An L(1/3) algorithm for ideal class group and regulator computation in certain number fields

Jean-François Biasse

AN $L(1/3)$ ALGORITHM FOR IDEAL CLASS GROUP AND REGULATOR COMPUTATION IN CERTAIN NUMBER FIELDS

JEAN-FRANÇOIS BIASSE

Abstract. We analyse the complexity of the computation of the class group structure, regulator, and a system of fundamental units of a certain class of number fields. Our approach differs from Buchmann's, who proved a complexity bound of $L(1/2, O(1))$ when the discriminant tends to infinity with fixed degree. We achieve a subexponential complexity in $O(L(1/3, O(1)))$ when both the discriminant and the degree of the extension tend to infinity by using techniques due to Enge and Gaudry in the context of algebraic curves over finite fields.

1. Introduction

Let $\mathbb{K} = \mathbb{Q}(\theta)$ be a number field of degree n and discriminant Δ. The ideal class group of its maximal order $\mathcal{O}_\mathbb{K}$ is a finite abelian group that can be decomposed as:

$$\mathrm{Cl}(\mathcal{O}_\mathbb{K}) = \bigoplus_i \mathbb{Z}/d_i\mathbb{Z},$$

with $d_i \mid d_{i+1}$. Computing the structure of $\mathrm{Cl}(\mathcal{O}_\mathbb{K})$, along with the regulator and a system of fundamental units of $\mathcal{O}_\mathbb{K}$ is a major task in computational number theory. In addition, many algorithms solving the discrete logarithm problem are based on the group structure computation.

In 1968, Shanks [12, 13] proposed an algorithm relying on the baby-step giant-step method to compute the structure of the ideal class group and the regulator of a quadratic number field in time $O\left(|\Delta|^{1/4+\epsilon}\right)$, or $O\left(|\Delta|^{1/5+\epsilon}\right)$ under the extended Riemann hypothesis [10]. Then, a subexponential strategy for the computation of the group structure of the class group of an imaginary quadratic extension was described in 1989 by Hafner and McCurley [9]. The expected running time of this method is

$$L_\Delta(1/2, \sqrt{2} + o(1)) = e^{\left(\sqrt{2}+o(1)\right)\sqrt{\log|\Delta|\log\log|\Delta|}}.$$

Buchmann [2] generalized this result to the case of an arbitrary extension, the complexity being valid for fixed degree n and Δ tending to infinity. Enge [5] used this technique in the context of discrete logarithm computations in the Jacobian of hyperelliptic curves, and developed with Gaudry [6] an algorithm for computing the group structure of the Jacobian and solving the discrete logarithm problem for a certain class of curves in time:

$$L_{q^g}\left(1/3, O(1)\right) = e^{O(1)\left(\log(q^g)^{1/3}\log\log(q^g)^{2/3}\right)}.$$

2000 *Mathematics Subject Classification.* Primary 54C40, 14E20; Secondary 46E25, 20C20.

Key words and phrases. Number fields, ideal class group, regulator, units, index calculus, subexponentiality.

The author was supported by a DGA grant.

In this paper, we adapt the $L(1/3)$ algorithm of Enge and Gaudry to the computation of the group structure of the ideal class group, the regulator, and a system of fundamental units of $\mathcal{O}_{\mathbb{K}}$. We deal with the case where both the discriminant and the degree of the extension grow to infinity in certain proportions, whereas in [2] the degree is assumed to be fixed.

2. Main idea

We consider a number field $\mathbb{K} = \mathbb{Q}(\theta)$ of discriminant Δ which can be written as:

$$\mathbb{K} = \mathbb{Q}[X]/T(X),$$

with $T(X) = t_n X^n + t_{n-1}X^{n-1} + \ldots + t_0 \in \mathbb{Z}[X]$, and $n := [\mathbb{K} : \mathbb{Q}]$. Let d be a bound on the bit size of the coefficients of T:

$$d := \max_i \{\log(t_i)\}.$$

In addition, we require that:

$$n \leq n_0 \log\left(|\Delta|\right)^{\alpha} (1 + o(1)) \tag{1}$$

$$d \leq d_0 \log\left(|\Delta|\right)^{1-\alpha} (1 + o(1)), \tag{2}$$

for some $\alpha \in \left[\frac{1}{3}, \frac{2}{3}\right[$, and some constants n_0 and d_0. We define $\kappa := n_0 d_0$. We also denote by r_1 the number of real places, by r_2 the number of complex places and we define $r := r_1 + r_2 - 1$. Our algorithm computes the group structure of $\mathrm{Cl}(\mathbb{Z}[\theta])$, its regulator, and a system of fundamental units of $\mathbb{Z}[\theta]$, in expected time lying in:

$$O\left(L_{\mathrm{Disc}(T)}(1/3, O(1))\right).$$

In the case of number fields satisfying $\mathbb{Z}[\theta] = \mathcal{O}_{\mathbb{K}}$ and the above restrictions, we compute the group structure of $\mathrm{Cl}(\mathcal{O}_{\mathbb{K}})$, R, and a system of fundamental units, in expected time $L_{\Delta}(1/3, O(1))$. From now on, we assume that $\mathbb{K}$ satisfies (1), (2), and $\mathbb{Z}[\theta] = \mathcal{O}_{\mathbb{K}}$.

Example. Let $\Delta \in \mathbb{Z}$, and $\mathbb{K}_{n,K}$ be an extension of $\mathbb{Q}$ defined by an irreducible polynomial of the form:

$$T(X) = X^n - K,$$

with

$$\log K = \left\lfloor \log\left(|\Delta|\right)^{1-\alpha} \right\rfloor$$
$$n = \left\lfloor \log\left(|\Delta|\right)^{\alpha} \right\rfloor,$$

for some $\alpha \in \left[\frac{1}{3}, \frac{2}{3}\right[$. Then, $\mathcal{O}_{\mathbb{K}_{n,K}}$ has discriminant satisfying:

$$\log(\mathrm{Disc}(\mathcal{O}_{\mathbb{K}_{n,K}})) = \log(n^n K^{n-1}) = \log(|\Delta|)(1 + o(1)).$$

If in addition we require that n and K be the largest prime numbers below their respective bounds such that:

$$n^2 \nmid K^{n-1} - 1,$$

then we meet the last restriction $\mathbb{Z}[\theta] = \mathcal{O}_{\mathbb{K}_{n,K}}$.

