MATHEMATICS OF COMPUTATION
Volume 75, Number 255, July 2006, Pages 1481–1492
S 0025-5718(06)01850-3
Article electronically published on March 21, 2006

QUADRATIC CLASS NUMBERS AND CHARACTER SUMS

ANDREW R. BOOKER

Abstract. We present an algorithm for computing the class number of the quadratic number field of discriminant d. The algorithm terminates unconditionally with the correct answer and, under the GRH, executes in $O_\varepsilon(|d|^{1/4+\varepsilon})$ steps. The technique used combines algebraic methods with Burgess' theorem on character sums to estimate $L(1,\chi_d)$. We give an explicit version of Burgess' theorem valid for prime discriminants and, as an application, we compute the class number of a 32-digit discriminant.

1. Introduction

1.1. **A class number algorithm.** In [27, 17], Hafner and McCurley, expanding on ideas of Lenstra and Lenstra [22, 23], gave an algorithm for computing the class number of an imaginary quadratic field $\mathbb{Q}(\sqrt{d})$ that runs heuristically in "sub-exponential" time $O_\varepsilon(|d|^\varepsilon)$, assuming the Generalized Riemann Hypothesis for the associated quadratic character L-function, $L(s,\chi_d)$. Later, Buchmann generalized their work to all number fields [6], and gave, again under GRH, a deterministic algorithm for quadratic fields that runs in time $O_\varepsilon(|d|^{1/4+\varepsilon})$ [4, Prop. 9.7.16]. Other authors [5, 7, 11, 8, 20] have given variations of the sub-exponential algorithm that perform well in practice, and have enabled the computation of class numbers of discriminants with more than 100 digits. However, in all of these works the GRH is an essential ingredient, in the sense that the results are not known to be correct without it.

On the other hand, the fastest known unconditional algorithms execute in time $O_\varepsilon(|d|^{1/2+\varepsilon})$; that can be done either algebraically [4, Prop 9.7.15] or analytically, with the approximate functional equation [26]. In this paper we show how to combine Buchmann's algorithm for quadratic fields with analytic methods to give an unconditional algorithm that always runs in time $O_\varepsilon(|d|^{1/2+\varepsilon})$, and in time $O_\varepsilon(|d|^{1/4+\varepsilon})$ if GRH is true. For ease of presentation, we concentrate on the real quadratic case, $d > 0$; the imaginary case follows with simple modifications. For $d > 0$, Dirichlet's class number formula takes the form

$$h(d)R(d) = \frac{\sqrt{d}}{2} L(1,\chi_d), \tag{1}$$

where $h(d)$ is the class number and $R(d) = \log \epsilon_d$ is the regulator. The latter may be computed deterministically to high precision (within an absolute error of d^{-1}, say) in time $O_\varepsilon(d^{1/4+\varepsilon})$ [2], and heuristically in time $O_\varepsilon(d^{1/6+\varepsilon})$ [14].

Received by the editor November 26, 2004 and, in revised form, July 21, 2005.
2000 *Mathematics Subject Classification.* Primary 11Y35.
The author was supported by an NSF postdoctoral fellowship.

©2006 American Mathematical Society
Reverts to public domain 28 years from publication

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

(The $d^{1/6+\varepsilon}$ method, based on a fast heuristic algorithm and verification, complements our result in the real quadratic case.) Hence, to determine the class number it suffices to compute $L(1,\chi_d)$.

There are two key points to our method, which we outline here and defer complete proofs until Section 2. First, Buchmann's algorithm guarantees unconditionally that the computed number, say h_0, is a divisor of the true class number $h(d)$. Rewriting Dirichlet's formula as an expression for the integer $h(d)/h_0$,

$$\frac{h(d)}{h_0} = \frac{\sqrt{d}}{2h_0R(d)}L(1,\chi_d), \tag{2}$$

we see that to determine the class number it suffices to compute $L(1,\chi_d)$ to within an absolute error bounded by $\frac{h_0R(d)}{\sqrt{d}}$. Second, Burgess' theorem gives a method for computing $L(1,\chi_d)$ to that accuracy with a short character sum, as long as Buchmann's algorithm produces (as expected) the correct class number, or a substantially large factor of it.

