COMPUTING ALGEBRAIC NUMBERS OF BOUNDED HEIGHT

JOHN R. DOYLE AND DAVID KRUMM

ABSTRACT. We describe an algorithm which, given a number field K and a bound B, finds all the elements of K having relative height at most B. Two lists of numbers are computed: one consisting of elements $x \in K$ for which it is known with certainty that $H_K(x) \leq B$, and one containing elements x such that $|H_K(x) - B| < \theta$ for a tolerance θ chosen by the user. We show that every element of K whose height is at most B must appear in one of the two lists.

1. INTRODUCTION

Let K be a number field with ring of integers $\mathcal{O}_K$, and let H_K be the relative height function on K. For any bound B it is known that the set of all elements $x \in K$ with $H_K(x) \leq B$ is finite [19, §3.1]. Moreover, there is an asymptotic formula for the number of such elements, due to Schanuel [18]:

$$\#\{x \in K : H_K(x) \leq B\} \sim C_K B^2,$$

where C_K is an explicit constant which depends only on K. However, there does not appear to be in the literature an algorithm that would allow fast computation of all these elements. In [15] Pethő and Schmitt require such an algorithm to be able to compute the Mordell-Weil groups of certain elliptic curves over real quadratic fields. They obtain an algorithm by showing that if $\omega_1, \ldots, \omega_n$ is an LLL-reduced integral basis of K, then every element $x \in K$ with $H_K(x) \leq B$ can be written as

$$x = \frac{a_1\omega_1 + \cdots + a_n\omega_n}{c},$$

where $a_1, \ldots, a_n, c$ are integers within certain explicit bounds depending only on B and K. The set of all such numbers x is finite, so one would only need to search through this set and discard elements whose height is greater than B. Unfortunately, in practice this method is slow because the search space is very large. We describe in this paper an algorithm which is faster, assuming class group representatives for $\mathcal{O}_K$ and a basis for the unit group of $\mathcal{O}_K$ can be computed efficiently. Sample computations showing the greater speed of our method may be seen in §6.

Our motivation for designing a fast algorithm that can handle relatively large bounds B comes from arithmetic dynamics. In [16] Poonen provides a conjecturally complete list of rational preperiodic graph structures for quadratic polynomial maps defined over $\mathbb{Q}$. It is then natural to ask which preperiodic graph structures can occur for such maps over other number fields. In order to gather data about these graphs one needs to be able to compute all the preperiodic points of a given quadratic polynomial. It is possible to give an explicit upper bound for the height of any preperiodic point of a given map, so a first step towards computing preperiodic points is computing all points of bounded height. Further details on this question, together with the generated data, will be presented in a subsequent paper [8].

Acknowledgements. We thank Xander Faber, Dino Lorenzini, and Andrew Sutherland for their comments on earlier versions of this paper; Pete Clark for help finding references; Robert Rumely for suggestions on improving the efficiency of the algorithm; and the referee for several helpful comments.

2. Background and notation

Let K be a number field; let $\sigma_1, \ldots, \sigma_{r_1}$ be the real embeddings of K, and $\tau_1, \overline{\tau}_1, \ldots, \tau_{r_2}, \overline{\tau}_{r_2}$ the complex embeddings. Corresponding to each of these embeddings there is an archimedean absolute value on K extending the usual absolute value on $\mathbb{Q}$. For an embedding σ, the corresponding absolute value $|\ |_\sigma$ is given by $|x|_\sigma = |\sigma(x)|_{\mathbb{C}}$, where $|\cdot|_{\mathbb{C}}$ is the usual complex absolute value. Note that $|\ |_{\overline{\tau}_i} = |\ |_{\tau_i}$ for every i. We will denote by M_K^∞ the set of absolute values corresponding to $\sigma_1, \ldots, \sigma_{r_1}, \tau_1, \ldots, \tau_{r_2}$.

For every maximal ideal $\mathfrak{p}$ of the ring of integers $\mathcal{O}_K$ there is a discrete valuation $v_\mathfrak{p}$ on K with the property that for every $a \in K^*$, $v_\mathfrak{p}(a)$ is the power of $\mathfrak{p}$ dividing the principal ideal (a). If $\mathfrak{p}$ lies over the prime p of $\mathbb{Z}$, there is an absolute value $|\ |_\mathfrak{p}$ on K extending the p-adic absolute value on $\mathbb{Q}$. Let $e(\mathfrak{p})$ and $f(\mathfrak{p})$ denote the ramification index and residual degree of $\mathfrak{p}$, respectively. This absolute value is then given by $|x|_\mathfrak{p} = (N\mathfrak{p})^{-v_\mathfrak{p}(x)/(e(\mathfrak{p})f(\mathfrak{p}))}$. We denote by M_K^0 the set of all absolute values $|\ |_\mathfrak{p}$, and we let $M_K = M_K^\infty \cup M_K^0$.

For an absolute value $v \in M_K$, let K_v be the completion of K with respect to v, and let $\mathbb{Q}_v$ be the completion of $\mathbb{Q}$ with respect to the restriction of v to $\mathbb{Q}$. Note that $\mathbb{Q}_v = \mathbb{R}$ if $v \in M_K^\infty$, and $\mathbb{Q}_v = \mathbb{Q}_p$ if $v \in M_K^0$ corresponds to a maximal ideal $\mathfrak{p}$ lying over p. The *local degree* of K at v is given by $n_v = [K_v : \mathbb{Q}_v]$. If v corresponds to a real embedding of K, then $K_v = \mathbb{Q}_v = \mathbb{R}$, so $n_v = 1$. If v corresponds to a complex embedding of K, then $K_v = \mathbb{C}$ and $\mathbb{Q}_v = \mathbb{R}$, so $n_v = 2$. Finally, if v corresponds to a maximal ideal $\mathfrak{p}$, then $n_v = e(\mathfrak{p})f(\mathfrak{p})$.

The *relative height* function $H_K : K \to \mathbb{R}_{\geq 1}$ is defined by

$$H_K(\gamma) = \prod_{v \in M_K} \max\{|\gamma|_v^{n_v}, 1\}$$

and has the following properties:

- For any $\alpha, \beta \in K$ with $\beta \neq 0$, $H_K(\alpha/\beta) = \prod_{v \in M_K} \max\{|\alpha|_v^{n_v}, |\beta|_v^{n_v}\}$.
- For any $\alpha, \beta \in K$, $H_K(\alpha\beta) \leq H_K(\alpha)H_K(\beta)$.
- For any $\alpha, \beta \in \mathcal{O}_K$ with $\beta \neq 0$, $H_K(\alpha/\beta) = N(\alpha, \beta)^{-1} \prod_{v \in M_K^\infty} \max\{|\alpha|_v^{n_v}, |\beta|_v^{n_v}\}$. Here $N(\alpha, \beta)$ denotes the norm of the ideal generated by α and β.
- For any $\gamma \in K^*$, $H_K(\gamma) = H_K(1/\gamma)$.
- For any $\gamma \in K$ and any root of unity $\zeta \in K$, $H_K(\zeta\gamma) = H_K(\gamma)$.

It will sometimes be convenient to use the logarithmic height function $h_K = \log \circ H_K$.

The following notation will be used throughout: $\mathcal{O}_K^\times$ is the unit group of $\mathcal{O}_K$, μ_K is the group of roots of unity in K, $r = r_1 + r_2 - 1$ is the rank of $\mathcal{O}_K^\times$, h is the class number of K, and Δ_K is the discriminant of K. For an ideal I of $\mathcal{O}_K$ we let $N(I) := \#(\mathcal{O}_K/I)$ denote the norm of the ideal.

Define a logarithmic map $\Lambda : K^* \to \mathbb{R}^{r+1}$ by

$$\Lambda(x) = (\log |x|_v^{n_v})_{v \in M_K^\infty} = \left(\log |x|_{\sigma_1}, \ldots, \log |x|_{\sigma_{r_1}}, \log |x|_{\tau_1}^2, \ldots, \log |x|_{\tau_{r_2}}^2\right).$$

Note that Λ is a group homomorphism. By a classical result of Kronecker, the kernel of Λ is μ_K. Letting $\pi : \mathbb{R}^{r+1} \to \mathbb{R}^r$ be the projection map that deletes the last coordinate, we set $\Lambda' = \pi \circ \Lambda$.

Recall that there is a system $\boldsymbol{\varepsilon} = \{\varepsilon_1, \ldots, \varepsilon_r\} \subset \mathcal{O}_K^\times$ of *fundamental units* such that every unit $u \in \mathcal{O}_K^\times$ can be written uniquely as $u = \zeta \varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}$ for some integers $n_1, \ldots, n_r$ and some $\zeta \in \mu_K$. We denote by $S(\boldsymbol{\varepsilon})$ the $r \times r$ matrix with column vectors $\Lambda'(\varepsilon_j)$.

3. The method

Let K be a number field, and let $H_K : K \to \mathbb{R}_{\geq 1}$ be the relative height function on K. Given a bound $B \geq 1$, we want to list the elements $\gamma \in K$ satisfying $H_K(\gamma) \leq B$. Our method for finding all such numbers is based on the observation that, using the ideal class group and unit group of $\mathcal{O}_K$, this problem can be reduced to the question of finding all units of bounded height. In essence, the idea is to generalize the following statement that holds over $\mathbb{Q}$: if $x \in \mathbb{Q}^*$ and $H_{\mathbb{Q}}(x) \leq B$, then x can be written as $x = \pm a/b$ where a and b are integers such that $(a, b) = 1$ and $|a|, |b| \leq B$. For a general number field K, the analogous statement we make is that given $x \in K^*$ with $H_K(x) \leq B$, it is possible to write x in the form $x = u \cdot a/b$, where $u \in \mathcal{O}_K^\times$ is a unit whose height is explicitly bounded; a and b are elements of $\mathcal{O}_K$ such that $(a, b) = \mathfrak{a}$, where $\mathfrak{a}$ is one ideal from a predetermined list of ideal class representatives for $\mathcal{O}_K$; and $|N_{K/\mathbb{Q}}(a)|, |N_{K/\mathbb{Q}}(b)| \leq B \cdot N(\mathfrak{a})$.

3.1. **The main algorithm.** Theorem 3.1 below provides the theoretical basis for our algorithm. In order to describe all elements of bounded height in K we fix integral ideals $\mathfrak{a}_1, \ldots, \mathfrak{a}_h$ forming a complete set of ideal class representatives for $\mathcal{O}_K$, and we fix fundamental units $\varepsilon_1, \ldots, \varepsilon_r$. Suppose we are given a bound $B \geq 1$. For each ideal $\mathfrak{a}_\ell$, let $g_{\ell,1}, \ldots, g_{\ell,s_\ell}$ be generators for all the nonzero principal ideals contained in $\mathfrak{a}_\ell$ whose norms are at most $B \cdot N(\mathfrak{a}_\ell)$. We define a *B-packet* to be a tuple of the form

$$P = (\ell, (i, j), (n_1, \ldots, n_r))$$

satisfying the following conditions:

- $1 \leq \ell \leq h$;
- $1 \leq i < j \leq s_\ell$;
- $(g_{\ell,i}, g_{\ell,j}) = \mathfrak{a}_\ell$; and
- $H_K(\varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}) \leq B \cdot H_K(g_{\ell,i}/g_{\ell,j})$.