We proceed by analogy with the approach of [6] in the context of algebraic curves, where the authors examined curves of the form:

$$\mathcal{C} : Y^n + X^d + f(X, Y),$$

such that any monomial X^iY^j occuring in f satisfies $ni + dj < nd$. The genus g is assumed to tend to infinity and:

$$n \approx g^{\alpha}$$
$$d \approx g^{1-\alpha}.$$

The idea in [6] is to look for functions $\phi(X, Y) \in \mathbb{F}_q[X, Y]$ satisfying:

$$\deg_Y \phi \approx g^{\alpha-1/3} \quad \text{and} \quad \deg_X \phi \approx g^{2/3-\alpha},$$

with $\mathcal{N}(\phi)$ splitting into polynomials of degree bounded by $B = \log(L(1/3, \rho))$ for some number ρ determined in the complexity analysis. Each time such a decomposition occurs, the ideal (ϕ) is necessarily a product of primes belonging to the set $\mathcal{B}$ of the prime ideals of degree bounded by B:

$$(\phi) = \prod_{\mathfrak{p}_i \in \mathcal{B}} \mathfrak{p}_i^{e_i}.$$

Such a decomposition of a principal ideal is called a *relation*. In the following, we will also denote the vector (e_i) itself a relation. Every time we find a relation, we add the row vector (e_i) to a matrix $M \in \mathbb{Z}^{m \times N}$ called the *relation matrix*, where $N := |\mathcal{B}|$, and $m \geq k$ is the number of relations collected. A linear algebra step is performed on this matrix. It consists in computing its Smith Normal Form, that is to say integers $d_1, \ldots, d_N$, with $d_N | d_{N-1} | \ldots | d_1$, such that there exist two unimodular matrices $U \in \mathbb{Z}^{m \times m}$ and $V \in \mathbb{Z}^{N \times N}$ satisfying:

$$M = U \begin{pmatrix} d_1 & & (0) & \\ & \ddots & & (0) \\ (0) & & d_k & \\ \hdashline & & (0) & \end{pmatrix} V.$$

The SNF of M provides us with the group structure of the Jacobian of the curve $\mathcal{C}$. Indeed, if $\mathcal{L}_{\mathbb{Z}}$ is the lattice spanned by all the possible relations, and if $\mathcal{J}$ denotes the Jacobian of $\mathcal{C}$, then we have:

$$\mathcal{J} \simeq \mathbb{Z}^N / \mathcal{L}_{\mathbb{Z}}.$$

Providing m is large enough to ensure that the rows of M generate $\mathcal{L}_{\mathbb{Z}}$, we have:

$$\mathcal{J} \simeq \bigoplus_i \mathbb{Z}/d_i\mathbb{Z}.$$

In our context, we need the group structure of $\mathrm{Cl}(\mathcal{O}_{\mathbb{K}})$, along with the regulator R, and a system of fundamental units of $\mathcal{O}_{\mathbb{K}}$. The computation of the group structure of $\mathrm{Cl}(\mathcal{O}_{\mathbb{K}})$ is done using methods similar to those used for the computation of the structure of $\mathcal{J}$. We look for relations of the form:

$$(\phi) = \prod_i \mathfrak{p}_i^{e_i},$$

where $\phi \in \mathbb{K}$, and where the $\mathfrak{p}_i$ are prime ideals of norm bounded by $L(1/3, \rho)$. Every time we find such a relation, we add the row vector $(e_i)_{i \leq N}$ to the relation

matrix denoted by $M_{\mathbb{Z}} \in \mathbb{Z}^{m \times N}$. To continue the analogy with [6], we require that ϕ be of the form:

$$\phi = A(\theta),$$

where $A \in \mathbb{Z}[X]$ of degree k. During the analysis, we will provide bounds on k and on the coefficients of A, that delimit the search space. Providing the rows of $M_{\mathbb{Z}}$ generate the lattice $\mathcal{L}_{\mathbb{Z}}$ of all the possible row vectors $(e_i)_{i \leq N} \in \mathbb{Z}^N$ representing a relation, we have:

$$\mathrm{Cl}(\mathcal{O}_{\mathbb{K}}) \simeq \mathbb{Z}^N / \mathcal{L}_{\mathbb{Z}} \simeq \bigoplus_{i \leq N} \mathbb{Z}/d_i\mathbb{Z},$$

where the d_i are the diagonal coefficients of the SNF of $M_{\mathbb{Z}}$. The main difference with the context of algebraic curves is the computation of R and of a system of fundamental units. The group of units of $\mathcal{O}_{\mathbb{K}}$ is of the form:

$$U(\mathbb{K}) \simeq \mu(\mathbb{K}) \times \mathbb{Z}^r,$$

where $\mu(\mathbb{K})$ is the multiplicative group of the roots of unity in $\mathcal{O}_{\mathbb{K}}$. A system of fundamental units (γ_i), $i \leq r$, is a set of elements of $\mathbb{K}$ satisfying:

$$U(\mathbb{K}) \simeq \mu(\mathbb{K}) \times \langle \gamma_1 \rangle \times \ldots \times \langle \gamma_r \rangle .$$

Once such a system is found, we use the logarithm map:

$$\begin{array}{rccc} & \mathbb{K} & \longrightarrow & \mathbb{R}^{r+1} \\ \mathrm{Log}: & \phi & \longmapsto & (\log|\phi|_1, \ldots, \log|\phi|_{r+1}), \end{array}$$

where the $|.|_j$ are the archimedian valuations on $\mathbb{K}$, to construct a matrix $A \in \mathbb{R}^{r \times (r+1)}$ whose rows are the vectors $\mathrm{Log}(\phi_i)$, for $i \leq r$. The regulator is defined as the determinant of any $r \times r$ minor of A. To construct A and a system of fundamental units, we augment the row vectors by columns containing the archimedian valuations, and add the row:

$$(e_1, \ldots, e_k, \log|\phi|_1, \ldots, \log|\phi|_{r+1}) \in \mathbb{Z}^N \times \mathbb{R}^{r+1}$$

to a relation matrix M whenever a relation $(\phi) = \prod_i \mathfrak{p}_i^{e_i}$ is found. A linear algebra step performed on M provides us with the group structure, the regulator, and a system of fundamental units. It is described in detail in §4.

3. The relation matrix

Let ρ be a constant to be determined later, and B a smoothness bound satisfying:

$$B = \lceil L_\Delta(1/3, \rho) \rceil.$$

We define the factor base $\mathcal{B}$ as the set of all non inert prime ideals of norm bounded by B. This factor base has cardinality:

$$N := |\mathcal{B}| = L(1/3, \rho + o(1)).$$

In the following, we will need to test the smoothness of principal ideals of the form (ϕ), where $\phi = A(\theta)$ with $A \in \mathbb{Z}[X]$. We will use the well-known result that is proved in [4], Lemma 3.3.4:

Lemma 1. *The norm of ϕ satisfies:*

$$\mathcal{N}(\phi) = \mathit{Res}\,(T(X), A(X)),$$

where Res denotes the resultant.