Precisely, by the approximate functional equation, we have

$$1 \le \frac{h(d)}{h_0} = \frac{\sqrt{d}}{2h_0R(d)}L(1,\chi_d) = \frac{1}{h_0R(d)}\sum_{n=1}^{X}\chi_d(n)F\left(\frac{n}{\sqrt{d}}\right) + \frac{E_X(d)}{h_0R(d)}, \tag{3}$$

where $F(x)$ is a certain smooth function of rapid decay as $x \to \infty$, and

$$E_X(d) := \sum_{n=X+1}^{\infty}\chi_d(n)F\left(\frac{n}{\sqrt{d}}\right) \ll_{r,\varepsilon} X^{-\frac{1}{r}}d^{\frac{1}{2}+\frac{r+1}{4r^2}+\varepsilon} \tag{4}$$

for any positive integer r and any $\varepsilon > 0$, by Burgess' theorem. Now, if $h_0 = h(d)$, then $h_0R(d) \gg_\varepsilon d^{1/2-\varepsilon}$, by Siegel's theorem (with an ineffective implied constant; see [21] for a more precise statement). Hence,

$$\frac{E_X(d)}{h_0R(d)} \ll_{r,\varepsilon} X^{-\frac{1}{r}}d^{\frac{r+1}{4r^2}+2\varepsilon} \tag{5}$$

is small provided that $X \gg_{r,\varepsilon} d^{\frac{1}{4}+\frac{1}{4r}+2r\varepsilon}$, and the numerical computation of (3) must yield a value close to 1, hence less than 2. Conversely, if for some X, using (3) and a version of (4) with explicit constants, we obtain numerically that $h(d)/h_0 < 2$, then $h(d) = h_0$. Moreover, if $h_0R(d) \ge d^{7/16+\delta}$ for some $\delta > 0$, then using (4) with $r = 2$, we have

$$\frac{E_X(d)}{h_0R(d)} \ll_{r,\varepsilon} X^{-1/2}d^{1/4+\varepsilon-\delta}, \tag{6}$$

which is small for $X \gg_\varepsilon d^{1/2-2\delta+2\varepsilon}$; thus we can use (3) to compute $h(d)/h_0$ (and hence $h(d)$) in time $O(d^{1/2-\eta})$ for some $\eta = \eta(\delta) > 0$.

We may formulate the above certification technique as an algorithm, as follows. First, we use Buchmann's algorithm to compute the divisor h_0. If $h_0R(d) > d^{7/16}$, then we use the above strategy to compute $h(d)$ from h_0; otherwise we fall back on the standard $O_\varepsilon(d^{1/2+\varepsilon})$ algorithm. Note that if GRH is true, then Buchmann's algorithm always produces the correct class number, and moreover we then have $h_0R(d) \gg \frac{d^{1/2}}{\log\log d}$, with an effectively computable constant (see equation (13) below). Applying (4) with r arbitrarily large, we obtain the following proposition.

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

Proposition 1. *Let d be a fundamental discriminant. There is an algorithm that unconditionally computes the class number $h(d)$. It always executes in time $O_\varepsilon(|d|^{1/2+\varepsilon})$, and in time $O_\varepsilon(|d|^{1/4+\varepsilon})$ if GRH is true.*

1.2. **Comparison with other results.** As mentioned above, there are heuristic methods for computing class numbers in expected time $O_\varepsilon(|d|^\varepsilon)$. For small discriminants, the fastest algorithm in practice is that of Lenstra [24], based on Shanks' "baby step-giant step" technique [30, 31]. It typically computes the class number and group structure (under GRH) in about $|d|^{1/5}$ steps, although it can take longer if the class group is far from cyclic. (More explicitly, the algorithm is known to be exponential in the number of elementary divisors of the group [9]. However, the Cohen-Lenstra heuristics [12] predict that this is rarely large, so it is not a serious defect in practice. Note also that recent results on 3-torsion in class groups [18, 28] could be used to give an improved worst case bound.) Lenstra's group structure algorithm may be used with our work, in place of Buchmann's algorithm, to give a GRH-free certification.