To a packet P we associate the number

$$c(P) = \varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r} \cdot \frac{g_{\ell,i}}{g_{\ell,j}} \in K^* \backslash \mathcal{O}_K^\times$$

and the set

$$F(P) = \{\zeta \cdot c(P) : \zeta \in \mu_K\} \cup \{\zeta / c(P) : \zeta \in \mu_K\}.$$

Note that the union defining $F(P)$ is disjoint, and that $F(P)$ does not contain units. Moreover, all the elements of $F(P)$ have the same height. If $r = 0$, then a packet is a tuple of the form $(\ell, (i, j))$ satisfying only the first three defining conditions above, and in this case $c(P) = g_{\ell,i}/g_{\ell,j}$.

With the notation and terminology introduced above we can now describe all elements of K whose height is at most B.

Theorem 3.1. *Suppose that $\gamma \in K^*$ satisfies $H_K(\gamma) \leq B$. Then either $\gamma \in \mathcal{O}_K^\times$ or γ belongs to the disjoint union*

$$\bigcup_{B\text{-packets } P} F(P).$$

Proof. Assuming that $\gamma \notin \mathcal{O}_K^\times$ we must show that there is a packet P such that $\gamma \in F(P)$. We can write the fractional ideal generated by γ as $(\gamma) = IJ^{-1}$, where I and J are coprime integral ideals. Since I and J are in the same ideal class, there is some ideal $\mathfrak{a}_\ell$ (namely the one representing the inverse class of I and J) such that $\mathfrak{a}_\ell I$ and $\mathfrak{a}_\ell J$ are principal; say $(\alpha) = \mathfrak{a}_\ell I$, $(\beta) = \mathfrak{a}_\ell J$. Note that $(\alpha, \beta) = \mathfrak{a}_\ell$ because I and J are coprime. Since $(\gamma) = (\alpha)(\beta)^{-1}$ we may assume, after scaling α by a unit, that $\gamma = \alpha/\beta$. From the bound $H_K(\gamma) \leq B$ it follows that

$$\prod_{v \in M_K^\infty} \max\{|\alpha|_v^{n_v}, |\beta|_v^{n_v}\} \leq B \cdot N(\mathfrak{a}_\ell).$$

In particular,

$$|N_{K/\mathbb{Q}}(\alpha)| = \prod_{v \in M_K^\infty} |\alpha|_v^{n_v} \leq B \cdot N(\mathfrak{a}_\ell) \quad \text{and} \quad |N_{K/\mathbb{Q}}(\beta)| = \prod_{v \in M_K^\infty} |\beta|_v^{n_v} \leq B \cdot N(\mathfrak{a}_\ell).$$

Since $N(\alpha), N(\beta) \leq B \cdot N(\mathfrak{a}_\ell)$, there must be some indices $a, b \leq s_\ell$ such that $(\alpha) = (g_{\ell,a})$ and $(\beta) = (g_{\ell,b})$. Hence, we have $\alpha = g_{\ell,a}u_a$ and $\beta = g_{\ell,b}u_b$ for some units u_a, u_b. Letting $t = u_a/u_b$ we have $\gamma = tg_{\ell,a}/g_{\ell,b}$, and since $H_K(\gamma) \leq B$, then

$$H_K(t) = H_K(\gamma g_{\ell,b}/g_{\ell,a}) \leq H_K(\gamma)H_K(g_{\ell,b}/g_{\ell,a}) \leq B \cdot H_K(g_{\ell,b}/g_{\ell,a}).$$

Write $t = \zeta\varepsilon_1^{m_1} \cdots \varepsilon_r^{m_r}$ for some integers $m_1, \ldots, m_r$ and some $\zeta \in \mu_K$. We define indices i, j and an integer tuple $(n_1, \ldots, n_r)$ as follows: if $a < b$, we let $i = a, j = b, (n_1, \ldots, n_r) = (m_1, \ldots, m_r)$; and if $a > b$, we let $i = b, j = a, (n_1, \ldots, n_r) = (-m_1, \ldots, -m_r)$. (The case $a = b$ cannot occur since γ is not a unit.) Note that in either case we have $i < j$ and $(g_{\ell,i}, g_{\ell,j}) = (\alpha, \beta) = \mathfrak{a}_\ell$. Letting $u = \varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}$ we have $H_K(u) = H_K(t)$, so $H_K(u) \leq B \cdot H_K(g_{\ell,i}/g_{\ell,j})$. This proves that $P := (\ell, (i, j), (n_1, \ldots, n_r))$ is a B-packet. Finally, if we set $c = ug_{\ell,i}/g_{\ell,j}$, then $\zeta c = \gamma$ if $a < b$; and $\zeta/c = \gamma$ if $a > b$. Therefore, $\gamma \in F(P)$.

We show now that the union in the statement of the theorem is disjoint. Suppose that

$$P = (\ell, (i, j), (n_1, \ldots, n_r)) \quad \text{and} \quad P' = \big(\ell', (i', j'), (n_1', \ldots, n_r')\big)$$

are packets such that $F(P) \cap F(P') \neq \emptyset$. We aim to show that $P = P'$. Let $u = \varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}$, and similarly define u'. From the assumption that $F(P)$ and $F(P')$ have a common element it follows that either

$$c(P) \cdot c(P') \in \mu_K \quad \text{or} \quad c(P)/c(P') \in \mu_K.$$

We consider the latter case first. There are ideals $\mathfrak{b}_{\ell,i}, \mathfrak{b}_{\ell,j}, \mathfrak{b}_{\ell',i'}, \mathfrak{b}_{\ell',j'}$ such that

$$(g_{\ell,i}) = \mathfrak{a}_\ell\mathfrak{b}_{\ell,i}\ ;\ (g_{\ell,j}) = \mathfrak{a}_\ell\mathfrak{b}_{\ell,j}\ ;\ (g_{\ell',i'}) = \mathfrak{a}_{\ell'}\mathfrak{b}_{\ell',i'}\ ;\ (g_{\ell',j'}) = \mathfrak{a}_{\ell'}\mathfrak{b}_{\ell',j'}\ . \tag{3.1}$$

Note that $\mathfrak{b}_{\ell,i}$ and $\mathfrak{b}_{\ell,j}$ are coprime because $(g_{\ell,i}, g_{\ell,j}) = \mathfrak{a}_\ell$; similarly, $\mathfrak{b}_{\ell',i'}$ and $\mathfrak{b}_{\ell',j'}$ are coprime. Now, since $c(P)/c(P') \in \mu_K$, there is an equality of ideals $(g_{\ell,i})(g_{\ell',j'}) = (g_{\ell,j})(g_{\ell',i'})$. Therefore, $\mathfrak{b}_{\ell,i}\mathfrak{b}_{\ell',j'} = \mathfrak{b}_{\ell,j}\mathfrak{b}_{\ell',i'}$ and by coprimality we conclude that

$$\mathfrak{b}_{\ell,i} = \mathfrak{b}_{\ell',i'} \quad \text{and} \quad \mathfrak{b}_{\ell,j} = \mathfrak{b}_{\ell',j'}. \tag{3.2}$$

Considering ideal classes, by (3.1) and (3.2) we obtain

$$[\mathfrak{a}_\ell]^{-1} = [\mathfrak{b}_{\ell,i}] = [\mathfrak{b}_{\ell',i'}] = [\mathfrak{a}_{\ell'}]^{-1},$$

so $\ell = \ell'$. Thus, again using (3.1) and (3.2),

$$(g_{\ell,i}) = \mathfrak{a}_\ell\mathfrak{b}_{\ell,i} = \mathfrak{a}_{\ell'}\mathfrak{b}_{\ell',i'} = (g_{\ell',i'}) = (g_{\ell,i'}),$$

and hence $i = i'$; similarly, $j = j'$. It follows that $u/u' = c(P)/c(P') \in \mu_K$, so $(n_1, \ldots, n_r) = (n_1', \ldots, n_r')$, and therefore $P = P'$.

The case where $c(P) \cdot c(P') \in \mu_K$ is dealt with similarly, and leads to the conclusion that $(i, j) = (j', i')$. But this is a contradiction, since $i < j$ and $i' < j'$; therefore, this case cannot occur. □

Remark. In the case where $r = 0$, Theorem 3.1 and its proof still hold if we omit mention of the fundamental units. See §4.4 for a refinement of the theorem in this case.

From Theorem 3.1 we deduce the following algorithm.

Algorithm 1 (Algebraic numbers of bounded height)**.**
Input: A number field K and a bound $B \geq 1$.
Output: A list of all elements $x \in K$ satisfying $H_K(x) \leq B$.

(1) Create a list L containing only the element 0.

(2) Determine a complete set $\{\mathfrak{a}_1, \ldots, \mathfrak{a}_h\}$ of ideal class representatives for $\mathcal{O}_K$.

(3) Compute a system $\boldsymbol{\varepsilon} = \{\varepsilon_1, \ldots, \varepsilon_r\}$ of fundamental units.

(4) Include in L all units $u \in \mathcal{O}_K^\times$ with $H_K(u) \le B$.

(5) For each ideal $\mathfrak{a}_\ell$:
   (a) Find generators $g_{\ell,1}, \ldots, g_{\ell,s_\ell}$ for all the nonzero principal ideals contained in $\mathfrak{a}_\ell$ whose norms are at most $B \cdot N(\mathfrak{a}_\ell)$.
   (b) For each pair of indices i, j such that $1 \le i < j \le s_\ell$ and $(g_{\ell,i}, g_{\ell,j}) = \mathfrak{a}_\ell$:
      (i) Find all units u of the form $u = \varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}$ such that $H_K(u) \le B \cdot H_K(g_{\ell,i}/g_{\ell,j})$.
      (ii) For all such units u, let $c = u \cdot g_{\ell,i}/g_{\ell,j}$. If $H_K(c) \le B$, then append to L all elements of the form $\zeta \cdot c$ and ζ/c with $\zeta \in \mu_K$.

(6) Return the list L.

Note that, by Theorem 3.1, the list L will not contain duplicate elements. There are known methods for carrying out steps (2) and (3) of Algorithm 1 — see, for instance, [6, §6.5]. To do step 5(a) one can use methods for finding elements in $\mathcal{O}_K$ of given norm; one algorithm for doing this can be found in [9]. It remains to explain how a set of units of bounded height can be computed.