Computing $\mathcal{N}(\phi)$ for $\phi \in \mathbb{K}$ allows us to decide whether ϕ is a product of prime ideals $\mathfrak{p} \in \mathcal{B}$. Indeed, it suffices to check if $\mathcal{N}(\phi) \in \mathbb{Z}$ is B-smooth which can be done by trial division or the ECM method in polynomial time. We assume that the coefficients a_i of the polynomial A have their logarithm bounded by an integer a, and that there exist two constants δ and ν to be determined later such that:

$$a \leq \left\lceil \delta \frac{\kappa \log |\Delta|/n}{(\log |\Delta|/\mathcal{M})^{1/3}} \right\rceil \tag{3}$$

$$k \leq \left\lceil \nu \frac{n}{(\log |\Delta|/\mathcal{M})^{1/3}} \right\rceil, \tag{4}$$

with $\mathcal{M} := \log\log|\Delta|$. Using Lemma 1 and Hadamard's inequality, we deduce an upper bound on $\log \mathcal{N}(\phi)$:

$$\log \mathcal{N}(\phi) \leq na + dk + n \log k + k \log n \tag{5}$$

$$\leq \kappa \log\left(|\Delta|\right)^{2/3} \mathcal{M}^{1/3}(\delta + \nu + o(1)). \tag{6}$$

In the following, we will also need a bound on the real coefficients $\log|\phi|_i$ occuring in the relation matrix. By the following proposition, we derive a bound on the $\log|\theta|_i$ from the imposed bounds on the coefficients of T:

Proposition 2. *Let σ_i be the n complex embeddings of $\mathbb{K}$ such that we have $T = \prod_i (X - \sigma_i(\theta))$, then the $\sigma_i(\theta)$ satisfy:*

$$\log(|\theta|_i) = \log(|\sigma_i(\theta)|) = O(\log\left(|\Delta|\right)^{1-\alpha}).$$

Proof. Landau-Mignotte's theorem [11] states that if $D \mid T$ with $\deg D = m$, then the coefficients d_j of D satisfy:

$$|d_j| \leq 2^{m-1}(|T| + t_n),$$

where $|T|$ is the euclidian norm of the vector of the coefficients of T. Applying this to $D = X - \sigma_i(\theta)$ and $m = 1$ allows us to obtain:

$$\log(|\theta|_i) \leq \log(|T| + t_n) \in O(\log\left(|\Delta|\right)^{1-\alpha}).$$

□

Corollary 3. *With $\phi = A$, and a and k respectively bounded by (3) and (4), we have:*

$$\log|\phi|_i \leq O(\log\left(|\Delta|\right)^{2/3} \mathcal{M}^{1/3}).$$

To compute the probability for ϕ to be $\mathcal{B}$-smooth, we have to make the following assumption:

Heuristic 4. *We assume that $\mathcal{N}(\phi)$ behaves like a random number whose logarithm satisfies*

$$\log(\mathcal{N}(\phi)) \leq \iota := \kappa \log\left(|\Delta|\right)^{2/3} \mathcal{M}^{1/3}(\delta + \nu + o(1)),$$

and whose distribution is given by the ψ function of [3].

Consequently, computing the probability for a given (ϕ) to be $\mathcal{B}$-smooth boils down to computing the probability for a number whose logarithm is bounded by ι to be smooth with respect to prime numbers with logarithm bounded by

$$\mu := \lceil \rho \log\left(|\Delta|\right)^{1/3} \mathcal{M}^{2/3} \rceil.$$

Using [3], and carrying out the same computation as in the proof of Theorem 1 of [6], one readily shows the following result on the probability of finding a relation:

Proposition 5. *Let:*

$$\iota = \lfloor \log L(\zeta, c) \rfloor = \lfloor c \log \left(|\Delta|\right)^{\zeta} \mathcal{M}^{1-\zeta} \rfloor$$
$$\mu = \lceil \log L(\beta, d) \rceil = \lceil d \log \left(|\Delta|\right)^{\beta} \mathcal{M}^{1-\beta} \rceil,$$

then we have:

$$\frac{\psi(\iota, \mu)}{e^{\nu}} \geq L\left(\zeta - \beta, \frac{-c}{d}(\zeta - \beta) + o(1)\right),$$

where $\psi(\iota, \mu)$ denotes the cardinality of the set of integers x such that $\log x \leq \iota$, and x is smooth with respect to the set of prime numbers p such that $\log p \leq \mu$.

4. The linear algebra phase

In this section, we start with an overview of the linear algebra phase, then we address its complexity in §4.1 and §4.2. We denote by M the relation matrix whose rows lie in $\mathbb{Z}^N \times \mathbb{R}^{r+1}$, and by $M_{\mathbb{Z}}$ and $M_{\mathbb{R}}$ the matrices formed respectively by the first N and the last $r+1$ columns of M. M thus has the following shape:

$$M = \left(\begin{array}{c:c} M_{\mathbb{Z}} & M_{\mathbb{R}} \end{array} \right).$$

To make sure we generate the full lattice of relations, we make the following assumption:

Heuristic 6. *We assume that there is a constant K_1 such that collecting $N + K_1 r$ allows us to generate the full lattice of relations.*

In the following, we assume that Heuristic 6 is satisfied. If this is not the case (which can be tested easily as we will see at the end of this section), we start all over again and construct another relation matrix. $M_{\mathbb{R}}$ contains rational approximations of the $\log |\phi_i|_j$ for $i \leq N + K_1 r$ and $j \leq r + 1$: the discussion of approximation issues when we add or multiply two real numbers is postponed to §5. As the rows of M are assumed to generate the full lattice of the relations, the determinant of the lattice $\mathcal{L}_{\mathbb{Z}}$ spanned by the rows of $M_{\mathbb{Z}}$ gives us the class number $h(\mathcal{O}_{\mathbb{K}})$, and its Smith Normal Form $\mathrm{diag}(d_1, \ldots, d_N)$ gives us the decomposition

$$\mathrm{Cl}(\mathcal{O}_{\mathbb{K}}) \simeq \mathbb{Z}^N / \mathcal{L}_{\mathbb{Z}} \simeq \bigoplus_i \mathbb{Z}/d_i\mathbb{Z}.$$