We also mention a probabilistic algorithm, due to Srinivasan [32], for computing quadratic class numbers in expected time $O(|d|^{1/5+\varepsilon})$. Unfortunately (despite claims in [32]), it is not clear to us that one can in all cases certify the results of Srinivasan's algorithm.

1.3. **An application.** In order to implement our algorithm, we need a version of Burgess' theorem with explicit constants. One such bound, valid for prime discriminants, is provided by Grosswald [16, Thm. 1]. In Section 3 we derive a bound, again for prime discriminant, with substantially better constants than those of [16], based on the method Iwaniec and Kowalski [19, Sec. 12.4]. In Section 4 we detail the implementation of the algorithm using this bound, and apply it to a 32-digit discriminant.

2. Proof of Proposition 1

2.1. **The approximate functional equation.** A fast method for evaluating the L-function at 1 (or any point, for that matter) is the approximate functional equation. We use a version due to Cohen [13, Sec. 5.6.2]:

$$\frac{\sqrt{d}}{2} L(1, \chi_d) = \sum_{n=1}^{\infty} \chi_d(n) F\left(\frac{n}{\sqrt{d}}\right), \tag{7}$$

where

$$F(x) = \int_x^\infty \left(\frac{1}{x} + \frac{1}{t}\right) e^{-\pi t^2}\, dt. \tag{8}$$

(N.B.: The choice of smoothing function F is not canonical. This particular F has the nice features of monotonicity and convexity.)

Lemma. *Suppose $\alpha > -1$ and $x > 0$. Then*

$$\int_x^\infty t^{-\alpha} e^{-\pi t^2}\, dt < \frac{e^{-\pi x^2}}{2\pi x^{\alpha+1}}. \tag{9}$$

Proof. Integration by parts. □

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

As mentioned above, the approximate functional equation yields an algorithm to compute the class number in time roughly $\sqrt{d}$. More precisely, suppose that we compute the sum up to the first X terms. For the remaining terms we have

$$\left| \sum_{n=X+1}^{\infty} \chi_d(n) F\left(\frac{n}{\sqrt{d}}\right) \right| \leq \int_X^{\infty} F\left(\frac{t}{\sqrt{d}}\right) dt = \sqrt{d} \int_{X/\sqrt{d}}^{\infty} F(x)\, dx. \tag{10}$$

The lemma shows that $F(x) < \frac{1}{\pi x^2} e^{-\pi x^2}$. Hence, applying the lemma once more, we have

$$\sqrt{d} \int_{X/\sqrt{d}}^{\infty} F(x)\, dx \leq \frac{\sqrt{d}}{\pi} \int_{X/\sqrt{d}}^{\infty} x^{-2} e^{-\pi x^2}\, dx < \frac{d^2 e^{-\pi X^2/d}}{2\pi^2 X^3}. \tag{11}$$

If we take $X \geq \sqrt{\frac{d}{2\pi} \log \frac{d}{2\pi}}$ (assuming d is sufficiently large for this to be defined), we see that the last line is at most $2(\log(d/2\pi))^{-3/2}$. Dividing by $R(d)$, the result is $< \frac{1}{2}$ for $d \geq 33$, and hence determines the class number uniquely.

