MATHEMATICS OF COMPUTATION
Volume 67, Number 223, July 1998, Pages 1285–1308
S 0025-5718(98)00965-X

COMPUTATIONS OF CLASS NUMBERS OF REAL QUADRATIC FIELDS

ANITHA SRINIVASAN

ABSTRACT. In this paper an unconditional probabilistic algorithm to compute the class number of a real quadratic field $\mathbb{Q}(\sqrt{d})$ is presented, which computes the class number in expected time $O(d^{1/5+\epsilon})$. The algorithm is a random version of Shanks' algorithm.

One of the main steps in algorithms to compute the class number is the approximation of $L(1,\chi)$. Previous algorithms with the above running time $O(d^{1/5+\epsilon})$, obtain an approximation for $L(1,\chi)$ by assuming an appropriate extension of the Riemann Hypothesis. Our algorithm finds an appoximation for $L(1,\chi)$ without assuming the Riemann Hypothesis, by using a new technique that we call the 'Random Summation Technique'. As a result, we are able to compute the regulator deterministically in expected time $O(d^{1/5+\epsilon})$. However, our estimate of $O(d^{1/5+\epsilon})$ on the running time of our algorithm to compute the class number is not effective.

1. INTRODUCTION

The result proved in this paper is the following:

Theorem. *Fix $\epsilon > 0$ and let $d > 0$ be a fundamental discriminant. Then the class number h of the real quadratic field $\mathbb{Q}(\sqrt{d})$ can be found, via a probabilistic algorithm, in expected time $O(d^{1/5+\epsilon})$.*

In 1970, Daniel Shanks ([14]) gave a deterministic algorithm for computing the class number of an imaginary quadratic field of negative discriminant d. His algorithm used a simple yet powerful technique that he called *baby-steps-giant-steps.* Under the assumption of an appropriate extension of the Riemann Hypothesis (ERH), Shanks' algorithm can be shown to have running time $O(|d|^{1/5+\epsilon})$.

Later H. W. Lenstra, Jr. [8], Schoof [12] and R. A. Mollin and H. Williams [10], just to name a few, modified Shanks' algorithm to run in real quadratic fields. They gave algorithms with probabilistic running time $O(d^{1/4+\epsilon})$ without asssuming the ERH, and with deterministic running time $O(d^{1/5+\epsilon})$ assuming the ERH.

In this paper a probabilistic algorithm is presented which is a version of Shanks' algorithm that does not assume the ERH and has an expected running time of $O(d^{1/5+\epsilon})$.

There are two different routines in Shanks' original algorithm where one needs to assume the ERH in the analysis of the running time. We first give a simplified

Received by the editor July 2, 1996 and, in revised form, January 31, 1997.
1991 *Mathematics Subject Classification.* Primary 11A51.
Key words and phrases. Class number, binary quadratic forms, real quadratic field, regulator.

overview of Shanks' algorithm: The first step in Shanks' algorithm is to get a good approximation to

$$L(1,\chi) = \prod_{p \text{ prime}} \left(1 - \left(\frac{d}{p}\right)\frac{1}{p}\right)^{-1},$$

where $(\frac{d}{p})$ is the Legendre Symbol. This is done by simply taking the product over the primes $p = O(d^{1/5+\epsilon})$; that this is a 'good enough' approximation is assured by assuming ERH. A good approximation for $L(1,\chi)$ ensures a good approximation for hR, where R is the regulator, because of Dirichlet's formula which is the following:

$$h = \begin{cases} \frac{\sqrt{|d|}}{\pi} L(1,\chi) & \text{for } d < -4, \\ \frac{\sqrt{d}}{R} L(1,\chi) & \text{for } d > 0. \end{cases} \tag{1.1}$$

The next step is to find a good approximation for R, from which we deduce an approximation for h; in fact h is shown to lie in an explicitly computed interval $(L, L + \tilde{l})$. The final step then is to find a subgroup H of the class group of order $\geq \tilde{l}$, in which case h equals $|H| \left(\left[\frac{L}{|H|}\right] + 1\right)$. This is because the order of H divides at most one integer in $(L, L + \tilde{l})$ (since the length of the interval is less than the order of H), which must exist and must be h (since the order of H must divide h, which lies in this interval). We determine such a subgroup H by finding generators for H. We find such generators one at a time, looking for forms that lie outside the subgroup generated by the forms already found. ERH guarantees that there is a set of generators of the form (a, b, c) for the whole class group, with all the values of $a = O(\log^2 d)$, and thus can be found rapidly.

In the modified algorithm presented here, the assumption of ERH is removed using the following "random" techniques:

The first new idea is that of a *Random Summation* (see section 2), a method that can be used to give a good and rapid approximation to certain sums involving many summands. 'Random summation' is used to approximate a sum that can be used to evaluate $L(1,\chi)$:

$$L(1,\chi) = \sum_{n \geq 1} \left(\frac{d}{n}\right)\frac{1}{n}.$$

We neglect the tail end of this sum for $n > d^2$, as it is smaller than the admissible error and then add up "randomly selected" terms in the remaining sum up to d^2. Hence we obtain an interval which contains hR with very high probability.

Note that the random summation technique provides a correct interval only with high probability and so it is possible that the interval obtained does not contain hR. However this is detected by the algorithm for computing the regulator, which is deterministic. Hence either the regulator is computed correctly or the algorithm terminates without an answer, in which case we conclude that the interval provided by random summation is incorrect. In this case we simply repeat the random summation. In section 2 we prove that the probability of obtaining an incorrect interval via random summation is less than $\frac{1}{d^\epsilon}$. Thus after at most d^ϵ tries of random summation, we obtain a correct interval.

In section 4 we show that given an interval containing hR, R is computed deterministically in time $O(d^{1/5+\epsilon})$.

The second new idea involves that part of the algorithm where one needs to find a set of forms that generate the whole class group. Although it has been previously proposed that one could select forms 'randomly' from the class group to do this, we do not know of a reference where a suitable 'random procedure' has been described and appropriately analyzed. We do this here.

Although we are able to determine a suitable unconditional upper bound on the running time of this part of the algorithm, we do so by invoking Siegel's theorem 6.8, which involves a constant which cannot be explicitly determined. Thus we are unable to explicitly determine the actual constant that the 'O' abbreviates in the stated running time of $O(d^{1/5+\epsilon})$, although one would, in practice, know the algorithm had ended and the correct answer given.

We also give a description of an algorithm for computing the regulator of a real quadratic field. This is in most respects the same as previous algorithms, like [8], [10] and [12], other than the major changes as described above. We have tried to give an exposition that would benefit both the calculator and the running time analyser.

An overview of the contents presented is as follows.

In section 2 the details of the random summation technique and an approximation for $L(1,\chi)$ and hence for hR are presented. In section 3 the algorithms necessary for the computation of the regulator are presented. In section 4 we prove that the regulator can be computed deterministically in expected time $O(d^{1/5+\epsilon})$. In section 5, we prove that given a set of generators, the running time for the computation of the order of the subgroup generated is $O(d^{1/5+\epsilon})$, which is the second part of Shanks' algorithm. Section 6 deals with the problem of selecting a random form. An algorithm together with the probability analysis is presented. Also the proof of the main theorem is given here. In the last section 7, we discuss the practical aspects of the algorithms and in particular that of the random summation technique.

2. The random summation technique

The key new idea used is the 'Random Summation Technique'. We use this to approximate the sum $S = \sum_{\substack{n\le d^2 \\ \text{odd } n}} \frac{\left(\frac{d}{n}\right)}{n}$ where $\left(\frac{d}{n}\right)$ is the Jacobi symbol. This is related to $h(d)$, via the formula above, since

$$\sum_{n \text{ odd}} \frac{\left(\frac{d}{n}\right)}{n} = \prod_{p \text{ odd prime}} \left(1 - \left(\frac{d}{p}\right)\frac{1}{p}\right)^{-1}.$$

We consider M independent random variables Y_i. Each such random variable can take on any odd integer value n in the range $1 \le n \le d^2$, each with probability $\frac{\lambda}{n}$, i.e. for $1 \le i \le M$ we have

$$\text{Probability}\,\{Y_i = n\} = \frac{\lambda}{n} \quad \text{for } 1 \le n \le d^2 \text{ and } n \text{ odd},$$

where λ is defined by $\sum_{\substack{n\le d^2 \\ n \text{ odd}}} \frac{\lambda}{n} = 1$ (since the total probability must be 1).

Let X_i be the random variable $\left(\frac{d}{Y_i}\right)$ for $1 \le i \le M$.

We then look at the random variable $X_1 + X_2 + \cdots + X_M$. Its expected value is M times that of any one of the X_i's, as they are all independent and have the same distribution.

So we have

$$E(X_1 + X_2 + \cdots + X_M) = ME(X_1) = ME\left(\left(\frac{d}{Y_1}\right)\right)$$
$$= M \sum_{\substack{n \le d^2 \\ \text{odd } n}} \left(\frac{d}{n}\right) \cdot \frac{\lambda}{n} = \lambda M \sum_{\substack{n \le d^2 \\ \text{odd } n}} \frac{\left(\frac{d}{n}\right)}{n} = \lambda M S.$$

Therefore $S = \frac{1}{\lambda M} E(X_1 + \cdots + X_M)$; that is we can approximate S by summing up the M Jacobi symbols X_i that result from randomly choosing values for each random variable Y_i (with the probability distribution as described above).

An Approximation for $L(1,\chi)$. We have

$$L(1,\chi) = \prod_{\text{prime } p} \left(1 - \left(\frac{d}{p}\right)\frac{1}{p}\right)^{-1}$$
$$= \left(1 - \left(\frac{d}{2}\right)\frac{1}{2}\right)^{-1} \prod_{\text{odd prime } p} \left(1 - \left(\frac{d}{p}\right)\frac{1}{p}\right)^{-1}$$
$$= \left(1 - \left(\frac{d}{2}\right)\frac{1}{2}\right)^{-1} \sum_{\text{odd } n} \frac{\left(\frac{d}{n}\right)}{n}.$$

Here $\left(\frac{d}{n}\right)$ is the Jacobi symbol defined for odd integers n. (We remove the "2 factor" from the product, since this can be computed separately. This simplifies the problem in that we do not have to deal with even n, which would require defining the Kronecker symbol, an extension of the Jacobi symbol.) To compute $\left(\frac{d}{2}\right)$, we have

$$\left(\frac{d}{2}\right) = \begin{cases} 0 & \text{if } d \equiv 0 \mod 2; \\ 1 & \text{if } d \equiv \pm 1 \mod 8; \\ -1 & \text{if } d \equiv \pm 5 \mod 8. \end{cases}$$

Thus to approximate $L(1,\chi)$, we approximate the sum $\sum_{\text{odd } n} \frac{\left(\frac{d}{n}\right)}{n}$.