On the other hand, we need to construct r relations of the form

$$(0, \ldots, 0, \log |\gamma|_1, \ldots, \log |\gamma|_{r+1}),$$

along with the corresponding values of γ (that are necessarily units), such that these relations generate the lattice $\mathcal{L}_{\mathbb{R}}$ of relations whose integer part contains only zero coefficients. To do this, we compute separately the Hermite Normal Form of $M_{\mathbb{Z}}$ and a basis $(\underline{u}_j)_{j \leq l}$ with $l \leq K_1 r$ of the kernel of $M_{\mathbb{Z}}$. Then, we apply the $\underline{u}_j$ to $M_{\mathbb{R}}$, thus obtaining a matrix $A_{\mathbb{R}} \in \mathbb{R}^{l \times (r+1)}$ whose rows correspond to the archimedian valuations of units $(\beta_j)_{j \leq l}$. More details on this part of the algorithm are given in §4.1. To compute the regulator R, we need to find r combinations of rows of $A_{\mathbb{R}}$, along with the corresponding units $(\gamma_i)_{i \leq r}$, that span the lattice of units $\mathcal{L}_{\mathbb{R}}$. This procedure is described in §4.2.

At the end of the linear algebra phase, we have to check a posteriori that $N+K_1r$ relations were enough to generate $\mathcal{L}_\mathbb{Z}$ and $\mathcal{L}_\mathbb{R}$. The analytic class number formula provides a number h^* computable in polynomial time satisfying:

$$h^* \leq h(\mathcal{O}_\mathbb{K})R < 2h^*.$$

Before going into more details on the linear algebra phase, we recall the main steps of this process:

Algorithm 1 Linear algebra phase

Input: M
Output: $h(\mathcal{O}_\mathbb{K})$, the structure of $\mathrm{Cl}(\mathcal{O}_\mathbb{K})$, R, and a system of fundamental units
1: Compute the HNF of $M_\mathbb{Z}$.
2: Compute the SNF of $M_\mathbb{Z}$ and deduce $h(\mathcal{O}_\mathbb{K})$ and the group structure of $\mathrm{Cl}(\mathcal{O}_\mathbb{K})$.
3: Compute a basis $(\underline{u}_j)_{j\leq l}$ of $\ker M_\mathbb{Z}$ and deduce $A_\mathbb{R}$
4: Find r independent relations generating $\mathcal{L}_\mathbb{R}$ along with the corresponding units.
5: Compute the determinant R of $\mathcal{L}_\mathbb{R}$.
6: Compute h^* and check if $h^* \leq h(\mathcal{O}_\mathbb{K})R < 2h^*$. If not create another M and go back to step one.

Notation 7. *In the following, $\mathbf{r}_i^X$ denotes the row number i of the matrix X.*

4.1. **Hermite and kernel basis computation.** To obtain the matrix $A_\mathbb{R}$, we apply the kernel basis computation algorithm described in [8] to the rectangular matrix $M_\mathbb{Z}$. It provides $l \leq K_1r$ vectors $\underline{u}_j$ in $\mathbb{Z}^{N+K_1r}$ representing linear dependencies between the rows of $M_\mathbb{Z}$. Applying those linear combinations to the rows of M yields l relations with zero coefficients on the first N coordinates. We denote by $\mathcal{L}_\mathbb{R}$ the lattice of the relations having only zeros on their first N coordinates. As we assume Heuristic 6, these l relations generate $\mathcal{L}_\mathbb{R}$. The last $r+1$ coordinates of each of the l relations created this way are added as a row vector to the matrix $A_\mathbb{R}$. In addition, for every $\underline{u}_j$ of the form:

$$\underline{u}_j = (u_j^{(1)}, \ldots, u_j^{(N+K_1r)}),$$

and for all $j \leq l$, the value $\beta_j = \prod_i \phi_i^{u_j^{(i)}}$ is the unit corresponding to the row

$$\mathbf{r}_j^{A_\mathbb{R}} = \sum_i u_j^{(i)} \mathbf{r}_i^M.$$

As we will see in §6, the coefficients $u_j^{(i)}$ are too large to allow us to compute directly $\prod_i \phi_i^{u_j^{(i)}}$ in subexponential time. We thus give the units β_j in compact representation, that is to say by storing the $\underline{u}_j$. It is proved in [14] that the computation of the $\underline{u}_j$ takes:

$$O(l^2N^3(\log N + \log |M_\mathbb{Z}|),$$

where $|M_\mathbb{Z}| = \max_{i,j}\left\{|M_\mathbb{Z}^{i,j}|\right\}$. We need a bound on $|M_\mathbb{Z}|$ to express this complexity in terms of the size of the input:

Proposition 8. $|M_\mathbb{Z}|$ *satisfies:*

$$|M_\mathbb{Z}| = O((\log|\Delta|)^{2/3}(\log\log|\Delta|)^{1/3}).$$

Proof. We restricted ourselves to ϕ satisfying

$$\log(\mathcal{N}(\phi)) \leq \kappa \left(\log|\Delta|\right)^{2/3} \left(\log\log|\Delta|\right)^{1/3} (\delta + \nu + o(1)).$$

If $\mathcal{N}(\phi) = \prod_i \mathcal{N}(\mathfrak{p}_i)^{e_i}$, then we clearly see that the vector (e_i) having the largest coefficient under the previous constraint is the one where e_1 is maximal and all the others are set to zero, providing we set $\mathfrak{p}_1$ to the prime ideal of smallest norm. In that case, e_1 satisfies:

$$e_1 = O((\log|\Delta|)^{2/3} (\log\log|\Delta|)^{1/3}).$$

□

Corollary 9. *The complexity of the computation of the kernel basis of $M_{\mathbb{Z}}$ is bounded by:*

$$O\left(L(1/3, 3\rho + o(1)\right).$$

We use the HNF algorithm described in [8]. Its bit complexity is bounded by:

$$O\left(lN^3 \left(\log N + \log|M_{\mathbb{Z}}|\right)^3 + N^5 \left(\log N + \log|M_{\mathbb{Z}}|^2\right)\right).$$

This allows us to determine explicitly the expected time taken by the computation of the HNF and of the kernel basis of $M_{\mathbb{Z}}$ with respect to the size of the entries:

Proposition 10. *The computation of the HNF and of the kernel basis of $M_{\mathbb{Z}}$ has bit complexity bounded by:*

$$O\left(L(1/3, 5\rho + o(1)\right).$$

In the following, we will need bounds on $|\underline{u}_j|$ and on $|A_{\mathbb{R}}|$. Direct application of the methods used in [8] leads to the following result:

Lemma 11. *$|\underline{u}_j|$ and $|A_{\mathbb{R}}|$ satisfy:*

$$\log|\underline{u}_j| = O\left(L(1/3, \rho + o(1))\right.$$
$$\log|A_{\mathbb{R}}| = O\left(L(1/3, \rho + o(1))\right.$$

