{6675}$, rated at the time at 56 bits of security [2, Table 1]. We compute the norm of e_1/e_2 over $\mathbf{Z}[x]/(x^n + 1)$ with $n = 2^{11}$ and solve the lattice problem over this smaller field. It took 13 core-days and 4 wall-time days to compute a solution. There are few running times of lattice reduction with high approximation factor on hard instances in the literature. It was reported in 2016 [1, Table 6] that the same problem with $n = 2^8$ and $q \approx 2^{240}$ takes 120 (single-threaded) hours with fplll [40]. As the complexity of their implementation is roughly proportional to $n^4 \log(q)^2$ we can estimate a running time of 40000 years, or 4000000 times slower than the algorithm presented in this work. This is the largest hard instance[10] of lattice reduction that we found in the literature.

[9] Gama, Howgrave-Graham and Nguyen [14] found instead that it gave a "smoother (better)" basis, showing a significant difference in their Figure 1. An other version of the paper does not include this comment, and their (perplexing) Figure 1 shows no difference in the exponential decrease of the Gram-Schmidt norms.

[10] There are easy instances with a larger dimension, for example in [14]. They considered a NTRU instance with degree 317 and modulus 128, and reduced it in 519 seconds. The low modulus implies that we only have to reduce the middle dimension 90 matrix, which fplll reduces in 0.2 second.

8.2. **Gentry-Szydlo algorithm.** The fast reduction procedure for cyclotomic ideals can be used to build a fast implementation of the Gentry-Szydlo algorithm [16]. This algorithm retrieves, in polynomial time, a generator of a principal ideal $f\mathcal{O}_{\mathbf{K}}$ given its relative norm $f\overline{f}$ in cyclotomic fields, or more generally in CM fields. This algorithm is a combination of algebraic manipulations of ideals in the field and lattice reduction.

8.2.1. *Gentry-Szydlo.* In this section, we briefly recall the crux of the Gentry-Szydlo algorithm [16]. This algorithm aims at solving the following problem, presented in its whole generality:

Problem (Principal ideal problem with known relative norm)**.** *Let $\mathbf{L}$ be a CM-field, of conjugation $x \mapsto \overline{x}$, and denote by $\mathbf{L}^+$ its maximal totally real subfield. Let $f \in \mathcal{O}_{\mathbf{L}}$ and set $\mathfrak{f} = f\mathcal{O}_{\mathbf{L}}$, the ideal spanned by this algebraic integer.*
Input: *The relative norm $\mathcal{N}_{\mathbf{L}^+/\mathbf{Q}}(f) = f\overline{f}$ and a $\mathbf{Z}$-basis of the ideal $\mathfrak{f}$.*
Output: *The element f.*

We can use the reduction of an ideal as follows: from $\mathfrak{f}$ and $f\overline{f}$ we start by reducing the $\mathcal{O}_{\mathbf{L}}$-lattice

$$\frac{f\mathcal{O}_{\mathbf{L}}}{\sqrt{f\overline{f}}},$$

of volume $\sqrt{|\Delta_{\mathbf{L}}|}$ and find an element of the shape fx where $x \in \mathcal{O}_{\mathbf{L}}$ and is small: $\|x\| = 2^{\tilde{\mathrm{O}}(n)}$. Now we have that:

$$\mathfrak{f} = \frac{f\overline{f}}{\overline{fx}} \cdot \overline{x}\mathcal{O}_{\mathbf{L}}$$

We also have $x\overline{x} = \frac{fx\overline{fx}}{f\overline{f}}$ so that we have reduced the problem to the smaller instance $(\overline{x}\mathcal{O}_{\mathbf{L}}, x\overline{x})$.

For the sake of simplicity, we give here the outline of the remaining part of the algorithm for a cyclotomic field of conductor a power of two. The algorithm selects an integer e such that $f^e \mod r$ is known with a large r. Binary exponentiation with the above reduction computes a $x\mathcal{O}_{\mathbf{L}}$ with a short $x \in \mathcal{O}_{\mathbf{L}}$ and such that

$$f^e = Px$$

with P known (and invertible) modulo r and q^k. Now we can deduce $x \bmod r$ and since x is small, we know x.

The last step is to extract an e-th root modulo q^k. We choose q such that $q\mathcal{O}_{\mathbf{L}} = \mathfrak{q}\overline{\mathfrak{q}}$ which always exists in power of two cyclotomic fields since $(\mathbf{Z}/2n\mathbf{Z})^\times/\{-1, 1\}$ is cyclic. Extracting e-th root modulo $\mathfrak{q}$ is easy, as e is smooth. There are $\gcd(e, q^{n/2}-1)$ such roots, and we can choose q such that for each $p|e$ with p not a Fermat prime, $q^{n/2} \neq 1 \bmod p$. If we choose $f \bmod \mathfrak{q}$ as a root, then we know $\overline{f} \bmod \overline{\mathfrak{q}}$, and we also know $f\overline{f}$ so we can deduce $f \bmod \overline{\mathfrak{q}}$. As a result, we know $f \bmod q$ and Hensel lifting leads to $f \bmod q^k$. For k sufficiently large, we recover f.

We choose e to be the smallest multiple of $2n$, such that r, the product of primes p such that $2n|p-1|e$, is sufficiently large. One can show [26] that $\log e =$

TABLE 3. Implementation results

Dimension	e	Running time	Processor
256	15360	30 minutes	Intel i7-8650 (4 cores)
512	79872	4 hours	Intel i7-8650 (4 cores)
1024	3194880	103 hours	Intel E5-2650 (16 cores)

$O(\log n \log\log n)$ is enough and heuristically taking e as the product of n and a primorial reaches this bound.

8.2.2. *Faster multiplication using lattice reduction.* The bottleneck of the Gentry-Szydlo algorithm is to accelerate the ideal arithmetic. We represent ideals with a small family of elements over the order of a subfield $\mathcal{O}_{\mathbf{K}}$. One can represent the product of two ideals using the family of all products of generators. However, this leads to a blow-up in the size of the family. A reasonable approach is simply to sample a bit more than $[\mathbf{L} : \mathbf{K}]$ random elements in the product so that with overwhelming probability the ideal generated by these elements is the product ideal itself. It then suffices to reduce the corresponding module to go back to a representation with smaller generators.

An important piece is then the reduction of an ideal itself. Our practical approach is here to reduce a square matrix of dimension $[\mathbf{L} : \mathbf{K}]$, and every two rounds to add a new random element with a small Gram-Schmidt norm in the ideal at the last position. With these techniques, the overall complexity of the Gentry-Szydlo now becomes a $\tilde{O}(n^3)$.

In our experiment, we reduce up to 1.05^n (respectively 1.1^n) the first ideal to accelerate the powering with $n \leq 512$ (respectively $n = 1024$). The smallest e such that this approximation works at the end was chosen. The other reductions are done with an approximation factor of $2^{n/5}$ (respectively $2^{n/3}$).

We emphasize that the implementation hardly used all cores: for example, the total running time over all cores in the last case was 354 hours.

The runtime of the first implementation published [4] in dimension 256 was 20 hours. Assuming it is proportional to n^6 leads to an estimate of 10 years for $n = 1024$, or 800 times slower than our algorithm. Our practical results are compiled in table 3.

There are applications in cryptography of this algorithm, such as when some lattice-based cryptography has a leak [16, 12, 1], for finding a generator of an ideal [4], for solving a norm equation [22] and for solving geometric problems on ideals [15, 26].

9. CONCLUSION

Through this article, we presented efficient LLL variants to reduce lattices defined over the ring of integers of cyclotomic fields, by exploiting the recursive

structure of tower of cyclotomic subfields. Our first algorithm has a complexity close to the number of swaps $\mathrm{O}\big(n^2 \cdot B\big)$ in LLL and the last one also exploits the symplectic symmetries naturally present in such towers. In this last case, we show that we can beat the natural lower bound on the number of swaps required to perform a reduction. One caveat of our algorithms is that their approximation factors are worse than the classical LLL approximation factor. However, such algorithms can be useful for some applications such as breaking graded encoding schemes or manipulating ideals, as in the Gentry-Szydlo algorithm. We implemented all our algorithms and their performances are close to the complexities that we proved under some mild assumptions. In particular, our implementation can use large base cases, that is all power of two cyclotomic fields of dimension ≤ 16.

This work raises several questions. First of all, on the need to rely on the introduced heuristics to prove the complexity. It is possible to remove them by using the pseudo-basis representation of modules over Dedekind rings, and will be the matter of a subsequent work. Second, we can wonder about the actual complexity of the symplectic algorithm for low bitsize and on the eventuality of decreasing the approximation factor: is it possible to recover the original LLL approximation factor while keeping the complexities of our fast variants? Third, our lattice reduction algorithm suggests that the algorithms for reducing lattices on polynomial rings may not be optimal [17], and in particular that an efficient algorithm with coarse-grain parallelism exists. Another interesting research direction is to design a faster reduction for lattices with a block-Toeplitz structure, which appear in Coppersmith's algorithm [9].

Finally, using the symplectic structure we can remark that we can halve the complexity of the DBKZ algorithm when the block size is less than n. We leave as an open problem the question of how to use similar techniques for larger gains.

Acknowledgement. We thank Bill Allombert for his help in the parallelization of the program.

References

[1] M. R. Albrecht, S. Bai, and L. Ducas. A subfield lattice attack on overstretched NTRU assumptions - cryptanalysis of some FHE and graded encoding schemes. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 153–178. Springer, Heidelberg, Aug. 2016.

[2] M. R. Albrecht, C. Cocis, F. Laguillaumie, and A. Langlois. Implementing candidate graded encoding schemes from ideal lattices. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 752–775. Springer, Heidelberg, Nov. / Dec. 2015.

[3] C. Batut, K. Belabas, D. Bernardi, H. Cohen, and M. Olivier. Pari-gp. *Avaliable from ftp://megrez. math. u-bordeaux. fr/pub/pari*, 1998.

[4] J.-F. Biasse, T. Espitau, P.-A. Fouque, A. Gélin, and P. Kirchner. Computing generator in cyclotomic integer rings - A subfield algorithm for the principal ideal problem in $\mathrm{L}_{|\Delta_{\mathbb{K}}|}(\frac{1}{2})$ and application to the cryptanalysis of a FHE scheme. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 60–88. Springer, Heidelberg, Apr. / May 2017.