3.2. **Units of bounded height.** For a given bound $D \ge 1$ we wish to determine all units $u \in \mathcal{O}_K^\times$ such that $H_K(u) \le D$. Our method for doing this makes use of the following classical result.

Theorem 3.2 (Dirichlet)**.** *The map $\Lambda' : \mathcal{O}_K^\times \to \mathbb{R}^r$ is a group homomorphism with kernel μ_K, and $\Lambda'(\mathcal{O}_K^\times)$ is a lattice of full rank in $\mathbb{R}^r$ spanned by the vectors $\Lambda'(\varepsilon_1), \ldots, \Lambda'(\varepsilon_r)$.*

Let $S = S(\boldsymbol{\varepsilon})$ be the $r \times r$ matrix with column vectors $\Lambda'(\varepsilon_j)$, and let $T = S^{-1}$ be the linear automorphism of $\mathbb{R}^r$ taking the basis $\Lambda'(\varepsilon_1), \ldots, \Lambda'(\varepsilon_r)$ to the standard basis for $\mathbb{R}^r$.

Proposition 3.3. *Suppose $u \in \mathcal{O}_K^\times$ satisfies $H_K(u) \le D$. Then there exist an integer point $(n_1, \ldots, n_r)$ in the polytope $T([-\log D, \log D]^r)$ and a root of unity $\zeta \in \mu_K$ such that $u = \zeta \varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}$.*

Proof. The bound $H_K(u) \le D$ implies that $|u|_v^{n_v} \le D$ for all $v \in M_K^\infty$. Since $H_K(1/u) = H_K(u)$ we also have $1/|u|_v^{n_v} \le D$. Therefore,

$$-\log D \le \log |u|_v^{n_v} \le \log D \quad \text{for all } v \in M_K^\infty,$$

so $\Lambda'(u) \in [-\log D, \log D]^r$. We can write $u = \zeta \varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}$ for some $\zeta \in \mu_K$ and some integers n_i. Then $(n_1, \ldots, n_r) = T(\Lambda'(u)) \in T([-\log D, \log D]^r)$. □

The above proposition leads to the following algorithm.

Algorithm 2 (Units of bounded height)**.**
Input: A number field K and a bound $D \ge 1$.
Output: A list of all units $u \in \mathcal{O}_K^\times$ satisfying $H_K(u) \le D$.

(1) If $r = 0$, return μ_K. Otherwise:

(2) Create an empty list U.

(3) Compute fundamental units $\varepsilon_1, \ldots, \varepsilon_r$.

(4) Find all integer points Q in the polytope $T([-\log D, \log D]^r)$.

(5) For all such points $Q = (n_1, \ldots, n_r)$:
   (a) Let $u = \varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}$.
   (b) If $H_K(u) \le D$, then include $u\zeta$ in U for all $\zeta \in \mu_K$.

(6) Return the list U.

Step (4) of Algorithm 2 can be done using known methods for finding integer points in polytopes; see the articles [1, 7].

Remark. With more work it is possible to replace the box $[-\log D, \log D]^r$ in step 4 of Algorithm 2 with a substantially smaller set, namely the polytope $\mathcal{P}(D)$ in $\mathbb{R}^r$ cut out by the inequalities

$$-\log D \leq \sum_{i \in I} x_i \leq \log D,$$

where I runs through all nonempty subsets of $\{1, \ldots, r\}$. This polytope is contained in the box $[-\log D, \log D]^r$, and one can show that its volume is smaller than that of the box by a factor of at least $(\lfloor r/2 \rfloor !)^2$. In addition to providing a smaller search space, using $\mathcal{P}(D)$ eliminates the need to check the heights of the units obtained. This is due to the fact that for units u, $H_K(u) \leq D$ if and only if $\Lambda'(u) \in \mathcal{P}(D)$. We omit the proofs of these statements since we will not use the polytope $\mathcal{P}(D)$ here; the box $[-\log D, \log D]^r$ works well in practice and will suffice for a theoretical analysis of the main algorithm.

For later reference we record the following facts concerning units of bounded height.

Lemma 3.4. *If the unit $u = \zeta \varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}$ satisfies $H_K(u) \leq D$, then*

$$\max_{1 \leq i \leq r} |n_i| \leq M := \lfloor \|T\| \cdot \sqrt{r} \cdot \log D \rfloor,$$

where $\|T\|$ denotes the operator norm of T.

Proof. By Proposition 3.3, the point $(n_1, \ldots, n_r)$ belongs to $T([-\log D, \log D]^r)$, and therefore has Euclidean norm bounded by $\|T\| \cdot \sqrt{r} \cdot \log D$. Since each n_i is an integer, it follows that $|n_i| \leq M$ for all i. □

Using Lemma 3.4 we can deduce upper bounds for the number of units of bounded height.

Corollary 3.5. *Fix $\kappa > 0$. There is a constant $q = q(\kappa, K, \boldsymbol{\varepsilon})$ such that for every bound $D \geq 1 + \kappa$, the number of units $u \in \mathcal{O}_K^\times$ satisfying $H_K(u) \leq D$ is at most $q \cdot (\log D)^r$.*

Proof. Suppose $u = \zeta \varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}$ is a unit with $H_K(u) \leq D$. By Lemma 3.4, $(n_1, \ldots, n_r) \in [-M, M]^r$. This gives at most $(2M+1)^r$ options for the tuple $(n_1, \ldots, n_r)$, and hence at most $(\#\mu_K) \cdot (2M+1)^r$ options for u. Therefore,

$$\frac{\#\{u \in \mathcal{O}_K^\times : H_K(u) \leq D\}}{(\log D)^r} \leq (\#\mu_K) \left(2\|T\| \cdot \sqrt{r} + \frac{1}{\log(1+\kappa)} \right)^r,$$

and the result follows. □

While Algorithms 1 and 2 form a theoretically accurate description of our method, for purposes of explicit computation they are not optimal. We discuss now a few changes to our method which will make it more efficient.

3.3. **Computational improvements to the method.** We aim in this section to modify Algorithms 1 and 2 with the following goals in mind: to avoid computing any given piece of data more than once; to minimize the cost of height computations; and to avoid, as much as possible, doing arithmetic with fundamental units. The latter is desirable because fundamental units in a number field can be very large, so that arithmetic operations with them might be costly.

Regarding the expense of height computations, we begin by noting that the height of an element of K can be computed by using the logarithmic map Λ. Indeed, suppose α, β are nonzero elements of $\mathcal{O}_K$; letting $\Lambda(\alpha) = (x_1, \ldots, x_{r+1})$ and $\Lambda(\beta) = (y_1, \ldots, y_{r+1})$ we have

$$\log(N(\alpha,\beta)) + h_K(\alpha/\beta) = \sum_{i=1}^{r+1} \max\{x_i, y_i\}. \tag{3.3}$$

In view of this fact, throughout this section we will use the logarithmic height function h_K rather than H_K. From a computational standpoint, h_K is also more convenient because it is defined as a sum rather than a product.

To minimize the amount of time spent on height computations in step (5) of Algorithm 1, we make the following observation: using (3.3), all of the heights required in that step can be computed from the data of the real vectors $\Lambda(\varepsilon_1), \ldots, \Lambda(\varepsilon_r)$ and the vectors $\Lambda(g_{\ell,i})$ for all indices ℓ, i:

- The height of a unit $u = \varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}$ can be found knowing only the tuple $(n_1, \ldots, n_r)$ — without actually computing u. Indeed, $h_K(u)$ can be computed from the vector $\Lambda(u)$, which is equal to $\sum_{j=1}^{r} n_j \Lambda(\varepsilon_j)$.
- The numbers $h_K(g_{\ell,i}/g_{\ell,j})$ required in step 5(b)(i) of Algorithm 1 can be computed from the vectors $\Lambda(g_{\ell,i})$ and $\Lambda(g_{\ell,j})$.
- Using the fact that $\Lambda(u \cdot g_{\ell,i}) = \Lambda(u) + \Lambda(g_{\ell,i})$, the number $h_K(u \cdot g_{\ell,i}/g_{\ell,j})$ in step 5(b)(ii) can be computed from the tuple $(n_1, \ldots, n_r)$ and the vectors $\Lambda(g_{\ell,i})$, $\Lambda(g_{\ell,j})$.

From these observations we conclude that the vectors $\Lambda(\varepsilon_1), \ldots, \Lambda(\varepsilon_r)$ and $\Lambda(g_{\ell,i})$ (for all appropriate indices ℓ, i) should be computed once and stored for later use; all height computations that take place within the algorithm can then make use of this precomputed data.

Step 5(b)(i) of Algorithm 1, in which we compute all units of height less than a given bound, must be performed many times — each time with a different height bound. It would be more efficient to let d be the largest height bound considered, and determine the list U of units u satisfying $h_K(u) \leq d$. This list will then contain all units needed throughout the algorithm. In particular, the units from step (4) can be obtained from U. Hence, step (4) should be carried out only after the list U has been computed. By making these changes, only one computation of units of bounded height will be required throughout the entire algorithm.

Finally, in order to speed up computations involving the vectors $\Lambda'(\varepsilon_1), \ldots, \Lambda'(\varepsilon_r)$, it will be useful to assume that these vectors form a suitably reduced basis for the lattice $\Lambda'(\mathcal{O}_K^\times) \subset \mathbb{R}^r$. We will therefore apply the LLL algorithm [11] to obtain a reduced basis.

With all of the above modifications in mind we now give an improved version of our method.

Algorithm 3 (Algebraic numbers of bounded height)**.**
Input: A number field K and a bound $B \geq 1$.
Output: A list L of all elements $\gamma \in K$ satisfying $H_K(\gamma) \leq B$.

(1) Find a complete set $\{\mathfrak{a}_1, \ldots, \mathfrak{a}_h\}$ of ideal class representatives for $\mathcal{O}_K$ and compute an LLL-reduced basis $\varepsilon = \{\varepsilon_1, \ldots, \varepsilon_r\}$ of fundamental units in $\mathcal{O}_K$. Compute the numbers $\log N(\mathfrak{a}_\ell)$ for every index ℓ, and the vectors $\Lambda(\varepsilon_j)$ for every j.

(2) Construct the set

$$\mathcal{N} := \bigcup_{\ell=1}^{h} \{m \cdot N(\mathfrak{a}_\ell) : m \leq B\}$$

and make a list $\mathcal{P}$ of all nonzero principal ideals of $\mathcal{O}_K$, each represented by a single generator g, having norm in $\mathcal{N}$. Record $\Lambda(g)$ for each g.

(3) For each ideal $\mathfrak{a}_\ell$, make a list $(g_{\ell,1}), \ldots, (g_{\ell,s_\ell})$ of all elements of $\mathcal{P}$ contained in $\mathfrak{a}_\ell$ whose norms are at most $B \cdot N(\mathfrak{a}_\ell)$.