4.2. **The computation of R and of the system of fundamental units.** To compute the regulator and a system of fundamental units, we have to find a set of r row vectors that span $\mathcal{L}_{\mathbb{R}}$. To do that, we take successive $r \times r$ determinants from submatrices extracted from $A_{\mathbb{R}}$, and we perform some elementary operations on the rows of $A_{\mathbb{R}}$. This procedure is described in Algorithm 2, which was first introduced in [4], Algorithm 6.5.7. It makes use of the real GCD algorithm, which is also presented in [4], Algorithm 5.9.3. Given two multiples of the regulator aR and bR, where a and b are integers, the real GCD algorithm outputs dR, where d is the GCD of a and b, under the assumption that $R > 0.2$. Algorithm 2 also calls the pre-computation step described in Algorithm 3. This step, not presented in [4], is essential to ensure the validity of Algorithm 2.

Algorithm 2 Computation of the regulator and a system of fundamental units

Input: $A_{\mathbb{R}}$ and the corresponding units β_i
Output: R and a system of fundamental units
  $R_1 \leftarrow 0$
  $i \leftarrow r-2$
  Find r linearly independent rows using Algorithm 3
  while $i < l$ **do**
    Let A be the matrix obtained by extracting any r columns and rows $i-r+2$ to i from $A_{\mathbb{R}}$.
    $R_2 \leftarrow \det A$
    Using the real GCD algorithm, compute u, v, R_3 such that

$$uR_1 + vR_2 = R_3$$

    $R_1 \leftarrow R_3$
    $\gamma_i \leftarrow \beta_i^v \times {\beta_{i-r+1}}^{(-1)^r u}$
    $i \leftarrow i+1$
  end while
  $R \leftarrow R_1$

Algorithm 3 Search for r independent rows

Input: $A_{\mathbb{R}}$
Output: A permutation of the rows of $A_{\mathbb{R}}$ such that the first r are independent
  $A_1 \leftarrow \mathbf{r}_1^{A_{\mathbb{R}}}$
  $i \leftarrow 1$
  for $i = 2$ to r **do**
    $m \leftarrow i$
    ret $\leftarrow 0$
    while ret $= 0$ **do**

$$A_i \leftarrow \begin{pmatrix} A_{i-1} \\ \hdashline \mathbf{r}_m^{A_{\mathbb{R}}} \end{pmatrix}.$$

      if $\det(A_i^t A_i) = 0$ **then**
        $m \leftarrow m+1$
      else
        Swap $\mathbf{r}_i^{A_{\mathbb{R}}}$ and $\mathbf{r}_m^{A_{\mathbb{R}}}$
        ret $\leftarrow 1$
      end if
    end while
  end for

The main loop of Algorithm 2 ensures that the sub-lattice $\mathcal{L}'_{\mathbb{R}}$ of $\mathcal{L}_{\mathbb{R}}$ corresponding to the γ_l, for $i-(r-1) \leq l \leq i$, has determinant R_3. Indeed, $\mathcal{L}'_{\mathbb{R}}$ is the sum of two sub-lattices of $\mathcal{L}_{\mathbb{R}}$ differing by a single element. The sign $(-1)^r$ is the signature of the permutation that is performed before this addition to make sure that $uR_1 + vR_2 = R_3$ holds by multilinearity of the determinant. The precomputation done

with Algorithm 3 ensures that the first determinant computed is not null, which is essential for the completeness of Algorithm 2. Whenever $\det(A_i^t A_i) \neq 0$, we have i linearly independent rows.

We postpone the computation of the complexity of Algorithms 2 and 3 to §5, where we calculate the precision we have to take for the rational approximations of the logarithms. In §5, we also ensure that this precision is accurate enough to enable us to decide whever $\det(A_i^t A_i) = 0$ or not. Algorithm 4 describes the real GCD computation. Its presentation and correctness can be found in [4].

Algorithm 4 Real GCD algorithm

Input: $R_1 = aR$ and $R_2 = bR$ with $R > 0.1$, $R_1 > R_2$ and $a, b \in \mathbb{Z}$
Output: $R_3 = dR$ and $u, v \in \mathbb{Z}$ such that $uR_1 + vR_2 = R_3$
  $u_0 \leftarrow 1$, $v_0 \leftarrow 0$
  $u_1 \leftarrow 0$, $v_1 \leftarrow 1$
  while $R_2 > 0.1$ **do**
    $q \leftarrow \lfloor R_1/R_2 \rfloor$, $r \leftarrow R_1 - R_2\lfloor R_1/R_2 \rfloor$
    $u_1 \leftarrow u_0 - qu_1$
    $v_1 \leftarrow v_0 - qv_1$
    $R_1 \leftarrow R_2$
    $R_2 \leftarrow r$
  end while
  $R_3 \leftarrow R_1$
  $u \leftarrow u_1$, $v \leftarrow v_1$

5. Approximation issues

The matrix $M_{\mathbb{R}}$ contains fixed point rational approximations $\hat{x_{ij}}$ of the logarithms of the units $x_{ij} := \log |\phi_i|_j$. In this section, we discuss the precision of the computation of the regulator. In the following, we count the precision in bits. For example, we say that $\hat{x}$ is a rational approximation of $x \in \mathbb{R}$ with precision q if $|\hat{x} - x| < 2^{-q}$. Let q_0 be the precision of the matrix $M_{\mathbb{R}}$. We have for $i \leq N + K_1 r$ and $j \leq r + 1$:

$$\hat{x_{ij}} = \sum_{k=-q_0}^{\lceil \log |x_{ij}| \rceil} 2^k a_k^{ij},$$

where the a_k^{ij} are the coefficients of the development of x_{ij} as $\sum_{k=-\infty}^{\infty} 2^k a_k^{ij}$. Before establishing the list of the steps where we might loose precision, we recall the following result that we will use to estimate the loss of precision whenever we add or multiply rational approximations:

Lemma 12. *Let $\hat{x}$ and $\hat{y}$ be rational approximations of precision q_1 of respectively x and y, and $u \in \mathbb{Z}$ such that $\lceil \log_2 u \rceil = q_2 < q_1$, then:*