Proposition 2.1. *For any integer d which is not a square,*

$$\left| \sum_{\substack{n > d^2 \\ \text{odd } n}} \frac{\left(\frac{d}{n}\right)}{n} \right| \le \frac{8}{|d|}.$$

Proof. The proposition can be proved using elementary properties of the Legendre symbol and partial summation. □

We have

$$S = \sum_{\substack{n \le d^2 \\ \text{odd } n}} \frac{\left(\frac{d}{n}\right)}{n}.$$

Proposition 2.1 shows that the sum S is within $\frac{8}{|d|}$ of the complete sum, $\sum\limits_{\text{odd } n} \frac{\left(\frac{d}{n}\right)}{n}$. Hence an approximation to S will provide an only slightly weaker approximation to $\sum\limits_{\text{odd } n} \frac{\left(\frac{d}{n}\right)}{n}$.

We do this using the Random Summation Technique described above by taking $M = \lceil |d|^{1/5} \rceil$ random variables Y_i. The approximation we use for S is then $A = \frac{1}{\lambda \cdot M} \sum\limits_{i=1}^{M} X_i$. The following claim shows how good this approximation is.

Proposition 2.2. *Fix $\epsilon > 0$. Let $A = \frac{1}{\lambda \cdot M} \sum_{i=1}^{M} X_i$ (where the independent random variables X_i are as described above) and $S = \sum_{\substack{n \le d^2 \\ \text{odd } n}} \frac{\left(\frac{d}{n}\right)}{n}$. Then*

$$\textit{Probability}\left\{ |A - S| \ge \frac{1}{\lambda |d|^{1/10-\varepsilon}} \right\} < \frac{1}{|d|^{2\varepsilon}}.$$

Proof. We use Chebyshev's Inequality ([11]), which states that if X is a random variable with mean μ and variance σ^2, then

$$\text{Probability}\{|X - \mu| \ge \delta\} \le \frac{\sigma^2}{\delta^2} \tag{2.1}$$

for any $\delta > 0$.

We apply this to the random variable $X = \frac{1}{M} \sum\limits_{i=1}^{M} X_i$, which has mean $\mu = \lambda \cdot S$ and variance

$$\sigma^2 = \frac{1}{M} \left(\lambda \cdot \sum_{\substack{n \le d^2 \\ n \text{ odd} \\ (n,d)=1}} \frac{1}{n} - \mu^2 \right).$$

Taking $\delta = \frac{|d|^{\varepsilon}}{\sqrt{M}}$ we get from (2.1)

$$\text{Probability}\left\{ \left| \frac{1}{M} \sum_{i=1}^{M} X_i - \lambda S \right| \ge \frac{|d|^{\varepsilon}}{\sqrt{M}} \right\} \le \sigma^2 \cdot \frac{M}{|d|^{2\varepsilon}} < \frac{1}{|d|^{2\epsilon}}, \tag{2.2}$$

since

$$\sigma^2 < \frac{1}{M} \left(\lambda \sum_{\substack{n \le d^2 \\ n \text{ odd}}} \frac{1}{n} \right) = \frac{1}{M}.$$

The result follows as $\frac{|d|^{\varepsilon}}{\sqrt{M}} \le \frac{1}{|d|^{1/10-\varepsilon}}$. □

Lemma 2.3 ([5]). *Let $R_n = \sum_{i=1}^{n} \frac{1}{i} - \log\left(n + \frac{1}{2}\right)$. Then*

$$|R_n - \gamma| < \frac{1}{24n^2}$$

where $\gamma = \lim\limits_{n \to \infty} R_n$ is Euler's constant.

Lemma 2.4.

$$\frac{1}{\lambda} = \sum_{\substack{1 \le n \le d^2 \\ n \text{ odd}}} \frac{1}{n} \le 1 + \log|d|.$$

Proof. Let u be the biggest odd integer less than or equal to d^2. Then

$$\frac{1}{\lambda} = 1 + \frac{1}{3} + \frac{1}{5} + \ldots + \frac{1}{u} < 1 + \frac{1}{2}\int_1^u \frac{1}{x}\, dx < 1 + \frac{1}{2}\log u \le 1 + \log|d|. \quad \square$$

Proposition 2.5 . Let $M = \lceil |d|^{1/5} \rceil$. Fix $\epsilon > 0$. Then for $|d|$ sufficiently large,

$$\text{Probability}\left\{ \left| \sum_{\text{odd } n} \frac{\left(\frac{d}{n}\right)}{n} - \frac{1}{\lambda M}\sum_{i=1}^{M} X_i \right| < \frac{1}{|d|^{1/10-\varepsilon}} \right\} > 1 - \frac{1}{|d|^{\varepsilon}}.$$

Proof. Using Proposition 2.2 (with ϵ replaced by $\frac{\epsilon}{2}$), Proposition 2.1 and Lemma 2.4, we have, with probability $\ge 1 - \frac{1}{|d|^{\varepsilon}}$,

$$\left| \frac{1}{\lambda M}\sum_{i=1}^{M} X_i - \sum_{\text{odd } n} \frac{\left(\frac{d}{n}\right)}{n} \right| \le \left| \frac{1}{\lambda M}\sum_{i=1}^{M} X_i - \sum_{\substack{n \le d^2 \\ n \text{ odd}}} \frac{\left(\frac{d}{n}\right)}{n} \right| + \left| \sum_{\substack{n > d^2 \\ n \text{ odd}}} \frac{\left(\frac{d}{n}\right)}{n} \right|$$

$$\le \frac{1 + \log|d|}{|d|^{1/10-\varepsilon/2}} + \frac{8}{|d|} \le \frac{1}{|d|^{1/10-\epsilon}}$$

for $|d|$ sufficiently large. □

We now look at an approximation for $\frac{1}{\lambda}$.

Proposition 2.6 . *Let* $M = \lceil |d|^{1/5} \rceil$. *Define* $\overline{\lambda}$ *by*

$$\frac{1}{\overline{\lambda}} = \log\left(d^2 + \frac{1}{2}\right) - \frac{1}{2}\log\left(\left[\frac{d^2}{2}\right] + \frac{1}{2}\right) + \frac{1}{2}\sum_{i \le M} \frac{1}{i} - \frac{1}{2}\log\left(M + \frac{1}{2}\right).$$

Then

$$\left| \frac{1}{\lambda} - \frac{1}{\overline{\lambda}} \right| < \frac{1}{16M^2}.$$

Proof. We have

$$\frac{1}{\lambda} = \sum_{\substack{n \le d^2 \\ n \text{ odd}}} \frac{1}{n} = \sum_{i \le d^2} \frac{1}{i} - \sum_{2i \le d^2} \frac{1}{2i} = \sum_{i \le d^2} \frac{1}{i} - \frac{1}{2}\sum_{i \le \left[\frac{d^2}{2}\right]} \frac{1}{i}.$$

Therefore, with R_n as in Lemma 2.3

$$\frac{1}{\lambda} - \frac{1}{\overline{\lambda}} = R_{d^2} - \frac{1}{2}R_{[d^2/2]} - \frac{1}{2}R_M$$

so that by the triangle inequality

$$\left| \frac{1}{\lambda} - \frac{1}{\overline{\lambda}} \right| \le |R_{d^2} - \gamma| + \frac{1}{2}\left|R_{[d^2/2]} - \gamma\right| + \frac{1}{2}|R_M - \gamma|$$

$$< \frac{1}{24d^4} + \frac{1}{48 \cdot \left(\left[\frac{d^2}{2}\right]\right)^2} + \frac{1}{48M^2} < \frac{3}{48M^2} = \frac{1}{16M^2},$$

for $|d| > 4$, by Lemma 2.3.

Thus we now have an approximation for $\sum_{\text{odd } n} \frac{\left(\frac{d}{n}\right)}{n}$, namely

$$\overline{A} = \frac{1}{\overline{\lambda} M} \sum_{i=1}^{M} X_i$$

and as a result we have an approximation for $L(1,\chi)$ that we denote by $\overline{L(1,\chi)}$. We now give an algorithm to compute $\overline{L(1,\chi)}$.

Algorithm 2.7 (Approximation for $L(1,\chi)$).

1. Compute $M = \lceil |d|^{1/5} \rceil$.
2. Compute $\overline{\lambda}$, where

$$\frac{1}{\overline{\lambda}} = \log\left(d^2 + \frac{1}{2}\right) - \frac{1}{2}\log\left(\left[\frac{d^2}{2}\right] + \frac{1}{2}\right) + \frac{1}{2}\sum_{i \le M} \frac{1}{i} - \frac{1}{2}\log\left(M + \frac{1}{2}\right).$$

3. Choose M random odd integers Y_i, with $1 \le Y_i \le d^2$ and

$$\text{Probability}\,\{Y_i = n\} = \frac{\lambda}{n} \quad \text{for } 1 \le n \le d^2 \text{ and } n \text{ odd},$$

and where λ is defined by $\sum\limits_{\substack{n \le d^2 \\ n \text{ odd}}} \frac{\lambda}{n} = 1$. (See section 7 for details on this step.)

4. Compute the Jacobi symbols $X_i = \left(\frac{d}{Y_i}\right)$ for $1 \le i \le M$.
5. Compute

$$\overline{A} = \frac{1}{\overline{\lambda} M} \sum_{i=1}^{M} X_i.$$

6. $\overline{L(1,\chi)} = \left(1 - \left(\frac{d}{2}\right)\frac{1}{2}\right)^{-1} \overline{A}$.