[5] L. Bluestein. A linear filtering approach to the computation of discrete Fourier transform. *IEEE Transactions on Audio and Electroacoustics*, 18(4):451–455, 1970.

[6] N. Bourbaki. *Eléments de mathématique : Algèbre commutative: chapitres 1 à 4*. Eléments de mathématiques. Masson, 1985.
[7] J. H. Cheon, J. Jeong, and C. Lee. An Algorithm for NTRU Problems and Cryptanalysis of the GGH Multilinear Map without an encoding of zero. In *ANTS*, 2016. http://eprint.iacr.org/2016/139.
[8] H. Cohen. *Advanced topics in computational number theory*, volume 193. Springer Science & Business Media, 2012.
[9] D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *Journal of Cryptology*, 10(4):233–260, Sept. 1997.
[10] J.-S. Coron, T. Lepoint, and M. Tibouchi. Practical multilinear maps over the integers. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 476–493. Springer, Heidelberg, Aug. 2013.
[11] R. Cramer, L. Ducas, C. Peikert, and O. Regev. Recovering short generators of principal ideals in cyclotomic rings. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 559–585. Springer, Heidelberg, May 2016.
[12] T. Espitau, P.-A. Fouque, B. Gérard, and M. Tibouchi. Side-channel attacks on BLISS lattice-based signatures: Exploiting branch tracing against strongSwan and electromagnetic emanations in microcontrollers. In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *ACM CCS 2017*, pages 1857–1874. ACM Press, Oct. / Nov. 2017.
[13] H. R. Ferguson and D. H. Bailey. A polynomial time, numerically stable integer relation algorithm. Technical report, RNR Technical Report RNR-91-032, 1998.
[14] N. Gama, N. Howgrave-Graham, and P. Q. Nguyen. Symplectic lattice reduction and NTRU. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 233–253. Springer, Heidelberg, May / June 2006.
[15] S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 1–17. Springer, Heidelberg, May 2013.
[16] C. Gentry and M. Szydlo. Cryptanalysis of the revised NTRU signature scheme. In L. R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 299–320. Springer, Heidelberg, Apr. / May 2002.
[17] P. Giorgi, C.-P. Jeannerod, and G. Villard. On the complexity of polynomial matrix computations. In *Proceedings of the 2003 international symposium on Symbolic and algebraic computation*, pages 135–142. ACM, 2003.
[18] G. Hanrot, X. Pujol, and D. Stehlé. Analyzing blockwise lattice algorithms using dynamical systems. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 447–464. Springer, Heidelberg, Aug. 2011.
[19] C. Heckler and L. Thiele. Complexity analysis of a parallel lattice basis reduction algorithm. *SIAM Journal on Computing*, 27(5):1295–1302, 1998.
[20] N. J. Higham. *Accuracy and stability of numerical algorithms*, volume 80. Siam, 2002.
[21] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. In *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, pages 267–288, 1998.
[22] N. Howgrave-Graham and M. Szydlo. A Method to Solve Cyclotomic Norm Equations $f\overline{f}$. In *International Algorithmic Number Theory Symposium*, pages 272–279. Springer, 2004.
[23] Y. Hu and H. Jia. Cryptanalysis of GGH map. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 537–565. Springer, Heidelberg, May 2016.
[24] J.-P. Kahane. Local properties of functions in terms of random Fourier series. *Stud. Math*, 19:1–25, 1960.
[25] T. Kim and C. Lee. Lattice reductions over euclidean rings with applications to cryptanalysis. In M. O'Neill, editor, *16th IMA International Conference on Cryptography and Coding*, volume 10655 of *LNCS*, pages 371–391. Springer, Heidelberg, Dec. 2017.
[26] P. Kirchner. Algorithms on ideal over complex multiplication order. Cryptology ePrint Archive, Report 2016/220, 2016. http://eprint.iacr.org/2016/220.

[27] P. Kirchner and P.-A. Fouque. Revisiting lattice attacks on overstretched NTRU parameters. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 3–26. Springer, Heidelberg, Apr. / May 2017.
[28] R. Kučera. On bases of the Stickelberger ideal and of the group of circular units of a cyclotomic field. *Journal of Number Theory*, 40(3):284–316, 1992.
[29] E. Landau. Über Dirichletsche Reihen mit komplexen Charakteren. *Journal für die reine und angewandte Mathematik*, 157:26–32, 1927.
[30] A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 75(3):565–599, 2015.
[31] C. Lee, A. Pellet-Mary, D. Stehlé, and A. Wallet. An LLL algorithm for module lattices. *IACR Cryptology ePrint Archive*, 2019:1035, 2019.
[32] A. K. Lenstra, H. W. J. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:515–534, 1982.
[33] D. K. Maslen and D. N. Rockmore. Generalized FFTs—a survey of some recent results. In *Groups and Computation II*, volume 28, pages 183–287. American Mathematical Soc., 1997.
[34] K. Mehlhorn and P. Sanders. *Algorithms and data structures: The basic toolbox*. Springer Science & Business Media, 2008.
[35] R. Moenck and A. Borodin. Fast modular transforms via division. In *13th Annual Symposium on Switching and Automata Theory (swat 1972)*, pages 90–96. IEEE, 1972.
[36] T. Mukherjee and N. Stephens-Davidowitz. Lattice reduction for modules, or how to reduce modulesvp to modulesvp. Cryptology ePrint Archive, Report 2019/1142, 2019. https://eprint.iacr.org/2019/1142.
[37] H. Napias. A generalization of the LLL-algorithm over Euclidean rings or orders. *Journal de théorie des nombres de Bordeaux*, 8(2):387–396, 1996.
[38] J. Neukirch. *Algebraic Number Theory*. Springer, Germany, 1988.
[39] A. Neumaier and D. Stehlé. Faster LLL-type Reduction of Lattice Bases. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC*, pages 373–380, 2016.
[40] P. Q. Nguyen and D. Stehlé. Floating-point LLL revisited. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 215–233. Springer, Heidelberg, May 2005.
[41] C. Peikert. A Decade of Lattice Cryptography. *Foundations and Trends in Theoretical Computer Science*, 10(4):283–424, 2016.
[42] M. Pohst. A modification of the LLL reduction algorithm. *Journal of Symbolic Computation*, 4(1):123–127, 1987.
[43] P. Sawyer. Computing the Iwasawa decomposition of the classical Lie groups of noncompact type using the QR decomposition. *Linear Algebra and its Applications*, 493:573–579, 2016.
[44] C. Schnorr. A More Efficient algorithm for lattice basis reduction. *J. Algorithms*, 9(1):47–62, 1988.
[45] A. Schönhage. On the power of random access machines. In *International Colloquium on Automata, Languages, and Programming*, pages 520–529. Springer, 1979.
[46] A. Schönhage. Fast Reduction and Composition of Binary Quadratic Forms. In *Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation*, ISSAC '91, pages 128–133, New York, NY, USA, 1991. ACM.
[47] M. Seysen. Simultaneous reduction of a lattice basis and its reciprocal basis. *Combinatorica*, 13(3):363–376, 1993.
[48] J.-G. Sun. Perturbation bounds for the Cholesky and QR factorizations. *BIT Numerical Mathematics*, 31(2):341–352, 1991.
[49] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 24–43. Springer, Heidelberg, May / June 2010.
[50] M. Van Hoeij. Factoring polynomials and the knapsack problem. *Journal of Number theory*, 95(2):167–189, 2002.

[51] G. Villard. Parallel Lattice Basis Reduction. In *Proceedings of the 1992 International Symposium on Symbolic and Algebraic Computation, ISSAC '92, Berkeley, CA, USA, July 27-29, 1992*, pages 269–277, 1992.
[52] L. C. Washington. *Introduction to Cyclotomic Fields*. Springer, Germany, 1997.
[53] B. P. C. Wesolowski. Arithmetic and geometric structures in cryptography. Technical report, EPFL, 2018.

Appendix A. Bounding precision

In this section, we give details on the precision required in our algorithms. We first indicate the loss of precision of elementary operations, then look at the precision and complexity of the QR decomposition, and finally the size-reduction procedure. The last part indicates how to use fast matrix multiplication to reach the same goal. We recall that w is the number of bits in the words.

A.1. **Elementary operations.**

A.1.1. *Fast computation of primitive roots of unity.* The fast Fourier transform algorithm needs a precise approximation of the primitive roots of unity to be performed in fixed-point arithmetic. In order to compute with high precision a primitive f-th root of unity, one can use Newton's method where we start with $1 + 6.3i/f$. The following lemma ensures that the convergence, in this case, is at least quadratic.

Lemma 11. *Let $x \in \mathbf{C}$ such that $|x| \geq 1 - \frac{1}{2f}$, then by setting $x' = x - \frac{x^f - 1}{f x^{f-1}}$ and with $\zeta^f = 1$, we have:*

$$|x' - \zeta| \leq f|x - \zeta|^2$$

Proof. Without loss of generality, by dividing everything by ζ, we can assume $\zeta = 1$. We then have the following equality:

$$\frac{x' - 1}{(x-1)^2} = \frac{(f x^{f-1}(x-1) - x^f + 1)(x-1)^{-2}}{f x^{f-1}} = \frac{\sum_{k=1}^{f-1} k x^{k-1}}{f x^{f-1}}$$

Applying the triangular inequality gives:

$$\left| \frac{x' - 1}{(x-1)^2} \right| \leq \frac{f(f-1)\max(1, |x|^{f-1})}{2f|x|^{f-1}} \leq \frac{1}{2} f \max(1, |x|^{1-f}).$$

We can conclude by noticing that $(1 - \frac{1}{2f})^{-f} \leq (1 - 1/6)^{-3} < 2$. ∎

For $f \geq 128$, it is now easy to show that the sequence converges towards $\exp(2i\pi/f)$; the finite number of remaining cases are easily done by direct computations.

A.1.2. *A bound on the loss when iterating unitary matrices.* We now show the following elementary lemma on the iterations of matrix-vector computations, which states that the error made when computing chained matrix-vector multiplications can be controlled.