- *$\hat{x} + \hat{y}$ is a rational approximation of $x + y$ of precision $q_1 - 1$.*
- *$u\hat{x}$ is a rational approximation of ux of precision $q_1 - q_2$.*
- *$\hat{x}\hat{y}$ is an approximation of xy of precision $q_1 - \max\{\log_2 |x|, \log_2 |y|\}$.*

q_0 is the precision taken for the approximation of the $\log |\phi_i|_j$. We set its value to:

$$q_0 := L(1/3, 3\rho).$$

The computation of the approximate value of each $\log|\phi_i|_j$ for $j \leq N + K_1 r$ and $j \leq r+1$ takes $O(\mathrm{M}(q_0)\log q_0) \in O\left(L(1/3, 3\rho + o(1))\right)$ bit operations [1]. As we have to perform this computation

$$(r+1)(N+K_1 r) \in O\left(L(1/3, \rho + o(1))\right)$$

times, the time taken for the creation of $M_{\mathbb{R}}$ is bounded by $O\left(L(1/3, 3\rho + o(1))\right)$. Now, let us procede with the enumeration of the steps in the algorithm that deteriorate the precision. The first source of error is the computation of the coefficients of the matrix $A_{\mathbb{R}}$. Indeed, it contains rational approximations of

$$\sum_{i=1}^{N+K_1 r} u_j^{(i)} \log|\phi_i|_j,$$

for $j = 1, \ldots, l$. The loss of precision is due to the multiplications by the $u_j^{(i)}$ and to the $N + K_1 r$ additions. We deduce from Lemma 11 the following proposition that gives us the loss of precision occuring in the computation of the coefficients of $A_{\mathbb{R}}$ with respect to the original precision taken during the construction of M:

Proposition 13. *The computation of $\sum_i u_j^{(i)} \log|\phi_i|_j$ for $j = 1, \ldots, r+1$, with precision q', requires that the precision q_0 of the $\log|\phi_i|_j$ be:*

$$q' + N + K_1 r + \max_{i,j}\left\{\log_2|u_j^{(i)}|\right\}.$$

Thus, the loss of precision during the computation of $A_{\mathbb{R}}$ is bounded by

$$O(L(1/3, \rho + o(1))).$$

Proof. Multiplying $\log_2|\phi_i|_j$ by u_i induces a loss of

$$\log_2|u_i| \in O(L(1/3, \rho + o(1)))$$

bits of precision. Furthermore every addition induces the loss of one bit of precision. As we perform $N + K_1 r = O(L(1/3, \rho + o(1)))$ of them, we thus lose another $N + K_1 r$ bits of precision. Consequently, the total loss of precision is bounded from above by:

$$N + K_1 r + \max_{i,j}\left\{\log_2|u_j^{(i)}|\right\} \in O(L(1/3, \rho + o(1))).$$

□

Once $A_{\mathbb{R}}$ is obtained, we need to compute successive $r \times r$ determinants extracted from this matrix. Every computation of such a determinant induces a loss of precision. The following proposition allows us to evaluate the loss of precision for one computation of an $r \times r$ determinant of a matrix $\hat{\Omega}$ extracted from $A_{\mathbb{R}}$.

Proposition 14. *The computation with precision q' of the determinant of an $r \times r$ matrix $\hat{\Omega}$ extracted from $A_{\mathbb{R}}$, and which is a rational approximation of $\Omega \in \mathbb{R}^{r \times r}$, requires that*

$$q = q' + (r/2+1)\log_2(r)\log_2\left(|\Omega|^{r-1} + 1\right),$$

where q is the precision of the coefficients of $A_{\mathbb{R}}$, and $|\Omega| = \max_{i,j}|\Omega_{ij}|$. Thus, the loss of precision during the computation of the determinant of an $r \times r$ submatrix of $A_{\mathbb{R}}$ is bounded by $O\left(L(1/3, \rho + o(1))\right)$.

Proof. We know that $\Omega = (\omega_1, \dots, \omega_r)$ and $\hat{\Omega} = (\hat{\omega_1}, \dots, \hat{\omega_r})$ are $r \times r$ matrices with $r \leq n \in O(\log_2(|\Delta|)^\alpha)$ and $|\Omega - \hat{\Omega}| \leq 2^{-q}$, and furthermore, by lemma 11, Ω satisfies $\log_2|\Omega| \in O(L(1/3, \rho+o(1)))$. We have by multilinearity of the determinant and by Hadamard's inequality:

$$\begin{aligned}|\det\hat{\Omega} - \det\Omega| &= |\sum_{i=1}^{r} \det(\omega_1, \dots, \omega_{i-1}, \hat{\omega}_i - \omega_i, \hat{\omega}_{i+1}, \dots, \hat{\omega}_r)| \\ &\leq r^{r/2+1}(|\Omega|^{r-1}+1)2^{-q}.\end{aligned}$$

Thus, the loss of precision is of

$$(r/2+1)\log_2(r)\log_2\left(|\Omega|^{r-1}+1\right) = O(L(1/3, \rho+o(1))).$$

□

The last source of loss of precision is the series of multiplications and additions involved in the computation of the real GCD of two approximations of multiples of the regulator. The following proposition gives us this loss of precision during the successive real GCD computations in Algorithm 2, knowing from Heuristic 6 that the real GCD need not be called more than $K_1 r$ times.

Proposition 15. *If we have the determinants of the successive $r \times r$ matrices with precision q, then we can obtain the regulator with precision q' providing*

$$q = q' + \frac{K_1 r^3}{4}\log_2^2(r)\log_2^2|A_{\mathbb{R}}|.$$

Thus, the loss of precision during the successive real GCD computations is bounded by $O\left(L(1/3, 2\rho+o(1))\right)$.

Proof. Whenever we compute another determinant R_2 of an $r \times r$ matrix extracted from $A_{\mathbb{R}}$, we have to perform the step

$$R_1 \leftarrow R_2 - R_1\lfloor R_2/R_1 \rfloor$$

at most $\log_2 R_2$ times to get the real GCD of R_1 and R_2, where R_1 is the previous approximation of the regulator R. We know that the coefficients of the submatrix whose determinant is R_2 have bit size bounded by

$$\log_2|\Omega| \leq \log_2|A_{\mathbb{R}}| \in O(L(1/3, \rho+o(1))).$$

By Hadamard's inequality, we have:

$$\log_2 R_2 \leq r/2\log_2 r \log_2|A_{\mathbb{R}}| = O(L(1/3, \rho+o(1))),$$

which gives us an upper bound on the number of times we enter the main loop of the real GCD algorithm. Every multiplication $R_1\left\lfloor\frac{R_2}{R_1}\right\rfloor$ induces the loss of at most $\log_2 R_2$ bits of precision. Thus, the total loss of precision of one call to the real GCD algorithm is of:

$$\frac{r^2}{4}\log_2^2(r)\log_2^2|A_{\mathbb{R}}| = O\left(L(1/3, 2\rho+o(1))\right).$$