(4) For each index ℓ:
  (a) Make a list R_ℓ of pairs (i,j) such that $1 \leq i < j \leq s_\ell$ and $(g_{\ell,i}, g_{\ell,j}) = \mathfrak{a}_\ell$.
  (b) For each pair (i,j) in R_ℓ, use the data recorded in step (2) to compute $h_{\ell,i,j} = h_K(g_{\ell,i}/g_{\ell,j})$.

(5) Let $d = \log B + \max_\ell \max_{(i,j)\in R_\ell} h_{\ell,i,j}$.

(6) Construct the matrix $S = S(\boldsymbol{\varepsilon})$ with column vectors $\Lambda'(\varepsilon_1), \ldots, \Lambda'(\varepsilon_r)$ and compute its inverse.

(7) Construct a list U consisting of all integer vectors $(n_1, \ldots, n_r)$ in the polytope $S^{-1}([-d,d]^r)$.

(8) Create a list L containing only the element 0, and create empty lists U_0 and L_0.

(9) For each tuple $\boldsymbol{u} = (n_1, \ldots, n_r)$ in U:
  (a) Compute the vector $\Lambda_{\boldsymbol{u}} = \sum_{j=1}^r n_j \Lambda(\varepsilon_j)$ using the data from step (1).
  (b) Use $\Lambda_{\boldsymbol{u}}$ to compute $h_{\boldsymbol{u}} = h_K\left(\varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}\right)$.
  (c) If $h_{\boldsymbol{u}} \leq \log B$, then append $\boldsymbol{u}$ to U_0.
  (d) If $h_{\boldsymbol{u}} > d$, then remove $\boldsymbol{u}$ from U.

(10) For each index ℓ :
  For each pair $(i,j) \in R_\ell$:
    Let $w = \log B + h_{\ell,i,j}$.
    For each tuple $\boldsymbol{u} = (n_1, \ldots, n_r)$ in U:
      If $h_{\boldsymbol{u}} \leq w$, then:
        (i) Let P be the packet $(\ell, (i,j), (n_1, \ldots, n_r))$.
        (ii) Use the data recorded in steps (2) and 9(a) to compute $h_K(c(P))$.
        (iii) If $h_K(c(P)) \leq \log B$, then append the packet P to L_0.

(11) Make a list consisting of the distinct tuples $(n_1, \ldots, n_r)$ appearing in U_0 or in some packet $P \in L_0$. Compute and store all units of the form $\varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}$ with $(n_1, \ldots, n_r)$ in this list.

(12) For each tuple $(n_1, \ldots, n_r)$ in U_0, append to L all numbers of the form $\zeta \varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}$ with $\zeta \in \mu_K$. Use the units computed in step (11).

(13) For each packet P in L_0, compute the number $c(P)$ using the data from step (11), and append to L all numbers of the form $\zeta \cdot c(P)$ and $\zeta / c(P)$ with $\zeta \in \mu_K$.

(14) Return the list L.

Remarks.

- Step (11) is done in order to avoid computing the same unit more than once: it may well happen that two distinct packets P and P' contain the same tuple $(n_1, \ldots, n_r)$, so that when computing $c(P)$ and $c(P')$ we would carry out the multiplication $\varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}$ twice. Similarly, a tuple from the list U_0 may appear in some packet P. By doing step (11) we are thus avoiding unnecessary recalculations and reducing arithmetic with fundamental units.
- Before carrying out step (10) it may prove useful to order the elements of U according to the corresponding height $h_{\boldsymbol{u}}$. Even though this has an additional cost, it could prevent the unnecessary checking of many inequalities $h_{\boldsymbol{u}} \leq w$ in step (10): traversing the list U, one would check this inequality only until it fails once, and then no further elements of U need to be considered.
- In order to reduce both the cost of arithmetic operations in K and the amount of memory used in carrying out steps (11) - (13) of Algorithm 3, compact representations of algebraic integers could be used (see [21]); this would be especially useful for storing and working with fundamental units. All

the operations on algebraic numbers that are needed in Algorithm 3 can be done efficiently working only with compact representations, thus avoiding the need to store and compute with the numbers themselves.

Note that several quantities appearing in Algorithm 3 involve real numbers, so that an implementation of the algorithm may require floating point arithmetic. For some applications this may not be an issue, but if one needs to know with certainty that all elements not exceeding the specified height bound have been found, then it is crucial to choose the precision for floating point calculations carefully. In the next section we will address this problem in detail.

4. Error analysis

There are two issues that must be considered in order to implement Algorithm 3 in such a way that the output is guaranteed to be complete and correct. These issues are due to the fact that in a computer we cannot work exactly with the real numbers that appear in the algorithm (heights of algebraic numbers, logarithms of real numbers, absolute values of algebraic numbers), so we must make do with rational approximations of them. We consider now the question of finding approximations that are good enough to guarantee correct results.

The first issue is that of computing the height of an algebraic number. In carrying out Algorithm 3 one must check inequalities of the form $h_K(x) \leq D$ for given $x \in K^*$ and $D \in \mathbb{R}$. In practice, one can only work with rational approximations $\tilde{h}$ of $h_K(x)$ and $\tilde{D}$ of D, and check whether $\tilde{h} \leq \tilde{D}$. However, it may happen that $h_K(x) \leq D$ even though $\tilde{h} > \tilde{D}$. To deal with this problem one must be able to find arbitrarily close rational approximations of $h_K(x)$.

The second issue is that of enumerating lattice points inside a polytope, which is needed in Algorithm 2. The polytopes considered are of the form $T(\mathcal{B})$, where $\mathcal{B} = [-d, d]^r$ is a box in $\mathbb{R}^r$ and $T : \mathbb{R}^r \to \mathbb{R}^r$ is a linear isomorphism. In practice, the box $\mathcal{B}$ must be replaced by a box $\tilde{\mathcal{B}}$ with rational vertices, and the matrix of T must be approximated by a rational matrix corresponding to a map $\tilde{T}$. We will not necessarily have an equality $\mathbb{Z}^r \cap T(\mathcal{B}) = \mathbb{Z}^r \cap \tilde{T}(\tilde{\mathcal{B}})$, so lattice points may be lost in this approximation process. One must therefore take care to ensure that good enough approximations are found so that at least there is a containment $\mathbb{Z}^r \cap T(\mathcal{B}) \subseteq \mathbb{Z}^r \cap \tilde{T}(\tilde{\mathcal{B}})$.

There are several ways of dealing with these issues, each one leading to a different implementation of the main algorithm. For concreteness, we describe in this section one way of solving these problems, and we give the corresponding modification of Algorithm 3.

We introduce the following terminology to be used throughout this section: if $\vec{x} = (x_1, \ldots, x_m)$ is a vector in the Euclidean space $\mathbb{R}^m$, we say that $\vec{y} = (y_1, \ldots, y_m) \in \mathbb{R}^m$ is a *δ-approximation* of $\vec{x}$ if $|x_i - y_i| < \delta$ for all $1 \leq i \leq m$.

4.1. **The height function.** Given an element $x \in K^*$ and a real number $\lambda > 0$, we wish to compute a rational number $\tilde{h}$ such that $|\tilde{h} - h_K(x)| < \lambda$. Writing $x = \alpha/\beta$ with $\alpha, \beta \in \mathcal{O}_K$ and using (3.3), we see that $h_K(x)$ can be approximated by first finding good approximations of the vectors $\Lambda(\alpha)$ and $\Lambda(\beta)$.

Lemma 4.1. *Fix $\lambda > 0$ and set $\delta = \lambda/(r+2)$. Let α, β be nonzero elements of $\mathcal{O}_K$. Let $\tilde{n}$, $(s_1, \ldots, s_{r+1})$, and $(t_1, \ldots, t_{r+1})$ be δ-approximations of $\log(N(\alpha, \beta))$, $\Lambda(\alpha)$, and $\Lambda(\beta)$, respectively. Then, with*

$$\tilde{h} := -\tilde{n} + \sum_{i=1}^{r+1} \max\{s_i, t_i\}$$

we have $|h_K(\alpha/\beta) - \tilde{h}| < \lambda$.

Proof. Let $\Lambda(\alpha) = (x_1, \ldots, x_{r+1})$ and $\Lambda(\beta) = (y_1, \ldots, y_{r+1})$. Using (3.3) we obtain

$$|h_K(\alpha/\beta) - \tilde{h}| \leq |\tilde{n} - \log(N(\alpha, \beta))| + \sum_{i=1}^{r+1} |\max\{x_i, y_i\} - \max\{s_i, t_i\}| < (r+2)\delta = \lambda.$$

□

For a nonzero element $y \in K$, each entry of the vector $\Lambda(y)$ is of the form $n_v \log |y|_v$ for some place $v \in M_K^\infty$. Corresponding to v there is an embedding $\sigma : K \hookrightarrow \mathbb{C}$ such that $|y|_v = |\sigma(y)|$. Since $\sigma(y)$ is a complex root of the minimal polynomial of y, known methods (see [14], for instance) can be applied to approximate $\sigma(y)$ with any given accuracy. In this way the vector $\Lambda(y)$ can be approximated to any required precision.

Lemma 4.1 provides a way of approximating the height of any element of K by using the map Λ. However, in practice a slightly different method will be needed for computing heights of units. As mentioned in §3.3, in order to avoid costly arithmetic with fundamental units we do not work directly with units u when carrying out Algorithm 3. Hence, we cannot approximate the vector $\Lambda(u)$ by computing $|u|_v$ for every place v. Instead, a unit $u = \varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}$ is encoded by the tuple $(n_1, \ldots, n_r)$, so we need a way of approximating $h_K(u)$ given only this tuple. Since $\Lambda(u) = \sum_{j=1}^r n_j \Lambda(\varepsilon_j)$, it is enough to approximate the vectors $\Lambda(\varepsilon_i)$ sufficiently well; we make this precise in the following lemma.