Lemma 12. *Let A_i be a family of k unitary matrices. Suppose that for each of these matrices A_i there exists an algorithm $\mathcal{A}_i$ that given some vector x, outputs $A_i x$ within a certain vector of errors e such that $\|e\| \leq \epsilon \|x\|$ with $\epsilon \leq \frac{1}{2k}$. Then, the algorithm which computes $(\prod_i A_i)x$ by composing the algorithms $\mathcal{A}_i$ returns $(\prod_i A_i)x$ within an error vector e such that $\|e\| \leq 2k\epsilon\|x\|$.*

Proof. Let $B = \prod_{i=2}^{k} A_i$ and $Bx + e'$ the error committed using the algorithms $\mathcal{A}_i$. The algorithm $\mathcal{A}_1$ outputs $A_1(Bx + e') + e$, so that the error committed towards $A_1 Bx$ is

$$\|A_1(Bx + e') + e - A_1 Bx\| \leq \|e'\| + \|e\| \leq \|e'\| + \epsilon\|Bx + e\|$$

We now prove by induction that this error is less than $\big((1+\epsilon)^k - 1\big)\|x\|$ with:

$$\begin{aligned}\|e'\| + \epsilon\|Bx + e\| \leq& \Big((1+\epsilon)^{k-1} - 1\Big)\|x\| + \epsilon\Big(\|x\| + \Big((1+\epsilon)^{k-1} - 1\Big)\|x\|\Big)\\ =& \Big((1+\epsilon)^k - 1\Big)\|x\|.\end{aligned}$$

The case $k = 1$ is immediate and $(1+\epsilon)^k - 1 < 2k\epsilon$ for $\epsilon < \frac{1}{2k}$ finishes the proof. ■

A.1.3. *Analysis of the Discrete Fourier transform.* We now show how to efficiently compute a close approximation of a Fourier transform. Indeed, the fast Fourier transform on 2^n points correspond to a product of n unitary matrices, so that we can get p bits of precision using a precision in $\mathrm{O}(p + \log n)$ by lemma 12. Using this, we obtain an algorithm to multiply integers with B bits with complexity $O(B/w \cdot \log(B/w)) = O(B)$.

Bluestein's algorithm [5] for Chirp-Z transform reduces discrete Fourier transform in any size to the computation of fast Fourier transform over power-of-two so that the same holds. Recall that Inverse Fourier transform can also be computed from a discrete Fourier transform.

All in all, we can evaluate the corresponding Fourier isomorphism and its inverse:

$$\mathbf{R}[x]/(\Phi_f) \cong \mathbf{C}^{\varphi(f)/2}$$

with limited loss in precision.

The complexity of this computation is a $\mathrm{O}(np + n\log n \cdot p/w) = \mathrm{O}(np)$ for $p = \Omega(w + \log n)$ with $n = \varphi(f)$. Indeed it breaks down as:

- Write the coefficients as polynomials with register-size coefficients and compute their Fourier transform with a cost of $\mathrm{O}(np)$
- Compute $\mathrm{O}(p/w)$ convolutions with Fourier transforms of size $\mathrm{O}(n)$
- Compute the inverse transform and propagate the carries for a running time of $\mathrm{O}(np)$.

(A modular implementation is probably faster if n is not tiny.)

In the general case, one would have to precompute the roots and use product and remainder trees [35].

A.2. **Householder orthogonalization.** The Householder orthogonalization algorithm transforms a complex matrix A into a product of QR, with Q unitary and R upper-triangular. Q is formed as a product of unitary reflections, which are all of the type $\mathrm{Id} - 2v\overline{v}^t$ for certain vectors $\|v\| = 1$.

The vector v corresponding to the first symmetry is chosen so that the first column of R has only its first coordinate to be non-zero. The algorithm then

applies this unitary operation to the matrix A and recursively orthogonalize the bottom-right of this new matrix.

More precisely, denote by a the first column of the matrix A. As such, the first column of R will be the vector

$$r = \left(-\|a\| \cdot \frac{a_1}{|a_1|}, 0, \dots, 0\right)^t,$$

with the quotient $\frac{a_1}{|a_1|}$ set to 1 if $a_1 = 0$. Then with $v = \frac{a-r}{\|a-r\|}$ and $Q = \mathrm{Id} - 2v\overline{v}^t$, we have that:

$$Qa = a - 2\frac{(a-r)\overline{(a-r)}^t a}{\|a-r\|^2} = a - \frac{2(\|a\|^2 - \overline{r}^t a)}{\|a-r\|^2}(a-r)$$

We now use the fact that $\overline{a}^t r \in \mathbf{R}$ and $\|r\| = \|a\|$ to get:

$$2(\|a\|^2 - \overline{r}^t a) = \|a\|^2 - \overline{r}^t a - \overline{a}^t r + \|r\|^2 = \|a-r\|^2$$

so that $Qa = r$.

The sign in the definition of r implies that $\|a - r\| \geq \|a\|$ so that we can compute v with the precision used to handle a.

If we use $p > \omega(\log d)$ bits of precision, we can multiply by $\mathrm{Id} - 2v\overline{v}^t$ with a relative error of $\mathrm{O}(d2^{-p})$. Using lemma 12, since we are performing d symmetries, each column is computed with a relative error of at most a $\mathrm{O}\big(d^2 2^{-p}\big)$. Hence, with $\hat{Q}$ the matrix output by the algorithm, each column of $\overline{Q}^t A$ has a relative error of $\mathrm{O}\big(d^2 2^{-p}\big)$ with respect to the computed R. This implies that there exists a matrix A' where each column is A within a relative error of $\mathrm{O}\big(d^2 2^{-p}\big)$, and whose R-factor in the QR decomposition is the returned R. Remark that the returned $R_{i,i}$ may not be real. While this is usually not a problem, R has to be multiplied on the left by a diagonal unitary matrix to obtain *the* QR-decomposition.

We define the conditional number of A as $\kappa(A) = \|A\|\|A^{-1}\|$. We can bound the stability of the QR decomposition [48]:

Theorem 8. *Given a matrix A, let R be the R-factor of its QR decomposition. For the matrix $A + \delta A$, let $R + E$ be the R-factor of its QR decomposition. Then:*

$$\|E\| \leq 3\kappa(A)\|\delta A\|$$

provided that $\kappa(A)\frac{\|\delta A\|}{\|A\|} < 1/10$.

Proof. Let $A = QR$ be the QR-decomposition. Without loss of generality, we assume $\|A\| = 1$. For a technical reason, we study the problem with δA a linear function where $\delta A(1)$ is the wanted matrix, which means that other quantities such as E are also functions.

We now obtain:

$$\overline{(A+\delta A)}^t(A+\delta A) = \overline{A}^t A + \overline{\delta A}^t A + \overline{A}^t \delta A + \overline{\delta A}^t \delta A$$

which is equal to:

$$\overline{(R+E)}^t(R+E) = \overline{R}^t R + \overline{E}^t R + \overline{R}^t E + \overline{E}^t E$$

so we deduce:

$$\overline{E}^t R + \overline{R}^t E + \overline{E}^t E = \overline{\delta A}^t A + \overline{A}^t \delta A + \overline{\delta A}^t \delta A.$$

We multiply by $\overline{A}^{-t}$ on the left and A^{-1} on the right:

$$\overline{A}^{-t}\overline{E}^t\overline{Q}^t + QEA^{-1} + \overline{A}^{-t}\overline{E}^t EA^{-1} = \overline{A}^{-t}\overline{\delta A}^t + \delta AA^{-1} + \overline{A}^{-t}\overline{\delta A}^t \delta AA^{-1}.$$

With $\rho = \|EA^{-1}\|$ and $\epsilon = \|\delta AA^{-1}\|$, we take the norm and get the inequality:

$$\rho - \rho^2 \leq 2\epsilon + \epsilon^2$$

so that for $\epsilon < 1/10$ we have $\rho \leq 3\epsilon$ if $\rho < 1/2$.

We now have to exclude the case $\rho > 1/2$, which we do with a topological argument. It is clear from the algorithm that the QR-decomposition is continuous over invertible matrices. Since

$$\|A^{-1}(A + \delta A(t)) - \mathrm{Id}\,\| \leq \|A^{-1}\|\|\delta A(t)\| < 1/2$$

for $0 \leq t \leq 1$, we have that $A + \delta A$ is invertible and therefore ρ is continuous over $[0; 1]$. As $\rho(0) = 0$ and $\rho([0; 1])$ is connex, we get $\rho(1) < 1/2$.

Finally, $\|E\| \leq \|EA^{-1}\|\|A\| = \rho$ gives the result. ■

Combining these results, we get:

Theorem 9. *Given a matrix A, we can compute the R-factor of its QR decomposition in time*

$$O\left(\frac{d^3 p}{w} + d^3 + d^2 p\right)$$

with a relative error of

$$O\big(\kappa(A) d^2 2^{-p}\big)$$

if this is smaller than a constant.

We can, of course, decrease the 3 in the exponent to a few matrix multiplications using aggregated Householder transformations and a divide-and-conquer algorithm, see [20, Subsection 18.4]. This is also at the end of the appendix.

A.3. **Size-reduction.** We first consider the size-reduction for unitriangular matrices (i.e. upper triangular matrices with ones on the diagonal). Such a matrix A is said to be size-reduced if both A and A^{-1} are small.