As we know that Algorithm 4 is called at most $K_1 r$ times, the loss of precision after the $K_1 r$ calls for the real GCD algorithm is still of $L(1/3, 2\rho+o(1))$ bits. The last thing we have to do is to check the validity of the value $\left\lfloor\frac{R_2}{R_1}\right\rfloor$. Indeed if R_2/R_1 is close to an integer, then we risk to compute $\left\lfloor\frac{R_2}{R_1}\right\rfloor \pm 1$. Assume that $R_1 = k_1 R$ and

$R_2 = k_2 R$, with $k_1 = k_1' d$, $k_2 = k_2' d$, and with k_1' and k_2' coprime. Theoretically, we have

$$\lfloor R_2/R_1 \rfloor = \lfloor k_2'/k_1' \rfloor,$$

but we can obtain the wrong value if $k_2' \sim K k_1'$ for some integer K, the worst case scenario being $k_2' = K k_1' \pm 1$ (we cannot have $k_2' = K k_1'$ since k_1' and k_2' are coprime). In that case, we have:

$$\left| \frac{k_2'}{k_1'} - K \right| \geq \frac{1}{k_1'}.$$

Thus, we need that the precision be at most of $2\log_2 |k_1'|$. As the loss of precision encountered so far is in $O\left(L(1/3, 2\rho + o(1))\right)$, and as the original precision is in $O\left(L(1/3, 3\rho + o(1))\right)$, the current precision of the value $\left\lfloor \frac{R_2}{R_1} \right\rfloor$ is still in $O\left(L(1/3, 3\rho + o(1))\right)$. Furthermore, $\log_2 |k_1'| \leq \log_2 R_1 \leq L(1/3, \rho + o(1))$, so the condition is satisfied and the value of the quotient can be trusted. □

Corollary 16. *The total loss of precision is of:*

$$N + K_1 r + \max_{i,j} \left\{ \log_2 |u_j^{(i)}| \right\} + \frac{K_1 r^3}{4} \log_2^2(r) \log_2^2 |A_{\mathbb{R}}| \in O\left(L(1/3, 2\rho + o(1))\right).$$

These considerations allow us to evaluate the complexity of Algorithm 2. Indeed, it consists of at most $K_1 r$ computations of the determinant of an $r \times r$ submatrix $\hat{\Omega}$ of $A_{\mathbb{R}}$. Let $I \subset [1, K_2 r]$ and $J \subset [1, r+1]$ both be subsets of cardinality r such that $\hat{\Omega} =: (\hat{y_{ij}})_{i \in I, j \in J}$. In addition, we define $A = (a_{ij})_{i \in I, j \in J} \in \mathbb{Z}^{r \times r}$ such that it satisfies:

$$\hat{y_{ij}} = \sum_{k=-q}^{\lceil \log_2 |y_{ij}| \rceil} 2^k a_k^{ij} =: \frac{a_{ij}}{2^q}.$$

We thus have by multilinearity:

$$\det \hat{\Omega} = \frac{\det A}{2^{rq}},$$

where $q \in O\left(L(1/3, 3\rho + o(1))\right)$ is the precision of the coefficients of $A_{\mathbb{R}}$. Furthermore, the computation of $\det A$ takes $\tilde{O}(r^4 \log_2 |A|)$ bit operations (see [14]), where $\tilde{O}$ denotes the complexity when we omit the logarithm factors. Therefore, the expected time for the computation of $\det A$ is in

$$\tilde{O}\left(r^4 \left(q + \log_2 |\hat{\Omega}| \right) \right),$$

since $\log_2 |A| = \max_{ij} \{\log_2 a_{ij}\} \leq q + \log_2 |\hat{\Omega}|$. As q is in $O\left(L(1/3, 3\rho + o(1))\right)$, and as we know from Lemma 11 that $\log_2 |\hat{\Omega}| \in O\left(L(1/3, \rho + o(1))\right)$, we have the following result on the complexity of Algorithm 2:

Proposition 17. *The complexity of Algorithm 2 lies in*

$$O(L(1/3, 3\rho + o(1))).$$

Now, let us check the validity and the complexity of Algorithm 3. Given an $r \times i$ submatrix A_i of $A_{\mathbb{R}}$, we want to determine whether its rows are approximations of independant rows. To do this, we compute $\det(A_i^t A_i)$ and decide whether this is

the approximation of a zero determinant. We use Minkowski's bound, which states that

$$\sqrt{\det A_i^t A_i} \geq \left(\frac{\|b_1^{(i)}\|_2}{\sqrt{r}} \right)^r, \tag{7}$$

where $b_1^{(i)}$ is the non-zero vector of minimal length in the lattice spanned by the rows of A_i. For every i, $b_1^{(i)}$ is the logarithm vector of a unit. In [7], it is shown that for every unit ϵ that is not a root of unity, we have:

$$\left(\sum_i \log |\epsilon|_i^2 \right)^{1/2} > \frac{21}{128} \frac{\log n}{n^2}. \tag{8}$$

Therefore, we can prove the following proposition:

Proposition 18. *The precision $q_0 = L(1/3, 3\rho)$ is accurate enough to ensure the validity of Algorithm 3, whose complexity is in*

$$O(L(1/3, 3\rho + o(1))).$$

Proof. First, we calculate the precision of the value $\det(A_i^t A_i)$. The coefficients $c_{kl}^{(i)}$ $(k, l \leq i)$ of $A_i^t A_i$ are given by:

$$c_{kl}^{(i)} = \sum_{h \leq r} a_{kh}^{(i)} a_{lh}^{(i)},$$

where the $a_{kl}^{(i)}$ $(k \leq i, l \leq r)$ are the coefficients of A_i. We know from Lemma 11 that the coefficients of A_i have bit size in $O\left(L(1/3, \rho + o(1))\right)$, thus, using Lemma 12, we prove that the precision of $c_{kl}^{(i)}$ $(k, l \leq i)$ is still in $O(L(1/3, 3\rho + o(1)))$. Using the same techniques as in Proposition 14, we prove that the loss of precision we encounter during the computation of $\det(A_i^t A_i)$ is of

$$(i/2 + 1) \log_2(i) \log_2 \left(|A_i^t A_i|^{i-1} + 1 \right).$$

As $\log_2 |A_i^t A_i| \in O\left(L(1/3, 2\rho + o(1))\right)$, this loss of precision is in $O\left(L(1/3, 2\rho + o(1))\right)$ as well. We thus have the value of $\det(A_i^t A_i)$ with a precision q satisfying:

$$q \in O\left(L(1/3, 3\rho + o(1))\right).$$

On the other hand, we have a lower bound on the value of $\det(A_i^t A_i)$ from the combination of (7) and (8) in the case where A_i contains approximations of independent rows:

$$\det(A_i^t A_i) \geq \left(\frac{21}{128} \right)^{2r} \frac{1}{r^r} \left(\frac{\log n}{n^2} \right)^{2r}.$$