Using the above approximation for $L(1,\chi)$ in Dirichlet's formula (1.1), we get an approximation for h in the case when $d < 0$ and an approximation for hR in the case of $d > 0$. In the rest of this section we discuss only the real case ($d > 0$) as the case when $d < 0$ is analogous (see section 7). We have from (1.1) $hR = \sqrt{d} L(1,\chi)$. The approximation we use then for hR is

$$\overline{hR} = \sqrt{d}\left(1 - \left(\frac{d}{2}\right)\frac{1}{2}\right)^{-1} \cdot \frac{1}{\overline{\lambda}} \cdot \frac{1}{M} \sum_{i=1}^{M} X_i, \tag{2.3}$$

where $M = \lceil d^{1/5} \rceil$ and each X_i is a random variable which satisfies

$$\text{Probability}\left\{X_i = \left(\frac{d}{n}\right)\right\} = \frac{\lambda}{n} \quad \text{for } 1 \le n \le d^2,\ n \text{ odd}. \quad \square$$

Theorem 2.8. *Fix $\epsilon > 0$. Then, for d sufficiently large, we have*

$$\text{Probability}\{|hR - \overline{hR}| < d^{2/5+\epsilon}\} > 1 - \frac{1}{d^{\epsilon}}$$

where $\overline{hR}$ is described above in (2.3).

Proof. We have

$$
\begin{aligned}
|hR - \overline{hR}| &= \sqrt{d}\left(1 - \left(\frac{d}{2}\right)\frac{1}{2}\right)^{-1}\left|\sum_{n \text{ odd}} \frac{\left(\frac{d}{n}\right)}{n} - \frac{1}{\overline{\lambda}} \cdot \frac{1}{M}\sum_{i=1}^{M} X_i\right| \\
&\leq \sqrt{d}\left(1 - \left(\frac{d}{2}\right)\frac{1}{2}\right)^{-1}\left\{\left|\sum_{n \text{ odd}} \frac{\left(\frac{d}{n}\right)}{n} - \frac{1}{\lambda M}\sum_{j=1}^{M} X_i\right| + \left|\frac{\sum_{i=1}^{M} X_i}{M}\right|\left|\frac{1}{\lambda} - \frac{1}{\overline{\lambda}}\right|\right\} \\
&\leq \sqrt{d}\cdot 2\left\{\frac{1}{d^{1/10-\epsilon}} + \frac{1}{16M^2}\right\}
\end{aligned}
$$

with a probability bigger than $1 - \frac{1}{d^{\varepsilon}}$, using Proposition 2.5 and Proposition 2.6 and the fact that $\left|\sum\limits_{i=1}^{M} X_i\right| \leq M$. Now, since $\frac{1}{16M^2} \leq \frac{1}{2d^{1/10}}$, we have

$$
|hR - \overline{hR}| < \frac{3\sqrt{d}}{d^{1/10-\epsilon}} < d^{2/5+\varepsilon}. \quad \square
$$

We now look at the running time for computing $\overline{hR}$.

Theorem 2.9. *The running time for computing $\overline{hR}$ using (2.3) is $O(d^{1/5+\epsilon})$.*

The two major computations are the computations of $\frac{1}{\lambda}$ and the evaluation of the $[d^{1/5}]$ Jacobi symbols, $X_i = \left(\frac{d}{Y_i}\right)$. Now $\frac{1}{\lambda}$ is comprised of a few logs and $[d^{1/5}]$ reciprocals. There are efficient algorithms (see [1]) to compute logs and reciprocals in time $O(\log^2 d)$, so the running time to determine $\frac{1}{\lambda}$ is $O(d^{1/5+\epsilon})$.

The Jacobi symbol $\left(\frac{a}{b}\right)$ can be computed in time $O(\log^2 d)$ ([13]). As we compute $[d^{1/5}]$ of these with $a, b \leq d^2$ the running time here would be $O(d^{1/5+\epsilon})$.

So the running time for computing $\overline{hR}$ is $O(d^{1/5+\epsilon})$.

3. Computations in real quadratic fields

In the following two sections, d will denote a positive integer that is a fundamental discriminant and all forms are binary quadratic forms of discriminant d. R is the regulator of the real quadratic field $\mathbb{Q}(\sqrt{d})$.

We assume that the reader is familiar with the theory of binary quadratic forms ([2] and [3]). We write $f \circ g$ for the composition of two forms f and g and fg for the product, which is the form obtained by composition of f and g followed by reduction. We write $(f^n)_\circ$ for the composition of f with itself n times and f^n for $(f^n)_\circ$ followed by reduction.

We fix a form in the principal cycle and denote this form by 1.

We bring the attention of the reader to the fact that all constants here are effective unless stated otherwise, i.e. $a \ll b$ (or $a = O(b)$) means there is a computable absolute constant k, such that $|a| < kb$. Thus all algorithms are deterministic.

Also $\epsilon > 0$ is any arbitrarily small real number.

We use the infrastructure of real quadratic fields, discovered by Shanks ([15]). The notations used are explained below. For further details the reader is referred to [8] and [12].

The notation $\delta(f, g)$ stands for the distance defined modulo R between two forms. $\delta_0(f, g)$ is the unique number which satisfies $0 \leq \delta_0(f, g) < R$ and $\delta_0(f, g) \equiv \delta(f, g)$ mod R.

$\rho(f)$ denotes the form that is right adjacent to the form f on the cycle containing f. Similarly $\rho^{-1}(f)$ stands for the form left adjacent to f.

Lastly we point out that in all our computations, d is assumed to be sufficiently large so that all fixed powers of $\log d$ are absorbed into d^ϵ.

Lemma 3.1. *Let F and G be forms on a cycle. Let x be a real number such that $\delta(F,G) \equiv x \mod R$ and $|x| < R$. Then either $\delta_0(F,G) = |x|$ or $\delta_0(G,F) = |x|$.*

Proof. The proof follows using the definition of distance. □

Lemma 3.2. *Let F and G be forms on a cycle of length ℓ such that $\rho^n(F) = G$, where $0 < n < \ell$. Then $n < \frac{4\delta_0(F,G)}{\log 2} + 1$.*

Proof. We have that $\delta_0(F,G) = \sum_{i=0}^{n-1} \delta_0(\rho^i(F), \rho^{i+1}(F))$. As the sum of any two consecutive summands here is greater than $\frac{\log 2}{2}$ ([8]), we have

$$\delta_0(F,G) > \begin{cases} \frac{n \log 2}{4} & \text{if } n \text{ is even },\\ \frac{(n-1)\log 2}{4} & \text{if } n \text{ is odd }, \end{cases}$$

which gives $\delta_0(F,G) > \frac{(n-1)\log 2}{4}$ and hence $n < \frac{4\delta_0(F,G)}{\log 2} + 1$. □

Lemma 3.3. *Given distinct forms F and G on a cycle, $\delta_0(F,G)$ can be computed in time $O(\delta_0(F,G)\log^2 d)$.*

Proof. Let ℓ be the length of the cycle. Then there is an integer n with $0 < n < \ell$, such that $\rho^n(F) = G$. We compute the forms $\rho(F), \rho^2(F), \rho^3(F), \dots, \rho^n(F) = G$, keeping track of the distances. Then

$$\delta_0(F,G) = \sum_{i=0}^{n-1} \delta_0(\rho^i(F), \rho^{i+1}(F)).$$

Now the n reductions can be computed in time $O(n \log^2 d)$. Computing the distances takes time $O(n\log^2 d)$ using efficient algorithms as in [1]. By Lemma 3.2, we have $n < \frac{4\delta_0(F,G)}{\log 2} + 1$. Hence the total time taken is $O(\delta_0(F,G)\log^2 d)$. □

Lemma 3.4. *Given a real number x with $R > x > 0$ and a form F on a given cycle, we can compute forms f and f', with $\delta_0(F,f) = x + E_1$ and $\delta_0(f',F) = x + E_2$, where $|E_1|, |E_2| < \frac{\log d}{2}$, in time $O(x\log^2 d)$.*

Proof. Starting with F on the given cycle we compute the forms

$$F, \rho(F), \rho^2(F), \dots$$

keeping track of the distances, till we reach a form, say $\rho^{n+1}(F)$, whose distance from F is at least x. Let $f = \rho^n(F)$. It can be shown that f satisfies the conditions in the theorem.

By Lemma 3.2, $n < \frac{4\delta_0(F,f)}{\log 2} + 1 = O(x)$. As each reduction takes time $O(\log^2 d)$, the total time taken is $O(x\log^2 d)$.

The proof for f' is similar, only we use ρ^{-1} instead of ρ. □

Lemma 3.5. *Let x be a real number with $0 < x = O(d)$. Suppose $R \gg \log^2 d$. Then we can find a form G on the principal cycle with*

$$\delta(1,G) \equiv x + y \mod R$$

where $y = O(\log d)$ in time $O(\log^5 d)$.

Proof. Let n be the largest power of 2 that is smaller than x, i.e., $2^n \leq x < 2^{n+1}$. Thus $1 \leq x/2^n \leq 2$. Let $f_0 = \rho(1)$ so that $\delta_0(1, f_0) = \frac{x}{2^n} + E_0$, where $|E_0| < \frac{\log d}{2}$. It can be shown that given a form f such that $\delta(1, f) \equiv \frac{x}{2^k} + E \mod R$ with $|E| = O(\log^2 d)$, we can determine a form f', such that $\delta(1, f') \equiv \frac{x}{2^{k-1}} + E' \mod R$ with $|E'| < \frac{\log d}{2}$, in time $O(\log^4 d)$, where f' is computed by basically squaring the form f. But then Lemma 3.5 follows by an induction hypothesis, for given f_0 as above, we just successively compute $f_1, f_2, \dots, f_n$ with $\delta(1, f_j) \equiv \frac{x}{2^{n-j}} + E_j \mod R$ where each $|E_j| < \frac{\log d}{2}$. Evidently the time taken will then be $O(n \log^4 d)$ and the result follows since $n = O(\log d)$. □

Lemma 3.6. *Let G be a form on the principal cycle, such that $2d^{1/5} < \delta_0(1, G) < \frac{R}{3}$. Then we can find an approximation $\overline{\delta_0(1, G)}$ for $\delta_0(1, G)$ with*

$$\delta_0(1, G) = \overline{\delta_0(1, G)} + O(\log d)$$

in time $O\left(d^{1/5} \log^2 d + \frac{\delta_0(1,G) \log^4 d}{d^{1/5}}\right)$.