Lemma 4.2. *Fix $\lambda, M > 0$ and set $\delta = \lambda/(r(r+1)M)$. Let $\{\varepsilon_1, \ldots, \varepsilon_r\}$ be a system of fundamental units for $\mathcal{O}_K^\times$, and for each j let $(s_{1,j}, \ldots, s_{r+1,j})$ be a rational δ-approximation of $\Lambda(\varepsilon_j) = (x_{1,j}, \ldots, x_{r+1,j})$. Suppose $u = \varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}$ is a unit with $|n_1|, \ldots, |n_r| \leq M$. Then, with*

$$\tilde{h} := \sum_{i=1}^{r+1} \max\left\{ \sum_{j=1}^{r} n_j s_{i,j}\, , 0 \right\}$$

we have

$$|h_K(u) - \tilde{h}| < \lambda.$$

Proof. Since $\Lambda(u) = \sum_{j=1}^r n_j \Lambda(\varepsilon_j)$, the i-th coordinate of $\Lambda(u)$ is given by $\sum_{j=1}^r n_j x_{i,j}$. Applying (3.3) to $u = u/1$ yields

$$h_K(u) = \sum_{i=1}^{r+1} \max\left\{ \sum_{j=1}^{r} n_j x_{i,j}\, , 0 \right\}.$$

Therefore,

$$|h_K(u) - \tilde{h}| = \left| \sum_{i=1}^{r+1} \left(\max\left\{ \sum_{j=1}^{r} n_j x_{i,j}, 0 \right\} - \max\left\{ \sum_{j=1}^{r} n_j s_{i,j}, 0 \right\} \right) \right| \leq \sum_{i=1}^{r+1} \sum_{j=1}^{r} |n_j| \cdot |x_{i,j} - s_{i,j}| < r(r+1)M\delta = \lambda.$$

□

4.2. **Units of bounded height.** We use here the notation from §3.2. Let $d = \log D$ and let $\mathcal{B} = [-d, d]^r$. Algorithm 2 requires that we enumerate all integer lattice points in the polytope $S^{-1}(\mathcal{B})$. In practice, the matrix S must be replaced by a rational approximation $\tilde{S}$, and the box $\mathcal{B}$ by a rational box $\tilde{\mathcal{B}}$. We show here how to choose these approximations so that $S^{-1}(\mathcal{B}) \subseteq \tilde{S}^{-1}(\tilde{\mathcal{B}})$. For the purpose of enumerating integer lattice points, we may then replace $\mathcal{B}$ with $\tilde{\mathcal{B}}$ and S with $\tilde{S}$, thus avoiding errors arising from floating-point arithmetic.

For a vector $v \in \mathbb{R}^r$ we denote by $|v|$ the usual Euclidean norm of v, and for a linear map $L : \mathbb{R}^r \to \mathbb{R}^r$ we let $\|L\|$ denote the operator norm,

$$\|L\| = \sup_{|x| \leq 1} |Lx|.$$

We also denote by L the matrix of L with respect to the standard basis for $\mathbb{R}^r$. Recall that the supremum norm of L is given by $\|L\|_{\sup} := \max_{i,j} |L_{i,j}|$, and that there is an inequality

$$\|L\| \leq r\sqrt{r} \cdot \|L\|_{\sup}. \tag{4.1}$$

We begin with two results which will be useful for approximating the inverse of a matrix.

Lemma 4.3. *Let V be an $r \times r$ invertible matrix over the real numbers, and let $\tilde{V}$ be a matrix such that*

$$\|\tilde{V} - V\| \cdot \|V^{-1}\| < 1.$$

Then $\tilde{V}$ is invertible and

$$\|\tilde{V}^{-1} - V^{-1}\| \leq \frac{\|\tilde{V} - V\| \cdot \|V^{-1}\|^2}{1 - \|\tilde{V} - V\| \cdot \|V^{-1}\|}.$$

Proof. See the proof of Theorem 9.8 in [17]. □

Using (4.1) we obtain:

Corollary 4.4. *With V as in the lemma, let m be a constant with $m \geq r^2 \cdot \|V^{-1}\|_{\sup}$. Given $\lambda > 0$, let $\tilde{V}$ be a matrix such that $\|\tilde{V} - V\|_{\sup} < \frac{\lambda}{r^2(m^2+m\lambda)}$. Then $\tilde{V}$ is invertible and $\|\tilde{V}^{-1} - V^{-1}\| < \lambda$.*

We can now give the required accuracy in approximating the matrix S.

Proposition 4.5. *Let S be an invertible $r \times r$ matrix over the real numbers, and let d be a positive real number. Given $\eta > 0$, define $\tilde{\mathcal{B}} = [-d - \eta, d + \eta]^r$. Let m be a real number such that*

$$m \geq r^2 \cdot \max\{\|S\|_{\sup}, \|S^{-1}\|_{\sup}\}.$$

Define constants

$$\lambda := \frac{\eta}{dr(1+m)} \quad \text{and} \quad \delta := \min\left\{\frac{\lambda}{r^2(m^2+m\lambda)}, \frac{1}{r^2}\right\}.$$

If $\tilde{S}$ is any $r \times r$ matrix such that $\|\tilde{S} - S\|_{\sup} < \delta$, then $\tilde{S}$ is invertible and $S^{-1}(\mathcal{B}) \subseteq \tilde{S}^{-1}(\tilde{\mathcal{B}})$.

Proof. It follows from Corollary 4.4 that $\tilde{S}$ is invertible and $\|\tilde{S}^{-1} - S^{-1}\| < \lambda$. For any $x \in \mathcal{B}$ we then have

$$|\tilde{S}(S^{-1}x) - x| = |\tilde{S}(S^{-1}x) - \tilde{S}(\tilde{S}^{-1}x)| \leq \|\tilde{S}\| \cdot \|S^{-1} - \tilde{S}^{-1}\| \cdot |x| < \eta.$$

Hence, we see that $\tilde{S}(S^{-1}x) \in \tilde{\mathcal{B}}$, so $S^{-1}x \in \tilde{S}^{-1}(\tilde{\mathcal{B}})$ and this completes the proof. □

Proposition 4.5 reduces the problem of finding an adequate approximation of S to finding upper bounds for $\|S\|_{\sup}$ and $\|S^{-1}\|_{\sup}$. The former can be easily done using any approximation of S. One way of finding an upper bound for $\|S^{-1}\|_{\sup}$ is to use the fact that $S^{-1} = \frac{1}{\det(S)} \cdot A$, where A is the adjugate matrix of S. By approximating S one can obtain a lower bound for $\det(S)$ and an upper bound for the entries of A.

4.3. **Revised algorithm.** Using the methods described above we give a new version of Algorithm 3 which takes precision issues into account. We assume here that $r > 0$; the case $r = 0$ is treated in §4.4 for imaginary quadratic fields, and is trivial when $K = \mathbb{Q}$.

Algorithm 4 (Algebraic numbers of bounded height)**.**
Input: A number field K, a bound $B \geq 1$, and a tolerance $\theta \in (0, 1]$, with $B, \theta \in \mathbb{Q}$.
Output: Two lists, L and L', such that:

- If $x \in K$ satisfies $H_K(x) \leq B$, then x is in either L or L'.
- For every $x \in L$, $H_K(x) < B$.
- For every $x \in L'$, $|H_K(x) - B| < \theta$.

(1) Set $t = \theta/(3B)$ and let $\delta_1 = t/(6r+12)$. Find a complete set $\{\mathfrak{a}_1, \dots, \mathfrak{a}_h\}$ of ideal class representatives for $\mathcal{O}_K$, and for each index ℓ compute a rational δ_1-approximation of $\log(N(\mathfrak{a}_\ell))$.

(2) Construct the set

$$\mathcal{N} := \bigcup_{\ell=1}^{h} \{m \cdot N(\mathfrak{a}_\ell) : m \le B\}$$

and make a list $\mathcal{P}$ of all nonzero principal ideals of $\mathcal{O}_K$, each represented by a single generator g, having norm in $\mathcal{N}$. For each g, find a δ_1-approximation of $\Lambda(g)$.

(3) For each ideal $\mathfrak{a}_\ell$, make a list $(g_{\ell,1}), \dots, (g_{\ell,s_\ell})$ of all elements of $\mathcal{P}$ contained in $\mathfrak{a}_\ell$ whose norms are at most $B \cdot N(\mathfrak{a}_\ell)$.

(4) For each index ℓ:
  (a) Make a list R_ℓ of pairs (i,j) such that $1 \le i < j \le s_\ell$ and $(g_{\ell,i}, g_{\ell,j}) = \mathfrak{a}_\ell$.
  (b) For each pair (i,j) in R_ℓ:
    Use Lemma 4.1 and data from steps (1) and (2) to find a rational approximation of $h_K(g_{\ell,i}/g_{\ell,j})$. The result will be a rational number $r_{\ell,i,j}$ such that $|r_{\ell,i,j} - h_K(g_{\ell,i}/g_{\ell,j})| < t/6$.

(5) Find a rational number b such that $\frac{t}{12} < b - \log(B) < \frac{t}{4}$ and set $\tilde{d} = b + \frac{t}{6} + \max_\ell \max_{(i,j)\in R_\ell} r_{\ell,i,j}$.

(6) Compute a system of fundamental units $\boldsymbol{\varepsilon} = \{\varepsilon_1, \dots, \varepsilon_r\}$ and find a constant m such that

$$m \ge r^2 \cdot \max\{\|S(\boldsymbol{\varepsilon})\|_{\sup}, \|S(\boldsymbol{\varepsilon})^{-1}\|_{\sup}\}.$$

(7) Define constants

$$\tilde{\lambda} = \frac{t/12}{\tilde{d}r(1+m)},\ \tilde{\delta} = \min\left\{\frac{\tilde{\lambda}}{r^2(m^2+m\tilde{\lambda})}, \frac{1}{r^2}\right\},\ M = \lceil \tilde{d}(m+\tilde{\lambda}\sqrt{r})\rceil,\ \delta_2 = \min\left\{\tilde{\delta}, \frac{t/6}{r(r+1)M}\right\}.$$

(8) Compute δ_2-approximations $v_1, \dots, v_r$ of the vectors $\Lambda(\varepsilon_1), \dots, \Lambda(\varepsilon_r)$, construct the $r \times r$ matrix $\tilde{S}$ whose j-th column is the vector v_j with its last coordinate deleted, and compute $\tilde{S}^{-1}$.

(9) Construct a list U consisting of all integer vectors $(n_1, \dots, n_r)$ in the polytope $\tilde{S}^{-1}([-\tilde{d}, \tilde{d}]^r)$.

(10) Create a list L containing only the element 0, and create empty lists U_0, U_0' and L_0, L_0'.

(11) For each tuple $\boldsymbol{u} = (n_1, \dots, n_r)$ in U:
  (a) Compute the vector $\tilde{\Lambda}_{\boldsymbol{u}} = \sum_{j=1}^{r} n_j v_j$.
  (b) Using $\tilde{\Lambda}_{\boldsymbol{u}}$ and Lemma 4.2, find a rational approximation of $h_K\left(\varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}\right)$. The result will be a rational number $r_{\boldsymbol{u}}$ such that $|r_{\boldsymbol{u}} - h_K\left(\varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}\right)| < t/6$.
  (c) If $r_{\boldsymbol{u}} < b - \frac{5}{12}t$, then append $\boldsymbol{u}$ to U_0.
  (d) If $b - \frac{5}{12}t \le r_{\boldsymbol{u}} < b + \frac{1}{12}t$, then append $\boldsymbol{u}$ to U_0'.
  (e) If $r_{\boldsymbol{u}} > t/12 + \tilde{d}$, then remove $\boldsymbol{u}$ from U.