Lemma 13. *Let A be a unitriangular matrix of dimension d with coefficients in $\mathbf{K} = \mathbf{Q}[\zeta_f]$, such that its coefficients in the power basis are bounded in absolute value by 1. Then $\|A\| \leq dn^{3/2}$ and $\|A^{-1}\| = (2n)^{O(d)}$ with $n = \varphi(f)$.*

Proof. It is clear that $\|A_{i,j}\| \leq \sqrt{nf} \leq n^{3/2}$ so that $\|A\| \leq dn^{3/2}$. Now let x be a column of A^{-1}. Consider a i which maximizes $\|x_i\|(2n^{3/2})^i$. Then we have

$$1 \geq \|(Ax)_i\| \geq \|x_i\| - \sum_{j>i} \|A_{i,j}\|\|x_j\| \geq \|x_i\| \left(1 - \sum_{j>i} \frac{n^{3/2}}{(2n^{3/2})^{j-i}}\right) > \|x_i\|/3$$

and we obtain $\|x_i\| \leq 3$ which gives $\|x\| \leq 3(2n^{3/2})^{d-1}\sqrt{d}$. ■

We can finally prove our size-reduction theorem:

Theorem 10. *Let A be a matrix of dimension d with coefficients in $\mathbf{K} = \mathbf{Q}[\zeta_f]$, and $n = \varphi(f)$. We are given p, where $\|A\|, \|A^{-1}\| \leq 2^p$ and also $\sqrt{n \log n \log\log n} + d \log n < p$. In time $O\big(d^3 np/w + d^2 pn \log d\big)$, we can find an integral triangular matrix U with $U_{i,i} \in \mathcal{O}_{\mathbf{K}}^{\times}$ and a matrix $R + E$ such that $\|E\| \leq 2^{-p}$, with R the R-factor of the QR decomposition of AU and*

$$\kappa(AU) \leq \left(\frac{\max_i \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}{\min_i \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}\right)^{1/n} 2^{O\left(\sqrt{n \log n \log\log n} + d \log n\right)}.$$

We also have $\|U\| \in 2^{O(p)}$

Proof. In the canonical basis of $\mathbf{K}$ repeated d times, A corresponds to a $d \times d$ block matrix, where each block is a diagonal complex matrix of size $n/2 \times n/2$, so that the QR decomposition can be obtained from $n/2$ complex QR decompositions of dimension d. We can transform into (and from) this basis at a cost of $\mathrm{O}\big(d^2 pn\big)$; and the same technique can be used with the size-reduction algorithm.

The algorithm computes R', the R-factor of the QR decomposition of A. Then we use algorithm 3 on R' which returns a U, and the algorithm returns U and $R'U$.

We have that

$$\|AU\| \leq d \sum_i \|R_{i,i}\| \leq d^2 \|A\|$$

so that $\|U\| \leq \|A^{-1}\| \|AU\| \leq d^2 2^{2p}$. As a result, we can use a precision of $\mathrm{O}(p)$ bits.

Let D be the diagonal of R. We have $\kappa(AU) = \kappa(R) \leq \kappa(D)\kappa(D^{-1}R)$. The reduction with units guarantees that

$$\kappa(D) \leq \left(\frac{\max_i \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}{\min_i \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}\right)^{1/n} 2^{\mathrm{O}\left(\sqrt{n \log n \log\log n}\right)}.$$

The previous lemma gives $\kappa(D^{-1}R) = 2^{\mathrm{O}(d \log n)}$. ∎

A.3.1. *On the reduction of well-conditioned matrices.* We finish this subsection with properties of lattices represented by a well-conditioned matrix. The following easy theorem indicates that if we want to reduce the lattice generated by A, we can always truncate the matrix and work with precision only $\mathrm{O}(\log(\kappa(A)))$. The transition matrix which will be computed by the algorithm also needs at most this precision. Up to an irrelevant (small) quantity, this is of course a

$$\mathrm{O}\left(\log\left(\frac{\max_i \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}{\min_i \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}\right)/n\right).$$

Theorem 11. *Let A, δA and U an integer matrix such that $\|AU\| \leq \kappa \|A\|$, $\kappa(AU) \leq \kappa$ and*

$$\frac{\|\delta A\|}{\|A\|} \leq \frac{\epsilon}{3\kappa^3}$$

with $\epsilon < 1/4$ and $\kappa \geq \kappa(A)$. Let R be the R-factor of the QR-decomposition of AU and $R + E$ be the one of $(A + \delta A)U$. Then $\|U\| \leq \kappa^2$ and

$$\frac{\|E\|}{\|A\|} \leq \epsilon.$$

Proof. First $\|U\| \leq \|A^{-1}\|\|AU\| \leq \kappa\|A^{-1}\|\|A\| \leq \kappa^2$. Then $\|U\| \geq 1$ since it is integral so that $1 \leq \|A^{-1}AU\| \leq \|A^{-1}\|\|AU\|$ and $\|AU\| \geq \frac{1}{\|A^{-1}\|} = \frac{\kappa(A)}{\|A\|}$. We deduce:

$$\frac{\|\delta AU\|}{\|AU\|} \leq \frac{\epsilon}{3\kappa^2}$$

and applying the stability theorem we get:

$$\frac{\|E\|}{\|AU\|} \leq \frac{\epsilon}{\kappa}.$$

Using the lower bound on $\|AU\|$ finishes the proof. ∎

In all LLL algorithms, $\max_i \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(R_{i,i})$ is non-increasing with respect to the round number and $\min_i \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(R_{i,i})$ is non-decreasing so that we can use the theorem for all U where AU is size-reduced with

$$\kappa \leq \left(\frac{\max_i \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}{\min_i \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(R_{i,i})}\right)^{1/n} 2^{\mathrm{O}\left(\sqrt{n \log n \log\log n} + d \log n\right)}.$$

Heuristically, for random lattices, we have $\|U\| \lesssim \sqrt{\kappa(A)}$ and $\kappa(AU)$ depends only on the dimension so a truncation of the R-factor of the QR-decomposition of A with error roughly $\|A\|/\kappa(A)$ is enough. The precision needed is therefore on the order of $2\log(\kappa(A))$.

A.4. **Faster algorithms.** We explain here algorithms running in time essentially equal to a matrix multiplication for all previous tasks. They are only used in section 5. We represent a matrix of real numbers by a matrix of integers and a denominator which is a power of two. Multiplication of matrices, therefore, do not depend on *how* the multiplication is computed, as long as it is correct: whether the corresponding algorithm for floating-point inputs is stable or not is not relevant here.

The QR-decomposition works as follows. Given the matrix $\begin{pmatrix} A & B \end{pmatrix}$ with n columns and $m \geq n$ rows, we first recursively compute the QR-decomposition of $A = Q_1 R_1$. We let $\overline{Q_1}^t B = \begin{pmatrix} B_1' \\ B_2' \end{pmatrix}$ where B_1' has as many rows as there are columns in A. Then we compute the QR-decomposition of $B_2' = Q_2 R_2$. The QR-decomposition of the input is then

$$\left(Q_1 \begin{pmatrix} \mathrm{Id} & 0 \\ 0 & Q_2 \end{pmatrix}\right) \begin{pmatrix} R_1 & B_1' \\ 0 & R_2 \end{pmatrix}.$$

Remark that

$$(\mathrm{Id}+XY)(\mathrm{Id}+ZW) = \mathrm{Id}+XY+ZW+XYZW = \mathrm{Id}+\begin{pmatrix} X & Z+XYZ \end{pmatrix}\begin{pmatrix} Y \\ W \end{pmatrix}$$

so we represent all Q matrices in this way, and the base case is done as usual.

We now consider the complexity for a square matrix of dimension d. At the k-th recursive levels, the matrices have at most $d/2^k+1$ columns and d rows. There are $\mathrm{O}(2^k)$ rectangular matrix products to be computed, each can be computed in 2^k+1 products of square matrices of dimension $\leq d/2^k+1$. The total complexity with p bits of precision is

$$\mathrm{O}\left(\sum_{k=1}^{1+\log d} 2^{2k}(d/2^k)^{\omega(d/2^k)}p/w + d^2p\right) = \mathrm{O}\left(\frac{d^\omega}{\omega-2}p/w + d^2p\log d\right).$$

We can prove by induction for $p \geq \mathrm{O}(\log d)$ that for the Q computed $\hat{Q}$, and for the Q matrix if it were computed exactly[11] $\breve{Q}$, we have

$$\|\overline{\breve{Q}}^t \hat{Q} - \mathrm{Id}\| = d^{\mathrm{O}(1)}2^{-p}.$$

As a result, $\overline{\breve{Q}}^t$ times the input matrix is with a relative error of $d^{\mathrm{O}(1)}2^{-p}$ the computed R; so that the computed R corresponds to a QR-decomposition of the input matrix with a relative error in the input matrix of $d^{\mathrm{O}(1)}2^{-p}$.

We deduce using theorem 8:

Theorem 12. *Given a matrix A, we can compute the R-factor of its QR decomposition in time $O\left(\frac{d^\omega p}{(\omega-2)w} + d^2p\log d\right)$ for $p \geq w + \log(d\kappa(A))$ with a relative error of 2^{-p}.*

We now show a fast size-reduction. The best-known algorithm for minimizing the condition number of a unitriangular matrix was given by Seysen [47]. Using this approach replaces the $(2n)^{\mathrm{O}(d)}$ term by a $(2nd)^{\mathrm{O}(\log d)}$ in the final condition number. We explain Seysen's size-reduction as it is both easier and better than the standard one.

It works as follows. Given a matrix $\begin{pmatrix} A & B \\ 0 & C \end{pmatrix}$, we can assume that both A and C are size-reduced, after two recursive calls. We then multiply it by

$$\begin{pmatrix} \mathrm{Id} & -\lfloor A^{-1}B \rceil \\ 0 & \mathrm{Id} \end{pmatrix}$$

and return this matrix.

The result is thus

$$\begin{pmatrix} A & B - A\lfloor A^{-1}B \rceil \\ 0 & C \end{pmatrix}.$$

and the top-right part is not much larger than A^{-1}. The inverse of the result is

$$\begin{pmatrix} A^{-1} & -\left(A^{-1}B - \lfloor A^{-1}B \rceil\right)C^{-1} \\ 0 & C^{-1} \end{pmatrix}$$

and the top-right part is not much larger than C^{-1}.

[11] This matrix is computed from erroneous inputs so that it need not be the Q part of the QR decomposition.

We first study an algorithm to invert unitriangular matrices.

Algorithm 8 — Invert

Input : An unitriangular matrix M
Output : An approximation of M^{-1}

1 **if** *dimension=1* **then return** 1
2 $\begin{pmatrix} A & B \\ 0 & C \end{pmatrix} \leftarrow M$ // with a dimension almost halved
3 $A' \leftarrow \textbf{Invert}(A)$
4 $C' \leftarrow \textbf{Invert}(C)$
5 **return** $\begin{pmatrix} A' & -A'BC' \\ 0 & C' \end{pmatrix}$

We first prove the performances of the inversion algorithm:

Theorem 13. *Given a unitriangular matrix M of dimension d with coefficients in $\mathbf{K} = \mathbf{Q}[\zeta_f]$, a field of dimension n, with $\|M\|, \|M^{-1}\| \leq 2^p$ and $p \geq w + \log(nd)$, **Invert** returns a matrix M' such that $\|M' - M^{-1}\| \leq 2^{-p}$ with a running time of $O\big(d^\omega np/w + d^2np\big)$.*

Proof. We use a precision $p' = 1 + 2p + \lceil \log(d) \rceil = \mathrm{O}(p)$.