Proof. Let m and n be integers such that

$$[\delta_0(1, G)] = m[d^{1/5}] + n \quad \text{with} \quad 0 \leq n < [d^{1/5}]. \tag{3.1}$$

We first find a form f with $\delta_0(1, f) = [d^{1/5}] + O(\log d)$, using Lemma 3.4. Let $k > 1$ be the least integer such that $\sum_{i=0}^{k-1} \delta_0(f^i, f^{i+1}) > \delta_0(1, G)$. Let $f^{k-1} = \rho^{-n}(G)$. Then from Lemma 3.2 and (3.1) above we have $n = O(d^{1/5})$. To find k, we compute the baby steps,

$$G, \rho^{-1}(G), \rho^{-2}(G), \dots, \rho^{-n}(G),$$

and the giant steps,

$$f, f^2, f^3, \dots, f^{2m+2},$$

compare the lists and find a match. Using Lemma 3.3, $\delta_0(f^{k-1}, G)$ can be found in time $O(d^{1/5} \log^2 d)$.

Using Lemma 3.5, we can find a form G' in time $O(\log^5 d)$, with

$$\delta(1, G') \equiv (k-1)[d^{1/5}] + x \mod R \tag{3.2}$$

where $x = O(\log d)$.

We can also find $\delta_0(G', f^{k-1})$ in time $O(k \log^4 d) = O(\frac{\delta_0(1,G) \log^4 d}{d^{1/5}})$, by Lemma 3.3.

We then have

$$\delta_0(1, G) = \delta_0(1, G') + \delta_0(G', f^{k-1}) + \delta_0(f^{k-1}, G) = \overline{\delta_0(1, G)} + O(\log d),$$

where $\overline{\delta_0(1, G)} = (k-1)[d^{1/5}] + \delta_0(G', f^{k-1}) + \delta_0(f^{k-1}, G)$. □

Lemma 3.7. *Let $R \gg \log d$. Given an integer x with $0 < x = O(d)$, it takes time $O(\log^5 d)$ to check if $x \equiv nR + e$ for some integer n and $e = O(\log d)$.*

Proof. By Lemma 3.5 we can find a form F in time $O(\log^5 d)$, with

$$\delta(1, F) \equiv x + y \mod R,$$

where $y = O(\log d)$. If $x \equiv nR + e \mod R$, then $\delta(1, F) \equiv nR + e + y \mod R$. By Lemma 3.1 either $\delta_0(1, F) = O(\log d)$ or $\delta_0(F, 1) = O(\log d)$.

Consider the case when $\delta_0(1, F) = O(\log d)$. Let $F = \rho^m(1)$ for some integer m with $0 \leq m < \ell$, where ℓ is the length of the cycle. Then to find $\delta_0(1, F)$ we must perform m reductions starting from 1. From Lemma 3.2 we have $m < \frac{4\delta_0(1,F)}{\log 2} + 1 = O(\log d)$. Hence the total time taken is $O(\log^5 d)$.

The case when $\delta_0(F, 1) = O(\log d)$ is dealt with similarly.

If neither $\delta_0(1, F)$ nor $\delta_0(F, 1)$ is $O(\log d)$, then $x \neq nR + O(\log d)$ for any integer n. □

4. Computation of R

Proposition 4.1. *Let d be a fundamental discriminant. If R is the regulator of a real quadratic field $\mathbb{Q}(\sqrt{d})$ and h is the class number, then $hR < \sqrt{d} \log d$.*

Proof. Hua ([7]) showed that $L(1, \chi) < \frac{1}{2} \log d + 1 < \log d$. By Dirichlet's formula (1.1) for $d > 0$, we have $hR = \sqrt{d} L(1, \chi) < \sqrt{d} \log d$. □

Theorem 4.2. *A suitable approximation to the regulator R can be computed deterministically in expected time $O(d^{1/5+\epsilon})$ (in fact the approximation will differ from R by at most $O(d^{-1/2})$).*

Proof. This is a three step procedure. First let us assume that $R = O(d^{1/5} \log^2 d)$.

Step 1. $R = O(d^{1/5} \log^2 d)$.

Starting from the form 1, we cycle through the principal cycle keeping track of the distances till we reach the form 1 again. If we find a form F on the principal cycle such that $\delta_0(1, F) > d^{1/5} \log^2 d$, then $R > d^{1/5} \log^2 d$ and we go to step 2.

Step 2. $R \gg d^{1/5} \log^2 d$.

We have Dirichlet's formula for $d > 0$:

$$hR = \sqrt{d} \prod_{p \text{ prime}} \left(1 - \left(\frac{d}{p}\right) \frac{1}{p}\right)^{-1}.$$

We find an approximation for the product above by using random summation on the corresponding sum as discussed in Section 2. This gives an approximation for hR with an error of $O(d^{2/5+\epsilon})$ in time $O(d^{1/5+\epsilon})$. Thus we have

$$hR = \overline{\lambda} + E \tag{4.1}$$

where $E = O(d^{2/5+\epsilon})$.

By Lemma 3.5 we can find a form F in time $O(\log^5 d)$ with

$$\delta(1, F) \equiv \overline{\lambda} + x \equiv -E + x \mod R$$

where $x = O(\log d)$, by (4.1). Assume $E < 0$ (the case when $E > 0$ is similar); then using Lemma 3.6 we find an approximation $\overline{\delta_0(1, F)}$ for $\delta_0(1, F)$ such that

$$\delta_0(1, F) = \overline{\delta_0(1, F)} + O(\log d). \tag{4.2}$$

As $\delta(1, F) \equiv \delta_0(1, F) \mod R$ and as $E = hR - \overline{\lambda}$ from (4.1), we have

$$\tilde{h}R = \overline{A} + O(\log d) \tag{4.3}$$

for some integer $\tilde{h}$, where $\overline{A} = \overline{\lambda} - \overline{\delta_0(1, F)}$.

Now if $\tilde{h}$ is divisible by integer q, then $\frac{\overline{A}}{q} = nR + O(\log d)$ for some integer n by (4.3). Conversely if $q \leq d^{1/5}$, and $\frac{\overline{A}}{q} = nR + O(\log d)$, then $\tilde{h} = qn$.

Whether or not $\frac{\overline{A}}{q} = nR + O(\log d)$ for some integer n (and thus whether q divides $\tilde{h}$), may be checked in time $O(\log^5 d)$ using Lemma 3.7. Thus we may test $\tilde{h}$ for divisibility by all primes $\leq d^{1/5}$ in time $O(d^{1/5+\epsilon})$. Dividing out all such primes (and their powers) from $\tilde{h}$; we will have determined R at this step (via (4.3)) provided all prime divisors of $\tilde{h}$ are less than $d^{1/5}$.

Let h_1 be the product of all the primes (with multiplicity) greater than $d^{1/5}$, that divide $\tilde{h}$; we may assume $h_1 \neq 1$, so $h_1 > d^{1/5}$. Let $A' = \frac{\overline{A}h_1}{\tilde{h}}$, so that

$$h_1 R = A' + O(\log d).$$

By Proposition 4.1, $hR < \sqrt{d} \log d$, thus from (4.1) we have $\overline{\lambda} = O(\sqrt{d}\log d)$. As $\delta_0(1, F) < R$, we obtain $\overline{\delta_0(1,F)} = O(R) = O(\sqrt{d}\log d)$ from (4.2). Thus $\overline{A} = \overline{\lambda} - \overline{\delta_0(1,F)} = O(\sqrt{d}\log d)$ and so $A' = O(\sqrt{d}\log d)$. Thus $h_1 R = O(\sqrt{d}\log d)$ which gives $R = O(\frac{\sqrt{d}\log d}{h_1}) = O(d^{3/10}\log d)$, as $h_1 > d^{1/5}$.

Now we move on to the third stage of the algorithm. As $d^{3/10}\log d = O(d^{2/5})$, we assume that $R = O(d^{2/5})$.

Step 3. $d^{1/5}\log^2 d \leq R = O(d^{2/5})$.

Let

$$[R] = m[d^{1/5}] + n \quad \text{where } 0 \leq n < d^{1/5} \text{ and } m = O(d^{1/5}). \tag{4.4}$$

We first compute, using Lemma 3.4, a form f on the principal cycle with

$$\delta_0(1, f) = [d^{1/5}] + O(\log d).$$

We take f^0 to be 1. Then for any $j \geq 1$, we have

$$\sum_{i=0}^{j-1} \delta_0(f^i, f^{i+1}) > \frac{j[d^{1/5}]}{2}.$$

Hence if $j \geq 2(m+1)$, then

$$\sum_{i=0}^{j-1} \delta_0(f^i, f^{i+1}) > (m+1)[d^{1/5}] > R$$

from (4.4).

Let $k \leq 2(m+1)$ be the smallest integer such that

$$\sum_{i=0}^{k-1} \delta_0(f^i, f^{i+1}) > R. \tag{4.5}$$

Let $f^k = \rho^s(1)$ so that, by Lemma 3.2, we have $s < \frac{4\delta_0(1,f^k)}{\log 2} + 1 = S$. To find k we compute the baby steps,

$$1, \rho(1), \rho^2(1), \dots, \rho^S(1),$$

and the giant steps,

$$f, f^2, f^3, \dots, f^{2(m+1)},$$

and find for a common element in the two lists.

We have now

$$R = \delta_0(1, f^{k-1}) + \delta_0(f^{k-1}, 1) = k[d^{1/5}] + O(d^{1/5}\log^2 d). \tag{4.6}$$

By Lemma 3.5 a form G can be found in time $O(\log^5 d)$, with

$$\delta(1, G) \equiv k[d^{1/5}] + y \mod R$$

where $y = O(\log d)$. Using (4.6), we have $\delta(1, G) \equiv R + \tilde{E} \mod R$, where $\tilde{E} = O(d^{1/5}\log^2 d)$. As $|\tilde{E}| < R$, by Lemma 3.1 we have either $\delta_0(1, G) = |\tilde{E}|$ or $\delta_0(G, 1) = |\tilde{E}|$. Using Lemma 3.3, $\tilde{E}$ can be found in time $O(\tilde{E}\log^2 d) = O(d^{1/5+\epsilon})$ and we have $R = \delta(1, G) - \tilde{E}$. □

The running time in Theorem 4.2 is the expected running time. This is because we get an approximation for hR using random summation. The algorithm to compute R is deterministic and if indeed the approximation for hR is correct, then the answer we get for R is correct. In the case when the interval provided by random summation is not correct, the algorithm does not give an answer. We then repeat random summation to get another interval. As the probability of getting a wrong interval using random summation is $< \frac{1}{d^\epsilon}$, the expected number of steps for getting a correct interval is $O(d^\epsilon)$ and thus R is computed in expected time $O(d^{1/5+\epsilon})$.