(12) For each index ℓ :
  For each pair $(i,j) \in R_\ell$:
    Let $w = b + r_{\ell,i,j} + \frac{1}{4}t$.
    For each tuple $\boldsymbol{u} = (n_1, \dots, n_r)$ in U:
      If $r_{\boldsymbol{u}} < w$, then:
        (a) Let P be the packet $(\ell, (i,j), (n_1, \dots, n_r))$.
        (b) Use the data from steps (1), (2), and 11(a), together with (3.3), to find a rational approximation of $h_K(c(P))$. The result will be a rational number r_P with $|r_P - h_K(c(P))| < t/3$.
        (c) If $r_P \le b - \frac{7}{12}t$, then append the packet P to L_0.
        (d) If $b - \frac{7}{12}t < r_P < b + \frac{1}{4}t$, then append the packet P to L_0'.

(13) Make a list consisting of the distinct tuples $(n_1, \dots, n_r)$ appearing in U_0, U_0' or in some packet P in L_0 or L_0'. Compute and store all units of the form $\varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}$ with $(n_1, \dots, n_r)$ in this list.

(14) For each tuple $(n_1, \dots, n_r)$ in U_0, append to L all numbers of the form $\zeta\varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}$ with $\zeta \in \mu_K$, and similarly for U_0' and L'. Use the units computed in step (13).

(15) For each packet P in L_0, append to L all numbers of the form $\zeta \cdot c(P)$ and $\zeta/c(P)$ with $\zeta \in \mu_K$, and similarly for L_0' and L'. Use the data from step (13) when computing the numbers $c(P)$.

(16) Return the lists L and L'.

We make the following comments regarding various steps of Algorithm 4:

- Let $S = S(\boldsymbol{\varepsilon})$. With d as in step (5) of Algorithm 3 we have $\tilde{d} > d + t/12$. Therefore, if we set $\eta = t/12$ and let λ and δ be defined as in Proposition 4.5, then $\tilde{\lambda} < \lambda$ and $\tilde{\delta} \leq \delta$. By construction, $\|\tilde{S} - S\|_{\sup} < \tilde{\delta} \leq \delta$, so that by Proposition 4.5 we have
$$S^{-1}([-d, d]^r) \subseteq \tilde{S}^{-1}([-d - \eta, d + \eta]^r) \subseteq \tilde{S}^{-1}([-\tilde{d}, \tilde{d}]^r).$$
- In order to use Lemma 4.2 in step 11(b) we must know that $|n_i| \leq M$ for all $1 \leq i \leq r$. Since $\boldsymbol{u} \in \tilde{S}^{-1}([-\tilde{d}, \tilde{d}]^r)$, we clearly have the upper bound $|n_i| \leq \tilde{d}\sqrt{r}\|\tilde{S}^{-1}\|$. By Corollary 4.4, $\|\tilde{S}^{-1} - S^{-1}\| < \tilde{\lambda}$, so applying (4.1) we have $\|\tilde{S}^{-1}\| \leq r\sqrt{r}\|S^{-1}\|_{\sup} + \tilde{\lambda}$. It follows that $|n_i| \leq \tilde{d}(m + \tilde{\lambda}\sqrt{r}) \leq M$.
- The condition $r_{\boldsymbol{u}} + \frac{5}{12}t < b$ from step 11(c) implies that $h_{\boldsymbol{u}} < \log B$; the condition $b - \frac{5}{12}t \leq r_{\boldsymbol{u}} < b + \frac{1}{12}t$ from step 11(d) implies $|h_{\boldsymbol{u}} - \log B| < t$. Moreover, every $\boldsymbol{u} = (n_1, \dots, n_r)$ for which $h_K(\varepsilon_1^{n_1} \cdots \varepsilon_r^{n_r}) \leq \log B$ is in U_0 or U_0', since $h_{\boldsymbol{u}} \leq \log B$ implies $r_{\boldsymbol{u}} < b + \frac{1}{12}t$.
- The condition $r_{\boldsymbol{u}} - t/12 > \tilde{d}$ from step 11(e) implies that $h_{\boldsymbol{u}} > d$.
- The condition $r_P + \frac{7}{12}t \leq b$ from step 12(c) implies that $h_K(c(P)) < \log B$; the condition $b - \frac{7}{12}t < r_P < b + \frac{1}{4}t$ from step 12(d) implies that $|h_K(c(P)) - \log B| < t$. Moreover, every packet P with $h(c(P)) \leq \log B$ is in either L_0 or L_0', since $h_K(c(P)) \leq \log B$ implies that $r_P < b + \frac{t}{4}$.
- Elements $x \in L$ satisfy $H_K(x) < B$, since they come from tuples in U_0 and packets in L_0. Elements $x \in L'$ satisfy $|h_K(x) - \log B| < t$, which implies that $|H_K(x) - B| < \theta$ by the Mean Value Theorem.

Note that the list L' of Algorithm 4 consists of elements $x \in K$ whose heights are so close to B that it is not possible to decide whether $H_K(x) \leq B$ with the tolerance specified as input. In particular, L' might contain elements of height exactly B. For general number fields K we cannot prevent this from occurring; however, for quadratic fields we can prevent it, as explained below.

4.4. **Case of quadratic fields.** We give in this section a way to shorten the list L' from Algorithm 4 in the case of real quadratic fields, and to eliminate it altogether in the case of imaginary quadratic fields.

Proposition 4.6. *Let K be a quadratic field and let $x \in K^*$. Let σ be the generator of* $\mathrm{Gal}(K/\mathbb{Q})$.

(1) If K is an imaginary field, then $H_K(x)$ is an integer.

(2) If K is a real field, then $H_K(x) \in \mathbb{Q}$ if and only if $\max\{|x|, |\sigma(x)|\} \leq 1$ or $\min\{|x|, |\sigma(x)|\} \geq 1$. Moreover, if $H_K(x) \in \mathbb{Q}$, then $H_K(x) \in \mathbb{Z}$.

Proof. Write $x = a/b$ with $a, b \in \mathcal{O}_K$, and let $\mathfrak{a} = (a, b)$ be the ideal generated by a and b in $\mathcal{O}_K$. Then $H_K(x) = N(\mathfrak{a})^{-1} \prod_{v \in M_K^\infty} \max\{|a|_v^{n_v}, |b|_v^{n_v}\}$. There are coprime ideals I and J of $\mathcal{O}_K$ such that $(a) = \mathfrak{a} \cdot I$ and $(b) = \mathfrak{a} \cdot J$. We then have $H_K(x) = N(J) \prod_{v \in M_K^\infty} \max\{|x|_v^{n_v}, 1\}$.

(1) If K is an imaginary field, then $H_K(x) = N(\mathfrak{a})^{-1} \max\{N_{K/\mathbb{Q}}(a), N_{K/\mathbb{Q}}(b)\} = \max\{N(I), N(J)\} \in \mathbb{Z}$.

(2) If K is a real field, then $H_K(x) = N(J) \max\{|x|, 1\} \cdot \max\{|\sigma(x)|, 1\}$. If $\max\{|x|, |\sigma(x)|\} \leq 1$, then $H_K(x) = N(J) \in \mathbb{Z}$. If $\min\{|x|, |\sigma(x)|\} \geq 1$, then
$$H_K(x) = N(J)|N_{K/\mathbb{Q}}(x)| = N(J)N(a)/N(b) = N(I) \in \mathbb{Z}.$$

Now suppose that $\max\{|x|, |\sigma(x)|\} > 1$ and $\min\{|x|, |\sigma(x)|\} < 1$. Then, without loss of generality we may assume that $|x| < 1 < |\sigma(x)|$. It follows that $x \notin \mathbb{Q}$, so $H_K(x) = N(J)|\sigma(x)| \notin \mathbb{Q}$.

□

In the case of real quadratic fields it is possible to detect some elements of the list L' from Algorithm 4 which should be in the list L: if $x \in L'$ has height $H_K(x) \in \mathbb{Q}$ — a condition which can be determined using Proposition 4.6 — then by construction of L' it must be the case that $H_K(x)$ is the unique integer closest to B (assuming the tolerance θ from Algorithm 4 was chosen to be less than $1/2$). If the nearest integer is $\lfloor B \rfloor$, then x can then be deleted from L' and appended to L. Otherwise, x can be deleted from L'. At the end of this process the list L' will only contain elements of K whose heights are irrational and very close to B.

For imaginary quadratic fields K there is a modification of Algorithm 3 that allows us to determine elements of bounded height without doing any height computations. Thus, for such fields we avoid the need for a list L' as in Algorithm 4.

With the notation and terminology of §3.1 we have:

Theorem 4.7. *Let K be an imaginary quadratic field. Then*

$$\{\gamma \in K^* : H_K(\gamma) \leq B\} = \mu_K \cup \bigcup_{B\text{-packets } P} F(P).$$

Proof. One containment follows from Theorem 3.1, since $\mathcal{O}_K^\times = \mu_K$ in this case. It is therefore enough to show that $H_K(c(P)) \leq B$ for every packet P. Letting $P = (\ell, (i, j))$ we have

$$N(\mathfrak{a}_\ell) H_K(c(P)) = \prod_{v \in M_K^\infty} \max\{|g_{\ell,i}|_v^{n_v}, |g_{\ell,j}|_v^{n_v}\} = \max\{N_{K/\mathbb{Q}}(g_{\ell,i}), N_{K/\mathbb{Q}}(g_{\ell,j})\} \leq B \cdot N(\mathfrak{a}_\ell).$$

Hence, $H_K(c(P)) \leq B$. □

Theorem 4.7 leads to the following algorithm.

Algorithm 5 (Numbers of bounded height in an imaginary quadratic field).
Input: An imaginary quadratic field K and a bound $B \geq 1$.
Output: A list of all elements $x \in K$ satisfying $H_K(x) \leq B$.

(1) Create a list L containing 0 and all elements of μ_K.

(2) Find a complete set $\{\mathfrak{a}_1, \ldots, \mathfrak{a}_h\}$ of ideal class representatives for $\mathcal{O}_K$.

(3) Construct the set

$$\mathcal{N} := \bigcup_{\ell=1}^{h} \{m \cdot N(\mathfrak{a}_\ell) : m \leq B\}$$

and make a list $\mathcal{P}$ of all nonzero principal ideals of $\mathcal{O}_K$, each represented by a single generator g, having norm in $\mathcal{N}$.

(4) For each ideal $\mathfrak{a}_\ell$, make a list $(g_{\ell,1}), \ldots, (g_{\ell,s_\ell})$ of all elements of $\mathcal{P}$ contained in $\mathfrak{a}_\ell$ whose norms are at most $B \cdot N(\mathfrak{a}_\ell)$.

(5) For each index ℓ:
For each pair of indices (i, j) such that $1 \leq i < j \leq s_\ell$ and $(g_{\ell,i}, g_{\ell,j}) = \mathfrak{a}_\ell$:
Let $c = g_{\ell,i}/g_{\ell,j}$ and append to L all elements of the form $\zeta \cdot c$ and ζ/c with $\zeta \in \mu_K$.