We prove that $\|M'^{-1} - M\| \leq 2d^{0.5}2^{-p'}$ by induction on d. The case $d = 1$ is easy, so we assume $d > 1$. Let E be such that the top-right part of M' is $-A'BC' + E$, and also $A'^{-1} = A + \delta A$, $B'^{-1} = B + \delta B$. Then, we have:

$$M'^{-1} - M = \begin{pmatrix} \delta A & -A'^{-1}EC'^{-1} \\ 0 & \delta C \end{pmatrix}.$$

We can guarantee $\|E\| \leq 2^{-p'-2p}$ with an intermediary bitsize $\mathrm{O}(p')$. This leads to our intermediary result.

Now let $M'^{-1} = M + F$. We get

$$M' = (M(\mathrm{Id} + M^{-1}F))^{-1} = (\mathrm{Id} + M^{-1}F)^{-1}M^{-1}$$

and therefore $\|M' - M^{-1}\| \leq \|M^{-1}\|\|(\mathrm{Id} + M^{-1}F)^{-1} - \mathrm{Id}\| \leq 2^{-p}$. ∎

Algorithm 9 — Seysen-Size-Reduce

Input : An unitriangular matrix M
Output : An integer unitriangular transformation U, and $(AU)^{-1}$

if *dimension=1* **then return** 1
$\begin{pmatrix} A & B \\ 0 & C \end{pmatrix} \leftarrow M$ // with a dimension almost halved
$U_1 \leftarrow$ **Seysen-Size-Reduce**(A)
$U_2 \leftarrow$ **Seysen-Size-Reduce**(C)
$A' \leftarrow$ **Invert**(AU_1)
$W \leftarrow \lfloor A'BU_2 \rceil$
return $\begin{pmatrix} U_1 & -U_1W \\ 0 & U_2 \end{pmatrix}$

We finally have:

Theorem 14. *Given a unitriangular matrix M of dimension d with coefficients in $\mathbf{K} = \mathbf{Q}[\zeta_f]$, a field of dimension n, with $\|M\|, \|M^{-1}\| \leq 2^p$ and $p \geq w + \log(nd)\log(d)$. Then **Seysen-Size-Reduce** returns an integer unitriangular matrix U with $\|U\| \leq 2^{O(p)}$ such that*

$$\|MU\|, \|(MU)^{-1}\| \leq (n^{3/2}d)^{\lceil \log d \rceil}$$

with a running time of $O\big(d^\omega np/w + d^2np\big)$.

Proof. We use a precision $p' = \mathrm{O}(p + \log(nd)\log(d)) = \mathrm{O}(p)$. We prove by induction on d that $\|MU\|, \|(MU)^{-1}\| \leq (n^{3/2}d)^{\lceil \log d \rceil}$. Initialization is clear, so we assume $d > 1$. We have that MU is

$$\begin{pmatrix} AU_1 & BU_2 - AU_1W \\ 0 & CU_2 \end{pmatrix}.$$

The top-right matrix is $AU_1((AU_1)^{-1}BU_2 - W)$ and we have, with $A' - (A_1U)^{-1} = 1 + \delta A$:

$$\|(AU_1)^{-1}BU_2 - W\| \leq \|\delta ABU_2\| + \|A'BU_2 - W\|.$$

The first term is bounded by $2^{\mathrm{O}(p)}\|\delta A\|$ and the second by $2dn^{3/2}/3$. We choose the precision so that the first term is at most $1/3$ and the result is proven, as $\|AU_1\|, \|CU_2\| \leq (n^{3/2}d)^{\lceil \log d \rceil - 1}$.

Next, the matrix $(MU)^{-1}$ is

$$\begin{pmatrix} (AU_1)^{-1} & -(AU_1)^{-1}(BU_2 - AU_1W)(CU_2)^{-1} \\ 0 & (CU_2)^{-1} \end{pmatrix}.$$

The top-right matrix is $((AU_1)^{-1}BU_2 - W)(CU_2)^{-1}$. The first term was already bounded above, so $\|(CU_2)^{-1}\| \leq (n^{3/2}d)^{\lceil \log d \rceil - 1}$ finishes the proof.

Finally, we have $\|U\| = \|M^{-1}MU\| \leq \|M^{-1}\|\|MU\| \leq 2^p(n^{3/2}d)^{\lceil \log d \rceil}$. ∎

Note that it is mandatory to have M well-conditioned if we want a U which is not much larger than M. This is also true for other variants of LLL (including fpLLL): outputting the transition matrix may lead to a slow-down by a factor of n.

Appendix B. Fast unit-rounding in cyclotomics fields

The goal of this section is to prove theorem 1. In particular we perform a novel analysis of the algorithm of [11] to obtain a faster running time and we extend their result for *arbitrary* cyclotomic fields.

B.1. **Prime power-case.**

As a starter, we prove that the techniques of [11] can be used for unit-rounding in prime-power cyclotomic fields with quasi-linear complexity. Formally we aim at proving the following:

Theorem 15. *Let $\mathbf{K}$ be the cyclotomic field of prime power conductor f. There is a quasi-linear randomized algorithm that given any element in $x \in (\mathbf{R} \otimes \mathbf{K})^\times$ finds a unit $u \in \mathcal{O}_{\mathbf{K}}^\times$ such that for any field embedding $\sigma : \mathbf{K} \to \mathbf{C}$ we have*

$$\sigma\big(xu^{-1}\big) = 2^{O\left(\sqrt{f \log f}\right)} \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(x)^{\frac{1}{\varphi(f)}}.$$

Compared to [11], there are two differences with the treatment proposed here: on the one hand we use fast arithmetic of the involved objects—namely Fourier-based multiplication in an abelian group-ring—and on the other hand we increase the success probability by using a better bound by the classical Berry-Esseen theorem, as it was hinted in their seventh footnote.

B.1.1. *Recall on the probability notions used in the proof.*

Before diving in the proof of theorem 15, let us recall the basis notions of probability theory we are using, namely subgaussians variables and the Berry-Esseen theorem.

On subgaussian random variables. The notion of subgaussian distribution goes back to the work of Kahane in [24], and encompasses a large family of real distributions with very convenient properties similar to the normal law.

Definition 8. *A real random variable X is said to be* τ-subgaussian *for some $\tau > 0$ if the following bound holds for all $s \in \mathbf{R}$:*

$$\mathbb{E}\big[\exp(sX)\big] \leq \exp\left(\frac{\tau^2 s^2}{2}\right). \tag{5}$$

A τ-subgaussian probability distribution is in an analogous manner.

Lemma 14. *A τ-subgaussian random variable X satisfies*

$$\mathbb{E}[X] = 0.$$

Proof. Follows from the Taylor expansion at 0 of $\mathbb{E}[\exp(sX)] = 1 + s\mathbb{E}[X] + \mathrm{O}\big(s^2\big)$. ∎

The main property of subgaussian distributions is that they satisfy a Gaussian-like tail bound.

Lemma 15. *Let X be a τ-subgaussian distribution. For all $t > 0$, we have*

$$\Pr[X > t] \leq \exp\Big(-\frac{t^2}{2\tau^2}\Big). \tag{6}$$

Proof. Fix $t > 0$. For all $s \in \mathbf{R}$ we have, by Markov's inequality:

$$\Pr[X > t] = \Pr[\exp(sX) > \exp(st)] \leq \frac{\mathbb{E}[\exp(sX)]}{\exp(st)}$$

since the exponential is positive. Using that X is τ-subgaussian, eq. ((5)) gives:

$$\Pr[X > t] \leq \exp\Big(\frac{s^2\tau^2}{2} - st\Big)$$

and the right-hand side is minimal for $s = t/\tau^2$, entailing the announced result. ∎

Many usual distributions over $\mathbf{Z}$ or $\mathbf{R}$ are subgaussian. This is in particular the case for distributions with finite supports and zero mean.

The Berry-Esseen approximation theorem. The Berry-Esseen theorem, or Berry-Esseen inequality, provides a quantitative estimate of the rate of convergence towards the normal distribution, as showing that the cumulative function (CDF) of the probability distribution of the scaled mean of a random sample converges to Φ at a rate inversely proportional to the square root of the number of samples. More formally we have:

Theorem 16. *There exists a positive $C < 0.5$ such that if $X_1, X_2, \cdots, X_n$ are independent and identically distributed random variables with zero mean, satisfying $\mathbb{E}(X_1^2) = \sigma^2 > 0$, $\mathbb{E}(|X_1|^3) = \rho$, and by setting*

$$Y_n = \frac{X_1 + X_2 + \cdots + X_n}{n}$$

the sample mean, with F_n the cumulative distribution function of $\frac{Y_n\sqrt{n}}{\sigma}$ and Φ the cumulative distribution function of the standard normal distribution, then for all x and n we have,

$$|F_n(x) - \Phi(x)| \leq \frac{C\rho}{\sigma^3\sqrt{n}}$$

B.1.2. *Going back on the rounding problem.* We now fix a cyclotomic field $\mathbf{K} = \mathbf{Q}[\zeta_f]$ with prime power-conductor f. We recall that in $\mathbf{K}$, the cyclotomic units are easily described:

Lemma 16 (Lemma 8.1 of [52])**.** *Let f be a prime power, then the group of cyclotomic units is generated by $\pm\zeta_f$ and $\frac{\zeta_f^\alpha - 1}{\zeta_f - 1}$ for $\alpha \in (\mathbf{Z}/f\mathbf{Z})^\times$.*

We first provide a convenient description of the cyclotomic units as an orbit of the element $\zeta_f - 1$ under the action of its Galois group.