5. Computation of h

When $R \gg d^{2/5}$, we determined the value of h as one of the steps in calculating R during the algorithm presented in Theorem 4.2:

Theorem 5.1. *If $R \gg d^{2/5}$, then h can be found in deterministic time $O(d^{1/5+\epsilon})$ with probability greater than $1 - \frac{1}{d^\epsilon}$.*

Proof. We follow the procedure in Step 2 of Theorem 4.2. In this case we obtain that $\tilde{h} = h$. Now $\tilde{h} < \sqrt{d}\log d/R \leq d^{1/10}$ by Proposition 4.1. But $h = \tilde{h}$ is completely determined by the algorithm in Step 2 of Theorem 4.2 when it is this small. □

When $R \ll d^{2/5+\epsilon}$, we compute h by our version of Shanks' algorithm. To begin with, we approximate the value of $L(1, \chi)$ (where χ is the real, primitive, non-principal character mod d) using the Random Summation method, as discussed in Section 2. By Dirichlet's formula (1.1) for $d > 0$, this leads to an approximation for hR:

$$hR = \overline{\lambda} + O(d^{2/5+\epsilon}).$$

As R has already been computed we get an approximation for h :

$$h = \frac{\overline{\lambda}}{R} + O(\frac{d^{2/5+\epsilon}}{R}).$$

So we have now found an interval $(L, L + \tilde{l}\ )$ containing h, where $L = \frac{\overline{\lambda}}{R}$ and $\tilde{l} = O(\frac{d^{2/5+\epsilon}}{R})$. So far our algorithm has taken time $O(d^{1/5+\epsilon})$ (by Theorem 2.9 and Theorem 4.2).

The second part of the algorithm is to determine a subgroup of the class group of order $\geq \tilde{l}$. We do this by 'randomly' choosing forms, which we hope lie outside the subgroup that we have already obtained, so building an even bigger subgroup. There are two practical difficulties that we need to discuss in detail: First, given a subgroup H and a form g, how do we determine the size of the subgroup generated by H together with g? This is what we will do in the rest of this section. Second, we need to be precise about what we mean by 'randomly' choosing forms, and we also need to analyse the probability that our 'randomly chosen' form will lie outside the subgroup that we have already generated. We discuss this in section 6, where we also show that this part of the algorithm runs in expected time $O(d^{1/5+\epsilon})$ (see the Main Theorem in section 6).

In the remainder of this section we will prove the following result.

Theorem 5.2. *The running time for computing the order of a given subgroup of the class group of a real quadratic field, given a set of $O(d^\epsilon)$ generators of the subgroup, using baby-steps-giant-steps, is $O(d^{1/5+2\epsilon})$.*

Lemma 5.3. *Let $d^{1/5} \ll R = O(d^{2/5+\epsilon})$. Let F be a reduced form. Then in time $O(\frac{R\log^2 d}{d^{1/5}})$ it can be checked if F is a principal form.*

Proof. Let

$$[R] = m[d^{1/5}] + n \quad \text{with } 0 \leq n < d^{1/5}. \tag{5.1}$$

We first compute using Lemma 3.4, a form f on the principal cycle with

$$\delta_0(1, f) = [d^{1/5}] + O(\log d)$$

in time $O(d^{1/5}\log^2 d)$. In exactly the same manner as in Theorem 4.2, Step 3 we define the integer $k \leq 2(m+1)$ as the smallest integer such that

$$\sum_{i=0}^{k-1} \delta_0(f^i, f^{i+1}) > R. \tag{5.2}$$

If $F \sim 1$, then F and 1 lie on the same cycle. So let r be the smallest integer such that

$$\sum_{i=0}^{r} \delta_0(f^i, f^{i+1}) > \delta_0(1, F). \tag{5.3}$$

Clearly $r \leq k-1$ from (5.2). Now

$$\delta_0(F, f^{r+1}) \leq \delta_0(F, f^{r+1}) + \delta_0(f^r, F) = \delta_0(f^r, f^{r+1}).$$

We can also show that

$$\delta_0(F, f^{r+1}) \leq [d^{1/5}] + O(\log^2 d). \tag{5.4}$$

Let t be the least non-negative integer such that

$$f^{r+1} = \rho^t(F). \tag{5.5}$$

From Lemma 3.2 and (5.4), we have

$$t < \frac{4([d^{1/5}] + O(\log^2 d))}{\log 2} + 1 = T. \tag{5.6}$$

We now compute the baby steps:

$$F, \rho(F), \rho^2(F), \rho^3(F), \dots, \rho^T(F)$$

and the giant steps:

$$f, f^2, f^3, \dots, f^k.$$

If $F \sim 1$, then from (5.5) it is clear that a match will be found in the two lists above.

From (5.6), we see that the number of baby steps is $O(d^{1/5})$ and hence the baby list is computed in time $O(d^{1/5}\log^2 d)$.

Now $k \leq 2(m+1)$ and from (5.1), we have $m < \frac{[R]}{[d^{1/5}]}$, thus the number of giant steps is $O(\frac{R}{d^{1/5}}) = O(d^{1/5+\epsilon})$, as $R = O(d^{2/5+\epsilon})$. Thus the giant list takes time $O(d^{1/5+\epsilon})$ to compute, as a product can be computed in time $O(\log^3 d)$.

Thus if $F \sim 1$, then a match is found between the baby forms and giant forms in time $O(d^{1/5+\epsilon})$. If a match is not found in time $O(d^{1/5+\epsilon})$, then F and 1 are not in the same cycle and hence F is not principal. □

Finding the Order of an Element f. We first find a number n, in the interval $(L, L+\tilde{l})$ such that $f^n = 1$:

Suppose $L < n < L + \tilde{l}$ and $f^n = 1$. Write $n = L + s$, with $0 < s < \tilde{l}$. Thus we wish to find s such that

$$f^{L+s} = 1, \qquad 0 < s < \tilde{l}.$$

We now present an algorithm to do this in time $O(d^{1/5+\epsilon})$; there are two cases, depending on the size of R.

Case 1. $R \ll d^{1/5+\epsilon}$.

Let $u = \left[\tilde{l}/d^{1/5}\right]$. We can write $s = a + bu$, with $0 \leq a < u$, and $0 \leq b \leq [d^{1/5}]$. Then

$$f^n = f^{L+s} = f^L f^{a+bu} = 1$$

or

$$f^L f^a = (f^{-u})^b.$$

Let $g = f^{-u}$. To find a and b, we compute the elements

$$f^L, f^L f, f^L f^2 \dots f^L f^{u-1} \tag{5.7}$$

and

$$1, g, g^2, \dots, g^{[d^{1/5}]}. \tag{5.8}$$

This requires computing $O(d^{1/5+\epsilon})$ products as $\tilde{l} = O(d^{2/5+\epsilon})$. Hence the time taken is $O(d^{1/5+\epsilon})$ since a product can be computed in time $O(\log^3 d)$.

For each form f in the list (5.7) we compute all the reduced forms equivalent to f (i.e., all the elements in its cycle) and compare this new list with the list (5.8) for a common element. Now if l is the length of a cycle, then $l < \frac{4R}{\log 2}$ ([8]); hence each cycle is computed in time $O(R\log^2 d)$. Thus the time taken for computing all of the cycles is $O(uR\log^2 d) = O(\frac{\tilde{l}R\log^2 d}{d^{1/5}})$, where $u = \left[\frac{\tilde{l}}{d^{1/5}}\right]$ as in the imaginary case. As $\tilde{l} = O(\frac{d^{2/5+\epsilon}}{R})$, the total time taken is $O(d^{1/5+2\epsilon})$.

Case 2. $d^{1/5+\epsilon} \ll R \ll d^{2/5+\epsilon}$.

We wish to find an integer s such that

$$f^{L+s} = 1, \qquad 0 < s < \tilde{l}.$$

We compute the elements

$$f^{L+1}, f^{L+2}, \dots, f^{L+\tilde{l}-1}.$$

For each of these elements we check if it is in the principal cycle using Lemma 5.3. This is done in time $O(\frac{\tilde{l}R\log^2 d}{d^{1/5}}) = O(d^{1/5+2\epsilon})$, as $\tilde{l} = O(\frac{d^{2/5+\epsilon}}{R})$.

Thus we have found an integer n in $(L, L+\tilde{l})$ with $f^n = 1$. To find the order of f, we first factor n as $n = p_1^{e_1}\dots p_r^{e_r}$. To find the order of f, for each prime p dividing n, we find the exact power of p that divides $o(f)$, as follows: We compute the powers, $f^{\frac{n}{p}}, f^{\frac{n}{p^2}}$, and so on and check each time if the form is in the principal cycle. If $f^{\frac{n}{p^r}}$ is in the principal cycle and $f^{\frac{n}{p^r}}$ is not, then p^{e-r} is the highest power of p dividing $o(f)$, where p^e is the highest power of p in n.

Computing a power of f takes time $O(\log^4 d)$ by using repeated squaring method.

By Lemma 5.3, it takes time $O(\frac{R\log^2 d}{d^{1/5}})$ to check if an element is in the principal cycle. As $e, r = O(\log n)$, we perform $O(\log^2 d)$ such checks and thus the time taken to find the order of f is $O(\frac{R\log^4 d}{d^{1/5}}) = O(d^{1/5+2\epsilon})$, as $R = O(d^{2/5+\epsilon})$.

Finding the Order of a Subgroup. Suppose we wish to find the order of the subgroup $\langle f, g\rangle$. Let $o_f(g)$ denote the smallest power of g that is also a power of f (note that this means the reduced form in the class of that power of g). Then the order of $\langle f, g\rangle$ is $o(f)\cdot o_f(g)$. So now we wish to find $o_f(g)$. We know $o_f(g)$ divides $o(g)$, so we first compute $o(g)$ and then factor it and find the highest power of each prime p which divides $o_f(g)$, as follows:

We compute $g^{o(g)/p}, g^{o(g)/p^2}, \dots$. Each time we check whether we get a power of f, that is if the power of g lies in the subgroup generated by f (see next paragraph). If $g^{o(g)/p^r}$ is a power of f, but $g^{o(g)/p^{r+1}}$ is not, and p^e is the highest power of p dividing $o(g)$, then p^{e-r} is the highest power of p dividing $o_f(g)$.