(6) Return the list L.

Note that, by Theorem 3.1, the list L will not contain duplicate elements.

5. Efficiency of the algorithm

We discuss in this section a measure of the efficiency of Algorithm 3 — henceforth abbreviated A3 — and of the algorithm of Pethő and Schmitt — abbreviated PS — proposed in [15]. Given a number field K and a height bound B, both methods begin by computing some basic data attached to K: in the case of PS we need an integral basis for $\mathcal{O}_K$, and in the case of A3 we need also the ideal class group and a set of fundamental units. After this step, both methods construct a set of elements of K which is known to contain the desired set of numbers of bounded height; this larger set will be called the *search space* of the method and denoted by $\mathcal{S}_{\mathrm{PS}}(B)$ or $\mathcal{S}_{\mathrm{A3}}(B)$. Once a search space is known, the two methods proceed to compute the height of each element in this set and check whether it is smaller than B. We will measure the efficiency of a method by comparing the size of the search space to the size of the set of elements of height $\leq B$. Thus, we define the *search ratio* of PS to be the number

$$\sigma_{\mathrm{PS}}(B) := \frac{\#\mathcal{S}_{\mathrm{PS}}(B)}{\#\{x \in K : H_K(x) \leq B\}} ,$$

and similarly for A3.

We begin by stating the main result on which PS is based. To do this we will need the *Minkowski embedding* $\Phi : K \hookrightarrow \mathbb{R}^n$ defined by

$$\Phi(x) = \Big(\sigma_1(x), \dots, \sigma_{r_1}(x), \sqrt{2}\Re\tau_1(x), \sqrt{2}\Im\tau_1(x), \dots, \sqrt{2}\Re\tau_{r_2}(x), \sqrt{2}\Im\tau_{r_2}(x)\Big) . \tag{5.1}$$

Recall that $\Phi(\mathcal{O}_K) \subset \mathbb{R}^n$ is a lattice of rank n and determinant $|\Delta_K|^{1/2}$. In PS one requires a reduced basis for this lattice, which can be obtained by applying the LLL algorithm to any given basis.

Theorem 5.1 (Pethő, Schmitt)**.** *Let K be a number field of degree n over $\mathbb{Q}$, and let $B \geq 1$ be a real number. Let $\{\omega_1, \dots, \omega_n\}$ be an LLL-reduced integral basis for $\mathcal{O}_K$. Then every element $x \in K$ with $H_K(x) \leq B$ can be written in the form*

$$x = \frac{a_1\omega_1 + \cdots + a_n\omega_n}{c},$$

where $a_1, \dots, a_n$ and c are integers satisfying

$$1 \leq c \leq B \quad \text{and} \quad |a_i| \leq 2^{n(n-1)/4} B c .$$

Proof. By [15, Thm. 1] it suffices to show that for every index i we have

$$\sqrt{n} \cdot \prod_{j \neq i} \|\Phi(\omega_j)\| \leq 2^{n(n-1)/4} |\Delta_K|^{1/2}.$$

This inequality is a consequence of the fact that the given basis is reduced (see (1.8) in [11]) together with the fact — noted in the proof of [15, Thm. 2] — that $\|\Phi(\omega_j)\| \geq \sqrt{n}$ for every j. □

Remark. The bound for $|a_i|$ given in Theorem 5.1 is slightly better than the one proved in [15, Thm. 2]; the reason for this is that the Minkowski embedding used in [15] does not include the factors of $\sqrt{2}$ for the complex embeddings of K.

Theorem 5.1 leads to the following algorithm:

Algorithm 6 (PS)**.**
Input: A number field K and a bound $B \geq 1$.
Output: A list of all elements $x \in K$ satisfying $H_K(x) \leq B$.

(1) Compute an LLL-reduced integral basis $\omega_1, \dots, \omega_n$ for $\mathcal{O}_K$.
(2) Create an empty list L.
(3) For $c = 1$ to $\lfloor B \rfloor$:

(a) Let $D = \lfloor 2^{n(n-1)/4} Bc \rfloor$.
(b) For every integer tuple $(a_1, \dots, a_n) \in [-D, D]^n$:
  (i) Let $x = \dfrac{a_1\omega_1 + \cdots + a_n\omega_n}{c}$.
  (ii) If $H_K(x) \leq B$, then append x to L.
(4) Return the list L.

We now give our main result comparing the efficiency of PS with that of A3.

Theorem 5.2. *Let K be a number field of degree n. The search ratios of* PS *and* A3 *satisfy*

$$\sigma_{\mathrm{PS}}(B) \gg B^{2n-2} \quad \text{and} \quad \sigma_{\mathrm{A3}}(B) \ll (\log B)^r,$$

where r is the rank of the unit group of $\mathcal{O}_K$.

Proof. By Schanuel's formula [18] we know that there is a constant C_K such that

$$\#\{P \in K : H_K(P) \leq B\} \sim C_K B^2.$$

A simple calculation shows that the size of the search space in PS satisfies $\#\mathcal{S}_{\mathrm{PS}}(B) > B^{2n} 2^{n^2(n-1)/4+n}$; the first statement in the theorem then follows easily. Now let $\mathfrak{a} = \{\mathfrak{a}_1, \dots, \mathfrak{a}_h\}$ be a complete set of ideal class representatives for $\mathcal{O}_K$ and $\boldsymbol{\varepsilon} = \{\varepsilon_1, \dots, \varepsilon_r\}$ a system of fundamental units. For each index ℓ let $g_{\ell,1}, \dots, g_{\ell,s_\ell}$ be generators for all nonzero principal ideals contained in $\mathfrak{a}_\ell$ whose norms are at most $B \cdot N(\mathfrak{a}_\ell)$. By [2, §2.5.4] we may assume that

$$|g_{\ell,i}|_v^{n_v} \leq E_K(B \cdot N(\mathfrak{a}_\ell))^{1/(r+1)} \tag{5.2}$$

for every place $v \in M_K^\infty$ and all indices ℓ, i. Here E_K is a constant which depends on $\boldsymbol{\varepsilon}$ but not on B.

Let $P(B) = \sum_{\ell=1}^h s_\ell$. Using the bound given in [13, Thm. 1] we find that $P(B) \ll B$. Using Theorem 3.1 we see that the size of the search space considered in A3 satisfies

$$\#\mathcal{S}_{\mathrm{A3}}(B) \leq 1 + \#\{u \in \mathcal{O}_K^\times : H_K(u) \leq D\} \cdot \left(1 + 2 \cdot P(B)^2\right) \ll B^2 \cdot \#\{u \in \mathcal{O}_K^\times : H_K(u) \leq D\}, \tag{5.3}$$

where D is any number such that $D \geq B \cdot \max_{\ell,i,j} H_K(g_{\ell,i}/g_{\ell,j})$. By (5.2) we have

$$H_K(g_{\ell,i}/g_{\ell,j}) \leq \frac{\prod_{v \in M_K^\infty} \max\{|g_{\ell,i}|_v^{n_v}, |g_{\ell,j}|_v^{n_v}\}}{N(\mathfrak{a}_\ell)} \leq \frac{\prod_{v \in M_K^\infty} E_K(B \cdot N(\mathfrak{a}_\ell))^{1/(r+1)}}{N(\mathfrak{a}_\ell)} = F_K B$$

for all ℓ, i, j and for some constant F_K independent of B. Hence, we may take $D = F_K B^2$. By Corollary 3.5,

$$\#\{u \in \mathcal{O}_K^\times : H_K(u) \leq D\} \ll (\log B)^r.$$

Therefore, by (5.3), the size of the search space in A3 satisfies $\#\mathcal{S}_{\mathrm{A3}}(B) \ll B^2 (\log B)^r$. The second statement in the theorem follows from this inequality and Schanuel's asymptotic estimate. □

6. Computational complexity and performance

Theorem 5.2 shows that, for a fixed field K, the method A3 is asymptotically (as $B \to \infty$) much more efficient than PS. However, the search ratio is not the only factor determining total computation time: for instance, in A3 the initial step of computing the ideal class group and unit group of $\mathcal{O}_K$ can be very time-consuming if the discriminant of K is large. We will briefly discuss here the time complexity of various steps in PS and A3. In order to illustrate the theoretical statements made here, we will also give examples based on explicit computations done with the two methods. Both algorithms have been implemented in Sage [20], and all computations below have been done on a Mac Pro with a Quad-Core 2.26 GHz processor and 8 GB

of memory. A precision must be chosen for floating point calculations, so we will fix a precision of 100 bits in all examples.

The first step in PS is to determine an integral basis for $\mathcal{O}_K$. It is known by work of Chistov [5, Thms. 2 and 3] that this is polynomial-time equivalent to determining the squarefree part of the discriminant of a defining polynomial P for K; in other words, the first problem can be solved in polynomial time if and only if the second one can. Unfortunately, finding the squarefree part of an integer is not much easier than factoring it, and current methods for finding integral bases (see, for instance, [6, §6.1]) do indeed begin by factoring $\mathrm{disc}(P)$. Assuming the general number field sieve [4] is used for this factorization, the expected running time will be

$$O\left(\exp((64d/9)^{1/3}(\log d)^{2/3})\right),$$

where $d = 1 + \lfloor \log_2 \mathrm{disc}(P) \rfloor$ is the bit length of $\mathrm{disc}(P)$. Once an integral basis $\boldsymbol{\omega} = \{\omega_1, \ldots, \omega_n\}$ is known, the LLL algorithm can be applied to the image of $\boldsymbol{\omega}$ under the Minkowski embedding (5.1) to obtain a reduced basis for $\mathcal{O}_K$. This step will run in time

$$O\left(n^6(\log \max_{1\le i\le n} \|\Phi(\omega_i)\|)^3\right).$$

After computing a reduced basis for $\mathcal{O}_K$, the remaining time in PS is spent computing the heights of all elements in a search space, and it is here that PS has its main drawback. As mentioned in the proof of Theorem 5.2, the size of the search space is greater than $B^{2n}2^{n^2(n-1)/4+n}$ — much larger than the size of the set PS is attempting to compute, which is roughly B^2. To illustrate the effect this large search space has on computation time, we give in Table 1 below the time required by PS to compute elements of bounded height in three number fields. We have chosen here the quadratic, cubic, and quartic fields of smallest discriminant (such data may be found in Jones's number field database [10]), so that the time required to compute an integral basis for K is negligible. In the table, a field is specified by giving a defining polynomial for it. Note that, even with the very small bounds B chosen here, the computing times with PS are far from optimal: for comparison, the same computations using A3 took 0.03 seconds for the quartic field, 0.02 seconds for the cubic field, and 0.03 seconds for the quadratic field. This great difference in computing times for A3 and PS is due to the extremely large search space used in PS.