B.1.3. *Log-embedding and action of* $(\mathbf{Z}/f\mathbf{Z})^\times/\{-1,+1\}$. Define the Log embedding to be the coefficient-wise composition of the real logarithm with the absolute value of the Archimedean embeddings:

$$\mathrm{Log} : \left| \begin{array}{ccc} \mathbf{K} & \longrightarrow & \mathbf{R}^{\frac{n}{2}} \\ \alpha & \longmapsto & [\log(|\sigma_i(\alpha)|)]_{i\in G} \end{array} \right. ,$$

where the embeddings are paired by conjugates and listed by the group $G = (\mathbf{Z}/f\mathbf{Z})^\times/\{-1,+1\}$. The image of the unit multiplicative group $\mathcal{O}_\mathbf{K}^\times$ is a full rank lattice by Dirichlet unit's theorem, and is called the *Log-unit lattice*.

We first remark that the group-ring $\mathbf{Z}[(\mathbf{Z}/f\mathbf{Z})^\times]$ acts on the group $(\mathbf{R}\otimes\mathbf{K})^\times$ in the following way: for any $g = \sum_\alpha g_\alpha \alpha \in \mathbf{Z}[(\mathbf{Z}/f\mathbf{Z})^\times]$ and $x \in (\mathbf{R}\otimes\mathbf{K})^\times$,

$$g \cdot x = \prod_{\alpha\in(\mathbf{Z}/f\mathbf{Z})^\times} \sigma_\alpha(x)^{g_\alpha},$$

where σ_α maps ζ_f to ζ_f^α. But σ_α acts as a permutation on the Archimedean embedding so that the embedding in the Log-unit lattice *commutes* with the action of $\mathbf{Z}[(\mathbf{Z}/f\mathbf{Z})^\times]$ in the following sense:

$$\mathrm{Log}(g\cdot x) = g\mathrm{Log}(x) \in \mathbf{R}[G],$$

for all $x \in (\mathbf{R}\otimes\mathbf{K})^\times$.

Henceforth, the cyclotomic units can be described using this action, as they correspond to the orbit of the element $\zeta_f - 1$ by the kernel, called the *augmentation ideal*, of $g \mapsto \sum_\alpha g_\alpha$:

$$\left\{ g\cdot(\zeta_f - 1) \mid \sum_\alpha g_\alpha = 0 \right\} \tag{7}$$

B.1.4. *An upper bound on the norm of* $\mathrm{Log}(\zeta_f - 1)$. We also have that $\mathrm{Log}(\zeta_f - 1)$ is invertible, and with a small inverse (for example $\|\mathrm{Log}(\zeta_f - 1)\| = \mathrm{O}(n^3)$) so that we can compute efficiently. Let us formalize this intuition. We first bound $\mathrm{Log}(\zeta_f - 1)$:

Lemma 17. *We have* $\|\mathrm{Log}(\zeta_f - 1)\|_\infty \le \log f$ *and* $\|\mathrm{Log}(\zeta_f - 1)\|_2 = O(\sqrt{f})$.

Proof. The coordinates are given by

$$\mathrm{Log}(\zeta_f - 1)_\alpha = \mathrm{Log}(|\zeta_f^\alpha - 1|) = \log(|2\sin(\pi\alpha/f)|),$$

for any $\alpha \in (\mathbf{Z}/f\mathbf{Z})^\times$. Now, for $0 \le x \le \frac{1}{2}$ and $\alpha \in (\mathbf{Z}/f\mathbf{Z})^\times$, we have $\sin(\pi x) \ge 2x$ and we can consider that $0 \le \frac{\alpha}{f} \le \frac{1}{2}$. We deduce that $\|\mathrm{Log}(\zeta_f - 1)\|_\infty \le \log\left(\frac{f}{4}\right)$ and

$$\|\mathrm{Log}(\zeta_f - 1)\|_2^2 \le \sum_\alpha \log^2\left(\frac{f}{4\alpha}\right) \le f\int_0^{\frac{1}{2}} \log^2\left(\frac{4}{x}\right)dx,$$

the latest integral being equal to $\frac{9}{2} + \frac{3}{\ln 2} + \frac{1}{\ln^2 2}$ entails the announced inequality. ∎

Remark 8. *The multiplication in the group ring $\mathbf{Z}[G]$ is quasi-linear as G is a finite abelian group. Indeed, we can use Fourier transform to reduce the multiplication to point-wise multiplications (see for instance* [33]*).*

B.1.5. *Fast rounding in the Log-unit lattice.* We can now describe the rounding algorithm, which essentially is a randomized coefficient-wise rounding using the orbital description of eq. ((7)).

Proof of theorem 15. Without loss of generality, we can assume $\mathcal{N}_{\mathbf{K}/\mathbf{Q}}(x) = 1$.

Then, using the description given by eq. ((7)) the problem is thus reduced to searching a unit u such that $\mathrm{Log}(u) \in \mathbf{Z}[G]$ which is close to $y = \frac{\mathrm{Log}(x)}{\mathrm{Log}(\zeta_f - 1)}$ and such that $\sum_\alpha \mathrm{Log}(u)_\alpha = 0$. The simplest idea consists in performing a coefficient wise rounding of the coefficients of the vector y. However, this approach does not succeed all the time, but we can take advantage of the two possible choices in the rounding to closest integers to randomize the rounding—that is to say, by randomizing the choice of floor or ceil instead of relying deterministically on the round function $\lfloor \cdot \rceil$.

Formally, for $\alpha \neq 1$, we sample z_α following the unique distribution on the two elements set $\{\lfloor y_\alpha \rfloor, \lceil y_\alpha \rceil\}$ with expectation y_α. Then, z_1 is set at $-\sum_{\alpha \neq 1} z_\alpha$ to ensure $\sum_\alpha z_\alpha = 0$. Clearly, $u = z \cdot (\zeta_f - 1)$ verifies our requirements if

$$\|\mathrm{Log}(\zeta_f - 1)(y - z)\|_\infty = \mathrm{O}\Big(\sqrt{f \log f}\Big).$$

The Berry-Esseen theorem indicates that $|y_1 - z_1| \leq \sqrt{n}/\log n$ with probability $\Theta(1/\log n)$. The coordinates of

$$\mathrm{Log}(\zeta_f - 1)(y - z - (y - z)_1 \sigma_1)$$

are subgaussians of parameter $\|\mathrm{Log}(\zeta_f - 1)\|_2$. Therefore, using the estimation of lemma 17, we know that their absolute values can all be bounded by $\mathrm{O}\big(\sqrt{f \log f}\big)$ except with probability at most $\Theta\Big(\frac{1}{\log^2 f}\Big)$. Hence, our requirement is fulfilled with probability $\Omega\Big(\frac{1}{\log n}\Big)$. We have $\mathrm{Log}(u) = z\mathrm{Log}(\zeta_f - 1)$ which can be computed in quasi linear time. Eventually a Fourier transform recovers $\sqrt{u\bar{u}}$, which is u up to an irrelevant torsion[12]. ∎

B.2. **Extension to arbitrary cyclotomic fields.** We now extend the result of theorem 15 to arbitrary cyclotomic fields, that is proving:

Theorem 17. *Let $\mathbf{K}$ be the cyclotomic field of conductor f. There is a quasi-linear randomized algorithm that given any element in $x \in (\mathbf{R} \otimes \mathbf{K})^\times$ finds a unit $u \in \mathcal{O}_\mathbf{K}^\times$ such that for any field embedding $\sigma : \mathbf{K} \to \mathbf{C}$ we have*

$$\sigma\big(xu^{-1}\big) = 2^{O\left(\sqrt{f \log f}\right)} \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(x)^{\frac{1}{\varphi(f)}}.$$

[12]One can compute u by simply removing the absolute values in the definition of Log, and taking any determination of complex logarithm. As we work inside a CM-field, this technicality is not needed.

B.2.1. *Setting.* Let us consider an integer f and take its prime decomposition $f = \prod_{i=1}^{r} p_i^{e_i}$. We set $q_i = p_i^{e_i}$ and we fix the cyclotomic field $\mathbf{K} = \mathbf{Q}[\zeta_f]$ of conductor f. Classically, the Galois group of $\mathbf{K}$ is equal to $G = {(\mathbf{Z}/f\mathbf{Z})^\times}/{\{-1, 1\}}$, whose elements are the σ_α, sending ζ_f to ζ_f^α for any $\alpha \in G$.

B.2.2. *Cyclotomic units and their generators.* The cyclotomic units are defined as all the products of $\pm\zeta_f$ and $\zeta_f^a - 1$ which are units. We let $\mathcal{Q}$ be the set of the 2^r possible products of the q_i.

A standard theorem of [28, Lemma 2.2] reduces the number of generators of the cyclotomic units:

Theorem 18. *The cyclotomic units are all the products of $\pm\zeta_f$ and $G \cdot (\zeta_f^a - 1)$ which are units, when a runs through $\mathcal{Q}$.*

Proof. Let $a \in \mathbf{Z}$, and define k to be the product of all the q_i dividing a, so that by construction $k \in \mathcal{Q}$. Now, we have:

$$1 - \zeta_f^a = \prod_{i=0}^{\frac{a}{k}-1} 1 - \zeta_f^{k+\frac{ifk}{a}}.$$

Let $p_j | k + \frac{ifk}{a}$. Remark that $p_j | \frac{fk}{a}$, so that $p_j | k$, and by definition of k we have $q_j | k$. We have therefore $q_j | \frac{fk}{a}$ and hence $\zeta_f^{k+ifk/a} - 1 \in \pm G \cdot \zeta_f^k - 1$. ∎

Theorem 19. *Let χ be an even Dirichlet character of conductor $c \mid f$ with $c > 1$ and $e \in \mathcal{Q}$. Then if c and e are coprime, then*

$$|\chi(\mathit{Log}(\zeta_f^e - 1))| = \frac{\varphi(e)\sqrt{c}}{2\ln(2)} \left(\prod_{\substack{i \\ p_i | \frac{f}{e}}} |1 - \chi(p_i)| \right) |L(1, \chi)|$$

else it is 0.

Proof. If $\gcd(c, e) > 1$, we have $\sum_{\alpha \in (\mathbf{Z}/\gcd(c,e)\mathbf{Z})^\times} \chi(\alpha) = 0$ so the result is zero. We therefore assume for now on that c and e are coprime.