The powers of g can be computed using repeated squaring method, in time $O(\log^4 d)$.

Suppose we have picked k forms. Let $G_k = \langle f_1, \dots, f_k\rangle$ be the subgroup generated by $f_1, \dots, f_k$ and let N_k denote its order. Then the current subgroup is G_k. Let $o_G(f)$ denote the order of f in G, i.e. the smallest power of f that is in G. Let $\theta_s = o_{G_{s-1}}(f_s)$ for each $s = 1, 2, \dots, k$. Then $N_k = \theta_1\theta_2\dots\theta_k$.

We assume here that k is at least 2. If N_k is less than $\tilde{l}$, then we have still not determined the class number. So we pick another form, say g and find the order of $\langle f_1, \dots, f_k, g\rangle$, which equals $N_k\cdot o_{G_k}(g)$. Therefore we wish to find $o_{G_k}(g)$. From the discussion at the beginning of this section, this requires computing certain powers of g and checking if they do or do not lie in G_k.

Determining if a Given Form Belongs to a Given Subgroup.

Lemma 5.4. *Let $k \geq 2$ and $N_k = \theta_1\theta_2\dots\theta_k$. Let t be such that $\theta_1\theta_2\dots\theta_t \leq d^{1/5}$ and $\theta_1\dots\theta_{t+1} \geq d^{1/5}$. If $m = \left[\frac{d^{1/5}}{\theta_1\dots\theta_t}\right]$, then*

$$\theta_1\dots\theta_t\cdot m \leq d^{1/5} \text{ and } \left(\left[\frac{\theta_{t+1}}{m}\right]+1\right)\theta_{t+2}\dots\theta_k \leq 3\frac{N_k}{d^{1/5}}.$$

Proof. Note first that

$$\theta_1\dots\theta_t\cdot m = \theta_1\dots\theta_t\left[\frac{d^{1/5}}{\theta_1\dots\theta_t}\right] \leq d^{1/5}.$$

Since $[x] > \frac{x}{2}$ for any $x \geq 1$, we get $m > \frac{d^{1/5}}{2\theta_1 \ldots \theta_t}$. Therefore

$$\theta_{t+2} \ldots \theta_k \left(\left[\frac{\theta_{t+1}}{m} \right] + 1 \right) < \theta_{t+2} \ldots \theta_k \left(\frac{\theta_{t+1}}{m} + 1 \right) = \frac{\theta_{t+1} \ldots \theta_k}{m} + \theta_{t+2} \ldots \theta_k$$
$$< \frac{2\theta_1 \ldots \theta_t}{d^{1/5}} \cdot \theta_{t+1} \ldots \theta_k + \theta_{t+2} \ldots \theta_k = \frac{2N_k}{d^{1/5}} + \theta_{t+2} \ldots \theta_k \leq 3\frac{N_k}{d^{1/5}}$$

since $\theta_{t+2} \ldots \theta_k \leq \frac{N_k}{d^{1/5}}$. □

Suppose we wish to check if g^c is in G_k, i.e. if there exist integers c_i, with $0 \leq c_i < \theta_i$, such that

$$g^c = f_1^{c_1} \ldots f_k^{c_k}, \quad \text{for some } 0 \leq c_i < \theta_i. \tag{5.9}$$

Case 1. $R \ll d^{1/5+\epsilon}$. We write $f_1^{c_1} \ldots f_k^{c_k}$ as

$$f_1^{c_1} \ldots f_t^{c_t} \cdot f_{t+1}^{a+bm} f_{t+2}^{c_{t+2}} \ldots f_k^{c_k}$$

where $0 \leq a < m$, $0 \leq b \leq \left[\frac{\theta_{t+1}}{m}\right]$.

Then we have from (5.9)

$$g^c \cdot f_1^{-c_1} \ldots f_t^{-c_t} \cdot f_{t+1}^{-a} = \left(f_{t+1}^m\right)^b \cdot f_{t+2}^{c_{t+2}} \ldots f_k^{c_k}. \tag{5.10}$$

To find c_i, for $1 \leq i \leq k$, we make the two lists:

$$g^c \cdot f_1^{-a_1} \ldots f_t^{-a_t} \cdot f_{t+1}^{-a'}, \ 0 \leq a_i < \theta_i \text{ for } 1 \leq i \leq t \text{ and } 0 \leq a' < m \tag{5.11}$$

and

$$\left(f_{t+1}^m\right)^{b'} \cdot f_{t+2}^{a_{t+2}} \ldots f_k^{a_k}, \ 0 \leq a_i < \theta_i \text{ for } t+1 \leq i \leq k \text{ and } 0 \leq b' \leq \left[\frac{\theta_{t+1}}{m}\right] \tag{5.12}$$

For each element in the list (5.12) we compute all the elements in its cycle and compare this list with (5.11). By Lemma 5.19 each cycle has length $O(R)$, so time taken to compute a cycle is $O(R \log^2 d)$ by Theorem 2.8. By Lemma 5.4 there are less than $3\frac{N_k}{d^{1/5}}$ elements in the list (5.12), thus as $N_k < \tilde{l}$ the computation of all the cycles takes time $O(\frac{R\tilde{l}\log^2 d}{d^{1/5}})$.

Case 2. $d^{1/5+\epsilon} \ll R \ll d^{2/5+\epsilon}$.

In this case to check if (5.9) holds, we compute the elements

$$g^{-c} f_1^{c_1} \ldots f_k^{c_k}, \quad 0 \leq c_i < \theta_i,$$

and check for each entry, if it is in the principal cycle. By Lemma 5.3 each check takes time $O(\frac{R\log^2 d}{d^{1/5}})$, so the total time taken is $O(\frac{\tilde{l}R\log^2 d}{d^{1/5}})$, as there are N_k elements in the list and $N_k < \tilde{l}$.

Going back to computing $o_{G_k}(g)$, as $e, r = O(\log n)$, we need to check for $O(\log^2 d)$ forms, if they lie in G_K. Thus the time taken is $O(\frac{R\tilde{l}\log^4 d}{d^{1/5}}) = O(d^{1/5+2\epsilon})$ as $\tilde{l} = O(\frac{d^{2/5+\epsilon}}{R})$.

Tying together the results above, we have proved Theorem 5.2.

6. Probability analysis

Choosing a Random Form. We now present an algorithm to choose a 'random' form from the class group and give a lower bound for the probability that this form lies outside a given subgroup. This algorithm chooses a form (A, B, C) of discriminant d with $1 \leq B \leq d^2$.

Algorithm 6.1. We will choose a binary quadratic form (A, B, C) of discriminant d with

$$1 \leq B \leq d^2 \text{ and } 1 \leq A \leq q, \quad \text{where } q = \sqrt{\frac{d^4 - d}{4}}.$$

Step 1. Choose B from 1 to d^2 with uniform distribution, i.e. select any given integer B in the range $1 \leq B \leq d^2$, with probability $\frac{1}{d^2}$.

Step 2. Factor $|\frac{B^2-d}{4}|$ (using the methods in [9]). Let $|\frac{B^2-d}{4}| = p_1^{c_1} \ldots p_k^{c_k}$ be the prime factorization.

Step 3. Select a random factor $A \leq q$ as follows.
Choose k random numbers $r_1, r_2, \ldots, r_k$, where $0 \leq r_i \leq c_i$.
Take $A_1 = p_1^{r_1} \ldots p_k^{r_k}$.
If $A_1 \leq q$, then let $A = A_1$. Otherwise, repeat step 3.

Step 4. $C = \frac{B^2-d}{4A}$.

Let $\epsilon > 0$ be fixed. Let τ be the divisor function. Then given any one particular form f, the probability of choosing a form (A, B, C) equivalent to f using the above Algorithm 6.1 is :

$$\frac{1}{d^2} \sum_{B=1}^{d^2} \frac{1}{\tau(\frac{B^2-d}{4})} \sum_{\substack{A \mid \frac{B^2-d}{4} \\ 1 \leq A \leq q}} \begin{cases} 1 & (A, B) \sim f, \\ 0 & \text{else} \end{cases}$$

$$\geq \frac{1}{|d|^\epsilon d^2} \sum_{\substack{1 \leq B \leq d^2 \\ A \mid \frac{B^2-d}{4} \\ 1 \leq A \leq q}} \begin{cases} 1 & (A, B) \sim f, \\ 0 & \text{else} \end{cases}$$

$$= \frac{1}{|d|^\epsilon d^2} \#\{(A, B) \sim f : 1 \leq B \leq d^2, 1 \leq A \leq q\},$$

since $\frac{B^2-d}{4} < d^4$ so that $\tau\left(\frac{B^2-d}{4}\right) \leq |d|^\epsilon$ for $|d|$ sufficiently large [see Theorem 315 in [6]].

Let $\mathcal{A} = \{(A, B) \sim f : 1 \leq B \leq d^2, 1 \leq A \leq q\}$. We proved above that

$$\text{Prob}\{(A, B, C) \sim f\} \geq \frac{1}{|d|^{\epsilon+2}} |\mathcal{A}|. \tag{6.1}$$

We define now an equivalence relation '∇' on the set $\mathcal{A}$ as follows: $(A_1, B_1, C_1) \nabla (A_2, B_2, C_2)$ if and only if $A_1 = A_2 = A$ and $B_1 \equiv B_2 \mod 2A$. Hence each equivalence class is represented by a unique form (A, B, C) with $1 \leq B \leq 2A$. The equivalence class represented by (A, B, C), $1 \leq B \leq 2A$, is

$$\{(A, B + 2Ax) : x \in \mathbb{Z}, 1 \leq B + 2Ax \leq d^2\};$$

the number of elements in this set is $\left[\frac{d^2-B}{2A}\right] + 1$, since $\frac{1-B}{2A} \leq x \leq \frac{d^2-B}{2A}$ and $0 \geq \frac{1-B}{2A} > -1$ as $1 \leq B \leq 2A$ and so $0 \leq x \leq \frac{d^2-B}{2A}$.

Thus summing over all such equivalence classes we have the following proposition.