If $\det(A_i^t A_i) \leq 1/2^q$, then it might equal zero, otherwise it is necessarily the approximation of a strictly non-zero determinant. Furthermore, the bound on $\det(A_i^t A_i)$ satisfies:

$$\left| \log \left[\left(\frac{21}{128} \right)^{2r} \frac{1}{r^r} \left(\frac{\log n}{n^2} \right)^{2r} \right] \right| \leq n \log(n)(1 + o(1)) \ll q.$$

We can thus conclude that if $\det(A_i^t A_i) \leq 1/2^q$, then the rows of A_i are necessarily dependent. □

This allows us to state the following proposition:

Proposition 19. *The complexity of the computation of R and of the system of fundamental units lies in*

$$O(L(1/3, 3\rho + o(1))).$$

In addition, we know the value of R with a precision:

$$q_R \in O\left(L(1/3, 3\rho + o(1))\right).$$

6. Subexponentiality

In this section, we show that we achieve a subexponential complexity for the overall running time of the algorithm. Direct application of Proposition 5 with the parameters

$$\beta = \frac{1}{3},\ d = \rho$$
$$\zeta = \frac{2}{3},\ c = \kappa(\delta + \nu + o(1)),$$

shows that the expected number of trials to obtain a relation is at most

$$L\left(1/3, \frac{\kappa(\nu+\delta)}{3\rho} + o(1)\right).$$

We know that the factor base has size $N \in O\left(L(1/3, \rho)\right)$, thus the complexity of the search for $N + K_1 r$ relations is bounded by:

$$L\left(1/3, \frac{\kappa(\nu+\delta)}{3\rho} + \rho + o(1)\right).$$

The number of ϕ in the search space is in $O\left(L(1/3), \nu\delta\kappa\right)$. We thus have the following constraint on the parameters:

$$\nu\delta\kappa = \frac{\kappa(\nu+\delta)}{3\rho} + \rho. \tag{9}$$

We can prove that the strategy minimizing the overall time is the one where the relation collection and the linear algebra take the same time. As the complexity of the linear algebra is dominated by the HNF computation which lies is $O(L(1/3, 5\rho + o(1)))$, we thus have the additional constraint:

$$\kappa\nu\delta = 5\rho. \tag{10}$$

From (9) and (10), we obtain:

$$\nu\delta = \frac{5\rho}{\kappa}$$
$$\nu + \delta = \frac{12b^2}{\kappa}.$$

Thus, δ and ν are roots of the polynomial:

$$X^2 - \frac{24\rho^2}{\kappa}X + \frac{5\rho}{\kappa}.$$

These roots exist providing we have:

$$\rho \geq \sqrt[3]{\frac{5\kappa}{144}}.$$

The optimal choice is to minimize ρ, thus fixing the parameters δ and ν:

$$\delta = \nu = \sqrt{\frac{5\rho}{\kappa}} = \sqrt[6]{\frac{625}{144\kappa^2}}.$$

The total running time becomes $L(1/3, 5\rho + o(1))$, with:

$$\rho = \sqrt[3]{\frac{5\kappa}{144}}.$$

Acknowledgments

The author thanks Andreas Enge for his support, the fruitful discussions we had, and his careful reading of this article. He also thanks Steven Galbraith for the original suggestion of adapting the $L(1/3)$ algorithm of [6] to the context of number fields, and Michael Pohst for pointing out [7].

References

[1] R.P. Brent. Fast multiple-precision evaluation of elementary functions. *Journal of the ACM*, 23:242–251, 1976.

[2] J. Buchmann. A subexponential algorithm for the determination of class groups and regulators of algebraic number fields. In Catherine Goldstein, editor, *Séminaire de Théorie des Nombres, Paris 1988–1989*, Progress in Mathematics, pages 27–41, Boston, 1990. Birkhäuser.

[3] E.R. Canfield, P. Erdős, and C. Pomerance. On a problem of Oppenheim concerning 'factorisatio numerorum'. *J. Number Theory*, 17:1–28, 1983.

[4] H. Cohen. *A course in computational algebraic number theory*, volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag, 1991.

[5] A. Enge. Computing discrete logarithms in high-genus hyperelliptic Jacobians in provably subexponential time. *Mathematics of Computation*, 71:729–742, 2001.

[6] A. Enge and P. Gaudry. An L $(1/3 + \epsilon)$ algorithm for the discrete logarithm problem for low degree curves. In *EUROCRYPT '07: Proceedings of the 26th annual international conference on Advances in Cryptology*, Lecture Notes in Computer Science, pages 379–393, Berlin, Heidelberg, 2007. Springer-Verlag.

[7] C. Fieker and M. Pohst. Dependency of units in number fields. *Mathematics of Computation*, 75:1507–1518, 2006.

[8] M. Giesbrecht, M. Jacobson, and A. Storjohann. Algorithms for large integer matrix problems. In S. Boztas and I. Shparlinski, editors, *Proceedings of the 14th International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, AAECC-14*, volume 2227 of *Lecture Notes in Computer Science*, pages 297–307, Heidelberg, 2001. Springer Verlag.

[9] J.L. Hafner and K.S. McCurley. A rigorous subexponential algorithm for computation of class groups. *Journal of American Society*, 2:839–850, 1989.

[10] A.K. Lenstra. On the calculation of regulators and class numbers of quadratic fields. In *Journées arithmétiques*, pages 123–150. Cambridge Univ. Press, 1982.

[11] M. Mignotte. An inequality about factors of polynomials. *Mathematics of Computation*, 28:1153–1157, 1974.

[12] D. Shanks. Class number, a theory of factorization, and genera. In W. J. LeVeque and E. G. Straus, editors, *Proceedings of Symposia in Pure Mathematics*, volume 20, pages 415–440. American Mathematical Society, 1969.

[13] D. Shanks. The infrastructure of a real quadratic field and its applications. In *Proceedings of the 1972 Number Theory Conference*, pages 217–224. Boulder: University of Colorado, 1972.

[14] A. Storjohann. *Algorithms for Matrix Canonical Forms*. PhD thesis, Department of Computer Science, Swiss Federal Institute of Technology – ETH, 2000.

LIX , École Polytechnique , 91128 PALAISEAU , France
E-mail address: biasse@lix.polytechnique.fr