We first compute:

$$\prod_{\substack{\beta \in G \\ \beta = 1 \bmod c}} 1 - \zeta_{\frac{f}{e}}^\beta.$$

Let $p_i | \frac{f}{ec}$ and $p_i | c$. Then:

$$\begin{aligned} \prod_{\substack{\beta \in G \\ \beta = 1 \bmod c}} 1 - \zeta_{\frac{f}{e}}^\beta &= \prod_{\substack{\beta \in G \\ \beta = 1 \bmod cp_i}} \prod_{j=0}^{p_i - 1} 1 - \zeta_{\frac{f}{e}}^\beta \zeta_{p_i}^j \\ &= \prod_{\substack{\beta \in G \\ \beta = 1 \bmod cp_i}} 1 - \zeta_{\frac{f}{e}}^{p_i \beta}. \end{aligned}$$

In the same way, we have if $p_i|\frac{f}{e}$ and $p_i \nmid c$, with $r^{-1} = \frac{f}{eq_i} \bmod p_i$:

$$\begin{aligned}\prod_{\substack{\beta\in G\\ \beta=1 \bmod c}} 1-\zeta_{\frac{f}{e}}^{\beta} &= \prod_{\substack{\beta\in G\\ \beta=1 \bmod cq_i}} \prod_{\substack{j=0\\ j\neq -r \bmod p_i}}^{q_i-1} 1-\zeta_{\frac{f}{e}}^{\beta}\zeta_{q_i}^{\beta j}\\ &= \prod_{\substack{\beta\in G\\ \beta=1 \bmod cq_i}} \frac{1-\zeta_{\frac{f}{e}}^{\beta q_i}}{1-\zeta_{\frac{f}{e}}^{\beta(q_i-\frac{rf}{e})/p_i}}\\ &= \prod_{\substack{\beta\in G\\ \beta=1 \bmod cq_i}} \frac{1-\zeta_{\frac{f}{eq_i}}^{\beta}}{1-\zeta_{\frac{f}{eq_i}}^{\frac{\beta}{p_i}}}.\end{aligned}$$

In case $p_i|e$, we have $q_i|e$ and therefore

$$\prod_{\substack{\beta\in G\\ \beta=1 \bmod c}} 1-\zeta_{\frac{f}{e}}^{\beta} = \prod_{\substack{\beta\in G\\ \beta=1 \bmod cq_i}} \left(1-\zeta_{\frac{f}{e}}^{\beta}\right)^{\varphi(q_i)}.$$

We can now compute our sum:

$$\begin{aligned}\sum_{\alpha\in G}\chi(\alpha)\log(|\zeta_f^{e\alpha}-1|) &= \sum_{\alpha\in(\mathbf{Z}/c\mathbf{Z})^\times/\{-1,1\}}\chi(\alpha)\log\left(\left|\sigma_\alpha\left(\prod_{\beta\in G\beta=1 \bmod c}\zeta_{\frac{f}{e}}^{\beta}-1\right)\right|\right)\\ &= \varphi(e)\left(\prod_{\substack{i\\ p_i|\frac{f}{e}\\ p_i\nmid c}} 1-\chi(p_i)\right)\sum_{\alpha\in(\mathbf{Z}/c\mathbf{Z})^\times/\{-1,1\}}\chi(\alpha)\log(|\zeta_c^\alpha-1|).\end{aligned}$$

We finish by the standard computation ([52, Theorem 4.9]) of the term on the right with the Gauss sum: $\tau = \sum_{\alpha\in(\mathbf{Z}/c\mathbf{Z})^\times}\overline{\chi}(\alpha)\zeta_c^\alpha$:

$$\begin{aligned}\sum_{\alpha\in(\mathbf{Z}/c\mathbf{Z})^\times}\overline{\chi}(\alpha)\ln(|\zeta_c^\alpha-1|) &= \sum_{\alpha\in(\mathbf{Z}/c\mathbf{Z})^\times}\overline{\chi}(\alpha)\ln(1-\zeta_c^\alpha)\\ &= \sum_{\alpha\in(\mathbf{Z}/c\mathbf{Z})^\times}\sum_{k=1}^{\infty}\overline{\chi}(\alpha)\frac{\zeta_c^{\alpha k}}{k}\\ &= \sum_{i=1}^{\infty}\frac{\tau\chi(k)}{k} = \tau L(1,\chi)\end{aligned}$$

and $\tau\overline{\tau} = c$. ∎

Definition 9. *The* augmentation ideal *is the kernel of the form:* $(\sum_\alpha x_\alpha\sigma_\alpha \to \sum_\alpha x_\alpha)$ *over* $\mathbf{Z}[G]$.

With this definition we can complete the description of the cyclotomic units:

Theorem 20. [28, Lemma 2.4] *The cyclotomic units are generated by:*

- *The pair* $\pm\zeta_f$,
- *the* $G \cdot \zeta_f^a - 1$ *for all* $a \in \mathcal{Q}$ *such that* $\frac{f}{a}$ *is not prime power,*
- *the orbit of* $\zeta_f^{f/q_i} - 1$ *by the action of the augmentation ideal.*

Proof. Note first that for any $a \in \mathcal{Q}$, $(1 - \sigma_\alpha) \cdot (\zeta_f^a - 1) \in \mathcal{O}_{\mathbf{K}}$. Next, we prove that an element u generated by the $\zeta_f^a - 1$ is a unit if $\mathcal{N}_{\mathbf{K}/\mathbf{Q}}(u) = 1$. We remark that

$$\varphi(f) \cdot u = \mathcal{N}_{\mathbf{K}/\mathbf{Q}}(u)\left(\left(\sum_\alpha 1 - \sigma_\alpha\right) \cdot u\right) = \left(\sum_\alpha 1 - \sigma_\alpha\right) \cdot u$$

so that it is a unit. The converse is clear. Finally $\mathcal{N}_{\mathbf{K}/\mathbf{Q}}(1-\zeta_f^a)$ is easily computed to be $p_i^{\varphi(a)}$ if $a = f/q_i$ and 1 else using the equations at the beginning of the proof of theorem 19. ∎

B.2.3. *Construction of an "orthogonal" basis.* We now define the family $(b_i)_{1 \le i \le |\mathcal{Q}|}$ by setting $b_i = \mathrm{Log}(\zeta_f^a - 1)$ where the $a \in \mathcal{Q}$ are taken in decreasing order. We can define some Gram-Schmidt orthogonalization on this family with the relations:

$$b_i^* = b_i - \sum_{j<i} \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle} b_j^* = b_i - \sum_{j<i} b_i b_j^* (b_j^*)^\dagger$$

where the dagger is the Moore-Penrose pseudo-inverse. As such, $\chi(b_i) = \chi(b_i^*)$ if $\chi(b_j^*) = 0$ for all $j < i$, and is equal to zero elsewhere. As $L(1, \chi) \neq 0$, we have for all $\chi \neq 1$ that $\chi(b_i^*) \neq 0$ iff $\mathrm{rad}(\frac{f}{e})|c|\frac{f}{e}$ where c is the conductor of the character χ. Furthermore, in this case, the term $\prod_{p_i|\frac{f}{e}}(1 - \chi(p_i))$ is one. We can now give our decoding algorithm, assuming again that the cyclotomic units have a finite index:

Proof of theorem 17. We let $b_i = \mathrm{Log}(\zeta_f^e - 1)$ and recall that for all χ with conductor not coprime with e we have $\chi(b_i) = 0$. We remark that if $\frac{f}{e}$ is a prime power, we have $b_i^* = b_i$ and as a result $\|b_i^*\|_\infty \le \log(\frac{f}{e})$. Also, we have for all i that $\|b_i^*\| \le \|b_i\| = \mathrm{O}\left(\sqrt{\varphi\left(\frac{f}{e}\right)}\right)$ using the same technique. The algorithm consists in using Babai reduction with our generating family, with the modification described above to round with respect to the augmentation ideal when we have to. More precisely, for any $y \in \mathbf{Z}[G]b_i^*$, we compute z a randomized rounding of y/b_i^* in the same way as in the previous section. If $\frac{f}{e}$ is a prime power, the rounding is $z - \sum_\alpha z_\alpha \sigma_1$, else it is z. If $|\sum_\alpha z_\alpha| \ge \frac{\sqrt{\frac{f}{e}}}{\log(\frac{f}{e})}$ in case where $\frac{f}{e}$ is a prime power, we restart the rounding. We then continue in the same way with $i-1$. The analysis is as before. The randomized rounding produces an error with subgaussian coordinates with parameter $\mathrm{O}\left(\sqrt{\sum_{e \in X} \varphi(\frac{f}{e})}\right) = \mathrm{O}(\sqrt{f})$. The correction for the prime power adds an error bounded by $\sum_i \log(q_i)\sqrt{q_i}/\log(q_i) = O(\sqrt{f})$.

Hence, the bound on the output holds. The running time is quasi-linear since we can work at each step with the ring

$$\mathbf{Z}\left[\left(\mathbf{Z}\Big/\left(\frac{f}{e}\right)\mathbf{Z}\right)^{\times}\right].$$

■

Remark that the running time is also quasi-linear if we work with the input and output in the logarithm space. Note that $\mathbf{K}^+ = \mathbf{Q}[\zeta_f + \overline{\zeta}_f]$ has the same units, up to torsion. As such, the same theorem is true for $\mathbf{K}^+$. It has the following algorithmic implication. Given an ideal $\mathfrak{a} \subset \mathcal{O}_{\mathbf{K}^+}$, as the class group order of $\mathbf{K}^+$ is usually small, it is simple to find an ideal $\alpha\mathcal{O}_{\mathbf{K}^+} \subset \mathfrak{a}$ with low norm. From there, we can compute a generator α in quantum polynomial time and using the above theorem on α, we have found quickly an element in $\mathfrak{a}$ with approximation factor $2^{O(\sqrt{f \log f})}$.

B.3. **BDD on the unit lattice.** The following theorem has deep implications in arithmetic. One part is due to Landau [29], another to Dirichlet.