Proposition 6.2. *For each form f of discriminant d, we have*

$$\#\left\{(A,B) \sim f : 1 \leq B \leq d^2, 1 \leq A \leq q\right\} = \sum_{\substack{(A,B)\sim f \\ 1\leq B\leq 2A \\ 1\leq A\leq q}} \left(\left[\frac{d^2-B}{2A}\right] + 1\right).$$

Lemma 6.3. *Let (a, b_1, c_1) and (a, b_2, c_2) be two reduced forms of discriminant $d > 0$ with $a > 0$ and $b_1 \equiv b_2 \mod 2a$. Then $b_1 = b_2$.*

Proof. As $b_1 \equiv b_2 \mod 2a$ there is an integer x such that

$$b_2 = b_1 + 2ax.$$

As (a, b_1, c_1) and (a, b_2, c_2) are reduced, they satisfy

$$\begin{aligned} 0 <& b_1 < \sqrt{d}, \\ \sqrt{d} - b_1 <& 2a < \sqrt{d} + b_1. \end{aligned} \tag{6.2}$$

Likewise

$$\begin{aligned} 0 < b_1 + 2ax &< \sqrt{d}, \\ \sqrt{d} - (b_1 + 2ax) < 2a &< \sqrt{d} + b_1 + 2ax. \end{aligned} \tag{6.3}$$

As $\sqrt{d} - b_1 > 0$ from (6.2), we have from the second part of (6.3) that $-2ax < 2a$ and so $x \geq 0$. Also from the first part of (6.3) we have $x < \frac{\sqrt{d}-b_1}{2a}$ and as $\frac{\sqrt{d}-b_1}{2a} < 1$ from (6.2), we have that $x \leq 0$ and hence $x = 0$ and so $b_1 = b_2$. □

Lemma 6.4. *Let f be a form of discriminant d. Then*

$$\sum_{\substack{(a,b,c) \text{ is reduced} \\ (a,b,c)\sim f \\ a>0}} \frac{1}{a} \geq \begin{cases} \sqrt{\frac{3}{|d|}} & \text{for } d < 0, \\ \frac{R}{\sqrt{d}\log d} & \text{for } d > 0. \end{cases}$$

Proof. If $d < 0$, then there is a unique reduced form F with $F \sim f$. If $F = (a, b, c)$, then by an easy consequence of the definition of a reduced form, we have $a \leq \sqrt{\frac{|d|}{3}}$. Thus $\frac{1}{a} \geq \sqrt{\frac{3}{|d|}}$.

If $d > 0$, then there is a cycle of reduced forms equivalent to f. Let l be the length of the cycle. Then the number of forms (a, b, c) in a cycle with $a > 0$ is $\frac{l}{2}$ as the a values of the forms in a cycle alternate in sign. Now if (a, b, c) is reduced we have as a consequence of the definition of a reduced form, $|a| < \sqrt{d} \Rightarrow \frac{1}{|a|} > \frac{1}{\sqrt{d}}$ and thus

$$\sum_{\substack{(a,b,c) \text{ is reduced} \\ (a,b,c)\sim f \\ a>0}} \frac{1}{a} > \frac{l}{2\sqrt{d}} > \frac{R}{\sqrt{d}\log d}$$

as $l > \frac{2R}{\log d}$ ([8]). □

Lemma 6.5. *Let f be a form of discriminant d, and $q = \sqrt{(d^4-d)/4}$. The map $\phi : \ \{(a,b,c) \sim f : (a,b,c)$ is reduced and $a > 0\} \longrightarrow \mathcal{T} := \{(A,B) \sim f : 1 \leq A \leq \frac{q}{2}$ and $1 \leq B \leq 2A\}$, defined by $\phi(a,b,c) = (a,B)$ where B is the least positive residue of $b \mod 2a$, is an injection.*

Proof. If $d < 0$, then $\phi(a,b,c) = (a,b,c) \in \mathcal{T}$, and the result follows immediately.

Now let $d > 2$ and suppose that $(a,b,c) \sim f$ is reduced with $a > 0$. It is easily verified that $(a,B) \sim (a,b,c)$ and hence to f. As (a,b,c) is reduced, we have $a < \sqrt{d}$. As $\sqrt{d} < \frac{q}{2}$ so $a < \frac{q}{2}$ and hence $(a,B) \in \mathcal{T}$ as required.

Next we show that ϕ is indeed an injection: Suppose that $f_1 = (a_1,b_1,c_1)$ and $f_2 = (a_2,b_2,c_2)$ are reduced forms, both equivalent to f, with $a_1, a_2 > 0$ and $\phi(f_1) = \phi(f_2)$. Then $(a_1,B_1) = \phi(f_1) = \phi(f_2) = (a_2,B_2)$ and so $a_1 = a_2 = a$, say. But then $b_1 \equiv B_1 = B_2 \equiv b_2 \mod 2a$, and so $b_1 = b_2$ by Lemma 6.3. However since $a_1 = a_2$ and $b_1 = b_2$ we evidently have $c_1 = c_2$, and so $f_1 = f_2$, and thus ϕ is indeed an injection, and the lemma follows. □

Corollary 6.6. *Fix $\epsilon > 0$. With the hypothesis of Lemma 6.5, we have for $|d|$ sufficiently large*

$$\sum_{A=1}^{\frac{q}{2}} \frac{1}{A} \sum_{\substack{(A,B)\sim f \\ 1 \leq B < 2A}} 1 \geq \begin{cases} \sqrt{\frac{3}{|d|}} & \text{for } d < 0, \\ \frac{R}{\sqrt{d}\log d} & \text{for } d > 0. \end{cases}$$

Proof. By Lemma 6.5 we have

$$\sum_{A=1}^{\frac{q}{2}} \frac{1}{A} \sum_{\substack{(A,B)\sim f \\ 1 \leq B < 2A}} 1 \geq \sum_{\substack{(a,b) \text{ is reduced} \\ (a,b)\sim f \\ a>0}} \frac{1}{a}$$

and the result follows from Lemma 6.4. □

Theorem 6.7. *Fix $\epsilon > 0$. Let f be a form of discriminant d. The probability of choosing a form equivalent to f, using Algorithm 6.1, is greater than*

$$\begin{cases} \frac{1}{|d|^{1/2+\epsilon}} & \text{for } d < 0, \\ \frac{R}{|d|^{1/2+\epsilon}} & \text{for } d > 0, \end{cases}$$

if $|d|$ is sufficiently large.

Proof. From (6.1) and Proposition 6.2, we know that the probability of choosing a form equivalent to f using Algorithm 6.1 is

$$\geq \frac{1}{|d|^{2+\epsilon/2}} \sum_{\substack{(A,B)\sim f \\ 1 \leq B \leq 2A \\ 1 \leq A \leq q}} \left(\left[\frac{d^2 - B}{2A} \right] + 1 \right) \geq \frac{1}{|d|^{2+\epsilon/2}} \sum_{A=1}^{q} \sum_{\substack{(A,B)\sim f \\ 1 \leq B \leq 2A}} \frac{d^2 - B}{2A}$$

$$\geq \frac{1}{|d|^{2+\epsilon/2}} \sum_{A=1}^{q/2} \sum_{\substack{(A,B)\sim f \\ 1 \leq B \leq 2A}} \frac{d^2}{8A} = \frac{1}{8|d|^{\epsilon/2}} \sum_{A=1}^{q/2} \frac{1}{A} \sum_{\substack{(A,B)\sim f \\ 1 \leq B \leq 2A}} 1$$

since $B \leq 2A \leq q$ and the result follows from Corollary 6.6. □

The Probability of Choosing a Form Outside a Given Subgroup. In the algorithm for computing the class number, we need to choose a form F outside a given proper subgroup H. We now compute the probability that F lies outside H, where F is chosen using Algorithm 6.1.

Theorem 6.8 (Siegel [4]). *For every $\epsilon > 0$ there exists an ineffective constant $c_\epsilon > 0$ such that*

$$L(1,\chi) > c_\epsilon q^{-\epsilon}$$

where χ is a real primitive non-principal character mod q.

Theorem 6.9. *Fix $\epsilon > 0$ and let $|d|$ be sufficiently large. Let F be a form of discriminant d chosen using Algorithm 6.1. If H is any given proper subgroup of the class group G, then there is an ineffective positive constant c_ϵ such that*

$$\text{Prob}\,\{F \notin H\} > \frac{c_\epsilon}{|d|^\epsilon}.$$

Proof. We first observe that $|H| \leq \frac{h}{2}$ since it is a proper subgroup. Therefore there are at least $\frac{h}{2}$ classes in G that are not in H. Now

$$\text{Prob}\,\{F \notin H\} = \sum_{f \notin H} \text{Prob}\{F \sim f\}$$

where the sum is over a set of representative forms f from the equivalence classes of $G\backslash H$. By Theorem 6.7 and the comments just above, we have

$$\sum_{f \notin H} \text{Prob}\{F \sim f\} > \frac{h}{2}\begin{cases} \frac{1}{|d|^{1/2+\epsilon}} \\ \frac{R}{d^{1/2+\epsilon}} \end{cases} = \begin{cases} \frac{h}{2|d|^{1/2+2\epsilon}} & \text{for } d < 0, \\ \frac{hR}{2d^{1/2+2\epsilon}} & \text{for } d > 0 \end{cases}$$
$$\gg \frac{L(1,\chi)}{|d|^\epsilon} > \frac{c'_\epsilon}{|d|^{2\epsilon}},$$

where the last two inequalities follow from Dirichlet's Theorem (1.1), and Siegel's Theorem 6.8. □

The Main Theorem. *Fix $\epsilon > 0$ and let $d > 0$ be a fundamental discriminant. Then the class number h of the quadratic field $\mathbb{Q}(\sqrt{d})$ can be found, via our probabilistic algorithm, in expected time $O(d^{1/5+\epsilon})$.*

Proof. The algorithm consists of two steps. In the first step, an approximation $\tilde{h}$ for h is determined using Dirichlet's class number formula and the Random Summation technique. This is done by first approximating hR and then R. Thus by Theorem 2.9 and Theorem 4.2 the expected time taken is $O(d^{1/5+\epsilon})$.

Hence we obtain an interval $(L, L+\tilde{l})$ which contains the class number, with $\tilde{l} = O(\frac{d^{2/5+\epsilon}}{R})$.

In the second step, the precise value of h is found using Shanks' baby-steps-giant-steps technique. Here we pick forms using Algorithm 6.1 and compute the order of the subgroup generated by them until we find a subgroup with more than $\tilde{l}$ elements. By Theorem 5.2, the time taken to compute the order of a subgroup using baby steps giant steps is $O(d^{1/5+\epsilon})$.