Number field K	Height bound B	PS time	$\sigma_{\mathrm{PS}}(B)$
$x^4 - x^3 + x^2 - x + 1$	2	36.15 hours	1.73×10^6
$x^3 - x^2 - 2x + 1$	4	3.15 hours	77,175
$x^2 - x + 1$	20	5.85 hours	12,630

TABLE 1. Sample computations with PS

Given these times, we do not consider PS to be a practical method to use when K has degree larger than 2, and even for quadratic fields the height bound B must be very small in order for PS to terminate in a reasonable time.

We discuss now the complexity of A3. In this algorithm, the initial step can have a substantial cost: in addition to a reduced integral basis as required by PS, in A3 we need to compute ideal class representatives and fundamental units for $\mathcal{O}_K$; by [12, Thm. 5.5], this step can be done in time

$$(2 + \log|\Delta_K|)^{O(n)}|\Delta_K|^{3/4}.$$

This appears to be the best known upper bound on the running time for a deterministic computation of class groups and unit groups. However, assuming the generalized Riemann hypothesis (GRH), there are

subexponential probabilistic algorithms: for instance, Buchmann [3] gives an algorithm which has expected running time

$$\exp\left((1.7+o(1))\sqrt{\log|\Delta_K|\log\log|\Delta_K|}\right).$$

If K is a field for which the cost of this initial step is high, then PS may perform better than A3, especially if the height bound B is small. As an example of this phenomenon, consider the field $K=\mathbb{Q}(\sqrt{2928239983})$, which has class number $1,472$. Using PS to find all elements of height at most 4, the time is 16.19 seconds, while using A3 and assuming GRH the time is 34.34 seconds. However, for the height bound $B=20$, the time with PS increases to 9 hours and 40 minutes, while the time for A3 (still assuming GRH) only increases to 36.37 seconds.

In step (2) of A3 one has to determine elements of given norm in $\mathcal{O}_K$. A good method for doing this is provided in [9]; however, the authors do not give a precise statement of the complexity of their method. In any case, the cost of using this algorithm is tied to the size of the fundamental units computed in step (1). If the units are very large, then this step can be costly and even dominate the computation time of the entire algorithm. Consider, for instance the field $K=\mathbb{Q}(\sqrt{123456789123})$. A fundamental unit for K has the form $a+b\sqrt{123456789123}$, where a and b are both greater than 10^{2096}. We apply A3 to this field with height bound $B=3,000$, and we assume GRH in order to quickly obtain the class group and unit group of $\mathcal{O}_K$. The total computation time is 3.7 minutes, with 88% of the time spent on step (2).

Steps (4), (9), and (10) of A3 are mostly spent on height computations. The cost of a single height computation depends on the precision used for floating point arithmetic, and the number of required height computations is determined by the size of the search space of A3. Since we have already studied the size of this space in the previous section, we will not discuss here the cost of these three steps. Note, however, that it will certainly be much smaller than the time spent on height computations in PS, since the search space is substantially smaller.

In step (7) of A3 one must find all integer lattice points that lie inside a polytope in $\mathbb{R}^r$ given by a collection of 2^r rational vertices. The polytope is obtained by applying the linear map S^{-1} to a box of the form $[-d,d]^r$. Here, S is the $r\times r$ matrix whose columns are the vectors $\tilde{\Lambda}(\varepsilon_j)$. The integer points inside this polytope can be determined once a generating function for the polytope has been computed; by [1, Thm. 4.4], this can be done in time $\mathcal{L}^{O(r)}$, where $\mathcal{L}$ is the sum of the bit lengths of the coordinates of all vertices of the polytope. If the embeddings $K\hookrightarrow\mathbb{C}$ are computed with precision p, then the vertices of the polytope consist of floating point numbers of precision p, and thus we have $\mathcal{L}=2^r\cdot rp$. Hence, the generating function can be computed in time

$$(2^r\cdot rp)^{O(r)}.$$

As suggested by the above complexity, an increase in the rank r can lead to a significant increase in the time required for this step. For example, consider the following two sextic fields:

$$K_1: x^6+2 \quad\text{and}\quad K_2: x^6-6x^4+9x^2-3.$$

The fields K_1 and K_2 have the same degree, discriminants of similar size (namely, -1492992 and 1259712), equal class numbers (namely 1), and both have small fundamental units; however, the unit group rank is 2 for K_1 and 5 for K_2. This means that in step (7) of the algorithm, in the case of K_1 we must compute integer points inside a polytope in $\mathbb{R}^2$, while in the case of K_2 we must work in $\mathbb{R}^5$. Applying A3 over both fields with height bound $B=500$, we find that for K_1 the time spent on step (7) is 0.005 seconds, while for K_2 it is 21.6 minutes.

In steps (11) - (13) of A3, the previously computed data of integer lattice points and packets is used to construct the final output list of numbers. This process consists entirely of arithmetic operations with elements of K, and its computational cost is largely determined by the size of the fundamental units computed in step

(1). These units can be very large: for instance, one can reasonably expect that there are infinitely many real quadratic fields K for which a fundamental unit cannot be written down with fewer than $|\Delta_K|^{1/2}$ bits (see [12, §5]). If the chosen fundamental units are large, then the cost of these final steps can be considerable; however, we remark that in practice the effect of large fundamental units on computation time is much greater in step (2) than in these final steps. For example, in the computation mentioned above for the field $K = \mathbb{Q}(\sqrt{123456789123})$, only 4% of the time was spent on steps (11) - (13).

6.1. **Sample computations with A3.** We end by giving a series of examples showing that A3 and Algorithm 5 (abbreviated A5) can be applied with number fields of various degrees and with several different height bounds which would be far beyond the practical range of applicability of PS. Note that, by Theorem 4.7, the search ratio of A5 is always 1.

B	A5 time	$\sigma_{\mathrm{A5}}(B)$	Elements found
200	0.85 seconds	1	15,275
1,000	18.46 seconds	1	393,775
5,000	7.61 minutes	1	9,761,079

TABLE 2. Computations with A5 over the field $K = \mathbb{Q}(\sqrt{-107})$

B	A3 time	$\sigma_{\mathrm{A3}}(B)$	Elements found
200	5.69 seconds	14.49	2,143
1,000	34.71 seconds	17.07	54,679
5,000	7.40 minutes	20.05	1,365,315

TABLE 3. Computations with A3 over the field $K = \mathbb{Q}(\sqrt{36865})$

B	A3 time	$\sigma_{\mathrm{A3}}(B)$	Elements found
100	4.76 seconds	88.66	5,123
500	1.68 minutes	149.78	124,911
1,000	7.42 minutes	175.62	489,255

TABLE 4. Computations with A3 over the field $K : x^6 + 2$

B	A3 time	$\sigma_{\mathrm{A3}}(B)$	Elements found
100	3.27 minutes	28,807	2,679
500	1.33 hours	23,635	81,251
1,000	11.14 hours	52,553	316,915

TABLE 5. Computations with A3 over the cyclotomic field $\mathbb{Q}(\zeta_{13})$

REFERENCES

[1] Alexander Barvinok and James E. Pommersheim. "An algorithmic theory of lattice points in polyhedra". *New perspectives in algebraic combinatorics (Berkeley, CA, 1996–97)*. Vol. 38. Math. Sci. Res. Inst. Publ. Cambridge Univ. Press, 1999, pp. 91–147.

[2] A. I. Borevich and I. R. Shafarevich. *Number theory*. Pure and Applied Mathematics, Vol. 20. Academic Press, 1966.

[3] Johannes Buchmann. "A subexponential algorithm for the determination of class groups and regulators of algebraic number fields". *Séminaire de Théorie des Nombres, Paris 1988–1989*. Vol. 91. Progr. Math. Birkhäuser Boston, 1990, pp. 27–41.

[4] J. P. Buhler, H. W. Lenstra Jr., and Carl Pomerance. "Factoring integers with the number field sieve". *The development of the number field sieve*. Vol. 1554. Lecture Notes in Math. Springer, 1993, pp. 50–94.

[5] A. L. Chistov. "The complexity of the construction of the ring of integers of a global field". *Dokl. Akad. Nauk SSSR* 306.5 (1989), pp. 1063–1067.

[6] Henri Cohen. *A course in computational algebraic number theory*. Vol. 138. Graduate Texts in Mathematics. Springer-Verlag, 1993.

[7] Jesús A. De Loera et al. "Effective lattice point counting in rational convex polytopes". *J. Symbolic Comput.* 38.4 (2004), pp. 1273–1302.

[8] John R. Doyle, Xander Faber, and David Krumm. *Preperiodic points for quadratic polynomials over quadratic fields*. In preparation.

[9] U. Fincke and M. Pohst. "A procedure for determining algebraic integers of given norm". *Computer algebra (London, 1983)*. Vol. 162. Lecture Notes in Comput. Sci. Springer, 1983, pp. 194–202.

[10] John Jones. *Number Fields*. URL: `http://hobbes.la.asu.edu/NFDB/`.

[11] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász. "Factoring polynomials with rational coefficients". *Math. Ann.* 261.4 (1982), pp. 515–534.

[12] H. W. Lenstra Jr. "Algorithms in algebraic number theory". *Bull. Amer. Math. Soc. (N.S.)* 26.2 (1992), pp. 211–244.

[13] M. Ram Murty and Jeanine Van Order. "Counting integral ideals in a number field". *Expo. Math.* 25.1 (2007), pp. 53–66.

[14] Victor Y. Pan. "Univariate polynomials: nearly optimal algorithms for numerical factorization and root-finding". *J. Symbolic Comput.* 33.5 (2002). Computer algebra (London, ON, 2001), pp. 701–733.

[15] A. Pethő and S. Schmitt. "Elements with bounded height in number fields". *Period. Math. Hungar.* 43.1-2 (2001), pp. 31–41.

[16] Bjorn Poonen. "The classification of rational preperiodic points of quadratic polynomials over **Q**: a refined conjecture". *Math. Z.* 228.1 (1998), pp. 11–29.

[17] Walter Rudin. *Principles of mathematical analysis*. Third. International Series in Pure and Applied Mathematics. McGraw-Hill Book Co., 1976.

[18] Stephen Hoel Schanuel. "Heights in number fields". *Bull. Soc. Math. France* 107.4 (1979), pp. 433–449.

[19] Joseph H. Silverman. *The arithmetic of dynamical systems*. Vol. 241. Graduate Texts in Mathematics. Springer, 2007.

[20] W. A. Stein et al. *Sage Mathematics Software (Version 5.9)*. `http://www.sagemath.org`. The Sage Development Team. 2013.

[21] Christoph Thiel. "Short proofs using compact representations of algebraic integers". *J. Complexity* 11.3 (1995), pp. 310–329.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF GEORGIA, ATHENS, GA 30602

E-mail address: `jdoyle@math.uga.edu, dkrumm@math.uga.edu`