Theorem 21. *Let χ be a character of conductor $c > 1$. If $\chi^2 = 1$ (χ is quadratic) we have $|L(1,\chi)| = \Omega(1/\sqrt{c})$, else $|L(1,\chi)| = \Omega(1/\log(c))$.*

Note that under the Generalized Riemann Hypothesis we can take $|L(1,\chi)| = \Omega(1/\log\log c)$ and for most characters we have $|L(1,\chi)| = \mathrm{O}(1)$. This justifies our previous assumptions. We let $\tau(f) = \prod_i 1 + e_i$ be the number of divisors of f; we have the well-known bound $\tau(f) = f^{\mathrm{O}(1/\log\log f)}$. We can now prove our BDD[13] theorem:

Theorem 22. *Given $\mathbf{K} = \mathbf{Q}[\zeta_f]$, there are $\varphi(f)/2$ (explicit) elements r_i of norm*

$$O\left(\frac{\sqrt{\tau(f)}}{n}\log(n)\right)$$

in $\mathbf{R}[G]$ with the following property. Let $x \in (\mathbf{R}\otimes\mathbf{K})^{\times}$ be such that there is a cyclotomic unit u with for all i, $|\langle r_i, \mathrm{Log}(x/u)\rangle| < 1/3$. Then, given x we can find u up to a power of ζ_f in quasi-linear time.

Proof. The algorithm is similar to the previous one. We first scale x to get $\mathcal{N}_{\mathbf{K}/\mathbf{Q}}(x) = 1$. For decreasing i, we compute z the (deterministic) rounding $\mathrm{Log}(x)/b_i^*$ where we force $\sum_\alpha z_\alpha = 0$ if $b_i = \mathrm{Log}(\zeta_f^e - 1)$ with $\frac{f}{e}$ a prime power, and we then divide x by $z \cdot \zeta_f^e - 1$. We first bound $\|(b_i^*)^\dagger\|_2$ where $b_i = \zeta_f^e - 1$. Thanks to our previous computations and the character orthogonality relation, we have

$$\left\|(b_i^*)^\dagger - \frac{\sum_\alpha \sigma_\alpha}{\sum_\alpha (b_i^*)_\alpha}\right\|^2 = \frac{1}{|G|}\sum_\chi \frac{4\ln^2 2}{c\varphi(e)^2}\frac{1}{|L(1,\chi)|^2}$$

[13]The usual definition of BDD is about the worst case decoding distance. The implied worst case bound is too large to be useful, but with high probability we can decode large Gaussian noise, which is enough for current applications.

where χ has a conductor $c > 1$ with $\mathrm{rad}(\frac{f}{e})|c|\frac{f}{e}$. The Chinese Remainder theorem implies that:

$$\sum_{\chi} \frac{1}{c} = \frac{1}{2} \prod_{p_i | \frac{f}{e}} \sum_{k=1}^{e_i} \frac{(p_i - 1)p_i^{k-1}}{p_i^k} = \frac{1}{2} \prod_{p_i | \frac{f}{e}} e_i(1 - 1/p_i)$$

with the same assumptions on χ. We have at most 2^{r+1} quadratic characters, so we get:

$$\left\| (b_i^*)^\dagger - \frac{\sum_\alpha \sigma_\alpha}{\sum_\alpha (b_i^*)_\alpha} \right\|^2 \leq \mathrm{O}\left(\frac{2^r + \log^2\left(\frac{f}{e}\right) \prod_{p_i | \frac{f}{e}} e_i}{\varphi(f)\varphi(e)^2} \right)$$

which is in $\mathrm{O}\big(n^{-1}\log^2(n)\tau(f)\big)$. Now each non-zero coefficient of z in the algorithm can be expressed as an inner product between an element of $G(b_i^*)^\dagger$ and $\mathrm{Log}(x/u)$, which is of unit norm. This leads to a r vector for each coefficient, and with the given condition this guarantees that $\mathrm{Log}(u)$ is exactly recovered. ■

This implies that given any generator of the ideal $\alpha\mathcal{O}_{\mathbf{K}}$ where α is sampled from a large discrete Gaussian, we can recover α in quasi-linear time; see [11, Section 5]. The practical average length of the r_i is of course on the order of $\sqrt{\frac{\prod_i e_i}{n}}$.

Appendix C. The symplectic structure in all number fields

In section 6, we described how to obtain a symplectic structure when $\mathbf{L} = \mathbf{K}[X]/(X^d + a)$. We show here the general case, with $\mathbf{L} = \mathbf{K}[X]/f(X)$. We first give a simple construction which recovers the one given above but has losses in the general case; and then describe a general construction without losses.

C.1. The dual integer construction.

We have the following lemma, proved in [38, Chapter III, Proposition 2.4]:

Lemma 18. *Let $a_i = X^i$ and $\sum_i b_i Y^i = \frac{f(Y)}{Y - X}$. Then $\mathrm{tr}_{\mathbf{L}/\mathbf{K}}(a_i b_j / f'(X))$ is equal to 1 if $i = j$ and 0 else.*

This suggests taking as a $\mathbf{K}-$basis for $\mathbf{L}^2$ the $(a_i, 0)$ followed by the $(0, b_i)$. With the notations of section 6, we now define $J'_{\mathbf{L}}$ as

$$\mathrm{tr}_{\mathbf{L}/\mathbf{K}}(J_{\mathbf{L}}/f'(X)).$$

It follows from the lemma that in our basis, this is represented by the Darboux matrix:

$$\begin{pmatrix} 0 & \mathrm{Id}_d \\ -\mathrm{Id}_d & 0 \end{pmatrix}$$

and, as usual, we can reverse the order of the second part of the basis to obtain the wanted matrix.

We can convert efficiently a number $z \in \mathbf{L}$ in the basis of b_i. Clearly, the coefficients are given by all the $\mathrm{tr}_{\mathbf{L}/\mathbf{K}}(z/f'(X) \cdot X^i)$. We then simply evaluate $z/f'(X)$

on all roots of f using a remainder tree, and follow by a Vandermonde matrix-vector multiplication, which is also a multipoint evaluation [35]. In particular, we do not need to compute the b_i.

There is however a loss with this basis: the algorithm tries to minimize the size of the coefficients in our basis of $\mathbf{L}^2$ instead of the canonical norm.

C.2. **The orthogonal construction.** We want to build an orthogonal $\mathbf{R} \otimes \mathbf{K}$-basis of $\mathbf{R} \otimes \mathbf{L}$. We assume for simplicity (only) that $\mathbf{L}$ (and therefore $\mathbf{K}$) is a totally real field. Hence, with $\mathbf{K} = \mathbf{Q}[Y]/g(Y)$, we have that all roots r_i of g are real, and when we evaluate all coefficients of f on r_i, the resulting polynomial has real roots $r_{i,j}$.

We then define the j-th element of the basis as being the element of $\mathbf{L}$ which, when we evaluate on $(X - r_{i,k}, Y - r_i)$, we obtain 1 if $j = k$ and 0 else. This is clearly an orthogonal basis for the canonical norm, and in this case, it is also its dual. Hence, using twice this basis leads again to the Darboux matrix for $J'_{\mathbf{L}} = \mathrm{tr}_{\mathbf{L}/\mathbf{K}}(J_{\mathbf{L}})$. Exactly the same construction works for totally imaginary $\mathbf{K}$ (and therefore $\mathbf{L}$).

The general case can be done in the same way, by taking care of ramified places.

Appendix D. Reduction with linear algebra

We shall prove that lattice reduction is no easier than linear algebra on a large field $\mathbf{Z}/p$. We start by defining the problems.

Definition 10 (Lattice reduction)**.** *The problem of lattice reduction consists in, given an integer matrix A of dimension d with $\|A\|, \|A^{-1}\| \leq 2^B$, outputting a matrix AU with U a unimodular integer matrix such that with $QR = AU$ the QR-decomposition, we have for all i:*

$$R_{i,i} \leq 2R_{i+1,i+1}.$$

Definition 11 (Kernel problem)**.** *The kernel problem consists in, given a square matrix A of dimension d over $\mathbf{Z}/p$, outputting a matrix K such that $AK = 0$ and the number of columns of K is* $\dim \ker A$.

Theorem 23. *If one can solve the lattice problem in dimension $2d$ with parameter B, one can solve the kernel problem in dimension d for any prime $p \leq 2^{B/2-d}d^{-1}$ with the same complexity, up to a constant.*

Proof. Let A,p be the input of the kernel problem. The matrix

$$L = \begin{pmatrix} pd2^{2d-1} \cdot p\mathrm{Id}_d & pd2^{2d-1}A \\ 0 & \mathrm{Id}_d \end{pmatrix}$$

is given to the lattice reduction oracle. The output is of the form

$$\begin{pmatrix} 0 & * \\ K & * \end{pmatrix}$$

where we maximize the number k of columns of K. The reduction returns this matrix K.

We have $\|L\| \leq d^2 2^{2d} p^2 \leq 2^B$ and $\|L^{-1}\| \leq 2d$ which is also less than 2^B since $p \geq 2$. It is clear that vectors in $L\mathbf{Z}^{2d}$ of the form $\begin{pmatrix} 0 \\ x \end{pmatrix}$ are exactly the integer solutions of $Ax = 0 \bmod p$. We let QR be the QR-decomposition of AU. Let K' be a basis of $\ker A$, where entries are integers smaller than p. Then, since U is unimodular, there is an integer matrix V such that

$$AUV = \begin{pmatrix} 0 \\ K' \end{pmatrix}.$$

If V has no nonzero entries $V_{i,j}$ with $i > k$, then it is clear that the output is correct. Hence, we consider v a column of V where it is not the case, and let i be maximal with $v_i \neq 0$. First, we have $\|AUv\| \leq \sqrt{d}p$. Second, as Q is orthogonal, we have $\|AUv\| = \|Rv\| \geq R_{i,i}$. Third, the definition of k implies that $R_{k+1,k+1} \geq d2^{2d-1}p$. As the lattice is reduced and $i > k$, we have $R_{i,i} \geq R_{k+1,k+1} 2^{1-2d}$. We conclude that:

$$\sqrt{d}p \geq \|AUv\| \geq R_{k+1,k+1} 2^{1-2d} \geq dp$$

which is a contradiction. ■

As we expect the kernel problem to have a complexity of $\Omega(d^\omega B/\log B + d^2 B)$, we can expect the same for the lattice reduction problem. The reduction can of course be extended with other rings, and also to compute a span.

Sorbonne universités, lip6, paris, france
E-mail address: t.espitau@gmail.com

Rennes Univ, irisa
E-mail address: pa.fouque@gmail.com
E-mail address: paul.kirchner@irisa.fr