Say we have a subgroup H whose order is less than $\tilde{l}$. If we pick a form F using Algorithm 6.1, then by Theorem 6.9, the probability that it does not belong to H is $> \frac{c_\epsilon}{d^\epsilon}$. Thus we will get at least one form outside H, with probability ~ 1, after we pick $O(d^\epsilon)$ forms using Algorithm 6.1.

Moreover, as h has at most $\log_2 h < \log_2(L + \tilde{l})$ prime factors, we need pick at most $O(\log(L + \tilde{l})d^\epsilon) = O(d^{2\epsilon})$ forms to get h.

Hence the total expected running time for finding h is $O(d^{1/5+\epsilon})$ after replacing ϵ by $\epsilon/2$ in the proof above. □

7. Discussion

In this concluding section we discuss the details of the algorithms presented and look at the running times of various computations involved.

Random Summation. One of the key tools used in our algorithm is that of 'Random Summation'. This is used to evaluate the sum $S = \sum_{\substack{n \le d^2 \\ \text{odd } n}} \frac{\left(\frac{d}{n}\right)}{n}$ where $\left(\frac{d}{n}\right)$ is the Jacobi symbol.

We could enhance the accuracy of the approximation for S by computing the exact sum up to $d^{1/5}$, and then only approximating the remaining terms of the sum S. Thus we wish now to approximate the sum

$$S_1 = \sum_{\substack{n=[d^{1/5}] \\ n \text{ odd}}}^{d^2} \frac{\left(\frac{d}{n}\right)}{n}.$$

We do this using Random Summation, wherein we consider the $M = [d^{1/5}]$ random variables Y_i, $1 \le i \le M$, where

$$\text{Prob}\{Y_i = n\} = \frac{\lambda}{n} \text{ with } 1 \le i \le M,\ n \text{ odd and } [d^{1/5}] \le n \le d^2,$$

and λ is defined by

$$\sum_{\substack{n=[d^{1/5}] \\ n \text{ odd}}}^{d^2} \frac{\lambda}{n} = 1.$$

We then let $X_i = \left(\frac{d}{Y_i}\right)$. The reason that this can be used to approximate S_1 is that the 'expected value' of each X_i is precisely λS_1.

We run into a practical difficulty: How do you choose an integer n in the given range, with probability exactly $\frac{\lambda}{n}$? As we do not know how to do this, we propose an algorithm that is practical and chooses n with probability close to, but not exactly, $\frac{\lambda}{n}$.

This practical algorithm will choose a random odd integer n in the range $[d^{1/5}] \le n \le d^2$. Let Y' denote this random selection. The algorithm chooses an integer n with probability close to, but not exactly equal to, the desired probability (i.e . $\frac{\lambda}{n}$). We then let $X' = \left(\frac{d}{Y'}\right)$. It can be shown that this new method of approximation works just about as well as the theoretical approximation obtained earlier.

Algorithm 7.1. Let $K = [d^{1/10}]$, and select δ to get $(1+\delta)^K = d^2/[d^{1/5}]$. For $1 \le k \le K$, we let I_k denote the interval $\left([d^{1/5}](1+\delta)^{k-1}, [d^{1/5}](1+\delta)^k\right]$, and let E_k be the number of odd integers in I_k.

Step 1. Choose an integer k uniformly from $[1, K]$, i.e. each integer is chosen with probability $\frac{1}{K}$.

Step 2. Choose an odd integer n uniformly from I_k, i.e. each integer is chosen with probability $\frac{1}{E_k}$.

Step 3. Let $Y' = n$ and let $X' = \left(\frac{d}{Y'}\right)$.

We now wish to determine the 'expected value' of the random variable X'. Now, if $n \in I_k$, then Prob $\{Y' = n\} = \frac{1}{KE_k}$; and so, the 'expected value' of X',

$$E(X') = \sum_{\substack{n=[d^{1/5}] \\ n \text{ odd}}}^{d^2} \left(\frac{d}{n}\right) \text{Prob}\{Y' = n\} = \sum_{k=1}^{K} \frac{1}{K} \sum_{\substack{n \in I_k \\ n \text{ odd}}} \frac{1}{E_k} \left(\frac{d}{n}\right). \tag{7.1}$$

Note that $\delta = \frac{9 \log d}{5d^{1/10}} + O\left(\frac{\log^2 d}{d^{1/5}}\right) = \frac{9 \log d}{5d^{1/10}}(1 + O(\delta))$ and $K = d^{1/10} + O(1) = d^{1/10}(1+O(\delta))$. Also $E_k = \frac{\delta}{2}[d^{1/5}](1+\delta)^{k-1}+O(1)$; and, since $\frac{\delta}{2}[d^{1/5}] \gg d^{1/10} \log d$, thus $E_k = \frac{\delta}{2}[d^{1/5}](1 + \delta)^{k-1}(1 + O(\delta))$. Now, if $n \in I_k$, then $n = [d^{1/5}](1 + \delta)^{k-1} \cdot (1 + O(\delta))$, so that $\frac{1}{E_k} = \frac{1}{n}\left(\frac{2}{\delta} + O(1)\right)$. Therefore, from (7.1),

$$\begin{aligned} E(X') &= \frac{1}{K}\left(\frac{2}{\delta} + O(1)\right) \sum_{k=1}^{K} \sum_{\substack{n \in I_k \\ n \text{ odd}}} \frac{1}{n}\left(\frac{d}{n}\right) \\ &= \frac{2}{K\delta} S_1(1 + O(\delta)) = \frac{10}{9 \log d} S_1(1 + O(\delta)). \end{aligned}$$

Finally, since $|S_1| = O(\log d)$, thus $S_1 = \frac{9 \log d}{10} E(X') + O\left(\frac{\log^2 d}{d^{1/10}}\right)$. In Proposition 2.5 we saw that we are prepared to allow the error $O(1/d^{1/10-\epsilon})$ when approximating S, and thus S_1; so we see that replacing the random variables X_i by X_i' will not significantly alter the power of algorithm, whilst rendering it practical.

Probability Analysis. The class number is computed exactly. However the algorithm is completed in an expected running time since the random summation technique is used and forms are chosen randomly.

We also remind the reader that our lower bound for the probability that a form, chosen using Algorithm 6.1, lies outside a given subgroup (Theorem 6.9) depends on Siegel's theorem (Theorem 6.8). Thus although the algorithm is practical, our estimate on its running time is ineffective as Siegel's constant is ineffective. However Tatuzawa ([16]) has provided an actual value for Siegel's constant which holds for all but at most one value of d.

The Imaginary Case. As there is no regulator in the case when $d < 0$, the algorithm is simpler. We use random summation to approximate $L(1, \chi)$ and then Dirichlet's formula to compute an approximation for h. Next we carry out the second part of Shanks' algorithm to find h exactly.

However, unlike the real case, here we do not have the means to verify if an interval provided by random summation is indeed correct. Hence although the algorithm may provide the wrong answer, it does so only with very low probability.

Acknowledements

This work is a result of the Ph.D. thesis completed under the guidance of Andrew Granville. I would like to thank Andrew for the numerous ideas and encouragement that he has given me, that have been instrumental in the writing of this paper. I would also like to thank Carl Pomerance for the many enlightening discussions on this subject.

References

1. R. P. Brent, *Fast Multiple-Precision Evaluation of Elementary Functions*, J. Assoc. Comp. Mach. **23** (1976), 242- 251. MR **52:**16111
2. D. Buell, *Binary Quadratic Forms*, Springer-Verlag, New York/Berlin/Heidelberg, 1989. MR **92b:**11021
3. H. Cohn, *Advanced Number Theory*, Dover, Inc. New York, 1980. MR **82b:**12001
4. H. Davenport, *Multiplicative Number Theory* (4th, ed.), Springer-Verlag, New York, 1980, 111-222. MR **82m:**10001

→ D. W. DeTemple, *A quicker Convergence to Euler's Constant*, Amer. Math. Monthly **100** (1993), 468-470. MR **94e:**11146

6. G. H. Hardy and E.M. Wright, *An introduction to the theory of numbers*, 5th ed, Oxford Science Publication. MR **81i:**10002
7. L. K. Hua, *Introduction to Number Theory*, Springer-Verlag, New York, 1982. MR **83f:**10001
8. H. W. Lenstra, Jr., *On the calculation of regulators and class numbers of quadratic fields*, Lond. Math. Soc. Lect. Note Ser, **56** (1982), 123-150. MR **86g:**11080
9. H. W. Lenstra, Jr. and C. Pomerance, *A rigorous time bound for factoring integers*, Journal of the Americain Mathematical Society **5,** (1992). MR **92m:**11145
10. R. A. Mollin and H. Williams, *Computation of the class number of real quadratic fields*, Utilitas Math., **41** (1992), 259-308. MR **93d:**11134
11. L. Sachs, *Applied Statistics, A handbook of techniques, 2nd ed., pp 64*, Springer-Verlag, New York/Berlin/Heidelberg/Tokyo, 1984. MR **85k:**62001
12. R. J. Schoof, *Quadratic fields and factorization, Computational methods in number theory*, (H. W. Lenstra, Jr., and R.Tijdeman, eds.), Math.Centrum, Number 155, part II, Amsterdam **1** (1983), 235-286. MR **85g:**11118b
13. J.Shallit, *On the worst case of three algorithms for computing the Jacobi symbol*, J. Symbolic Computation **10** (1990), 593-610. MR **91m:**11112
14. D.Shanks, *Class number, a theory of factorization, and genera*, Proc. Symp. Pure Math., Amer. Math. Soc. **20** (1971), 415-440. MR **47:**4932
15. D.Shanks, *The infrastructure of real quadratic fields and its application*, Proc.1972 Number Theory Conf.,Boulder, Colorado (1973), 217-224. MR **52:**10672
16. T. Tatuzawa, *On a theorem of Siegel*, Japan J. Math. **21** (1951), 163-178. MR **14:**452c

Department of Mathematics, University of Georgia, Athens, Georgia 30602

E-mail address: as@turing.upr.clu.edu

Current address: Department of Mathematics, University of Puerto Rico, CUH Station, 100 Carretera 908, Humacao, Puerto Rico 00791-4300