2.2. **Verification of a faster algorithm.** Now suppose, as in the Introduction, that we have computed a number h_0 which divides $h(d)$ unconditionally, and that on GRH they are equal. By the class number formula, we then have $h_0 R(d) = \frac{\sqrt{d}}{2} L(1, \chi_d)$. Dividing this into the error term (11), we see that the error in estimating $h(d)/h_0$ is at most

$$\frac{(\pi X^2/d)^{-3/2} e^{-\pi X^2/d}}{\sqrt{\pi} L(1, \chi_d)}. \tag{12}$$

Assuming GRH again, we have the bound of Littlewood [25]:

$$L(1, \chi_d) > (1 + o(1)) \frac{\pi^2}{12 e^{\gamma}} (\log \log d)^{-1}. \tag{13}$$

Thus (12) is at most

$$(1 + o(1)) \frac{12 e^{\gamma}}{\pi^{5/2}} \left(\frac{\pi X^2}{d}\right)^{-3/2} e^{-\pi X^2/d} \log \log d. \tag{14}$$

From (14) we see that to compute the class number by this method can require up to about $\sqrt{\frac{d}{\pi} \log \log \log d}$ terms, a modest improvement over the approximate functional equation alone. (It does give a practical improvement for small discriminants; in joint work with A. Strömbergsson [3], we combined this method with Lenstra's algorithm to compute $h(t^2 \pm 4)$ for all $t < e^{18} \approx 66 \times 10^6$.) However, note that unlike (11), the bound (12) would decay near $X \approx \sqrt{d}$ even without the help of the exponential factor. That, in turn, makes it possible to get a better result by nontrivially estimating the sum in (10). This is related to bounds for the L-function in the critical strip, in the sense that any sub-convexity bound gives an improvement over the $O_\varepsilon(d^{1/2+\varepsilon})$ algorithms above, while an effective Lindelöf hypothesis would give $O_\varepsilon(d^\varepsilon)$. Below we present an argument using Burgess' bounds for short character sums.

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

2.3. **Burgess' bounds.** Let $S(n) = \sum_{X<j\le n} \chi_d(j)$. Applying partial summation to the error term in (10), we have

$$\sum_{n=X+1}^{\infty} \chi_d(n) F\Big(\frac{n}{\sqrt{d}}\Big) = \sum_{n=X+1}^{\infty} S(n) \left[F\Big(\frac{n}{\sqrt{d}}\Big) - F\Big(\frac{n+1}{\sqrt{d}}\Big)\right]. \tag{15}$$

Now, Burgess' theorem [10] gives, for any positive integer r,

$$S(n) = O_{r,\varepsilon}\Big(n^{1-\frac{1}{r}} d^{\frac{r+1}{4r^2}+\varepsilon}\Big). \tag{16}$$

We substitute this into the above and use the trivial estimate $F'(x) \ll x^{-2}$ to get the upper bound

$$d^{\frac{1}{2}+\frac{r+1}{4r^2}+\varepsilon} \sum_{n>X} \frac{1}{n^{1+1/r}} \ll_r X^{-\frac{1}{r}} d^{\frac{1}{2}+\frac{r+1}{4r^2}+\varepsilon}. \tag{17}$$

Now again we divide this by $h_0 R(d) \gg_\varepsilon d^{1/2-\varepsilon}$ to get $O_{r,\varepsilon}\big(X^{-\frac{1}{r}} d^{\frac{r+1}{4r^2}+2\varepsilon}\big)$. This is small provided that $X \gg_{r,\varepsilon} d^{\frac{1}{4}+\frac{1}{4r}+2r\varepsilon}$. Finally, note that each term of (7) for $n \le X$ may be computed to high precision in time $O\big((\log d)^{O(1)}\big)$. Since r and ε are arbitrary, this gives an algorithm running in time $O_{\varepsilon'}\big(d^{1/4+\varepsilon'}\big)$, as claimed.

3. An explicit bound for character sums

In this section, we derive the following explicit version of Burgess' theorem.

Proposition 2. *Let $d > 10^{20}$ be a prime number $\equiv 1 \pmod 4$, $r \in \{2, \dots, 15\}$, and M, N integers with $0 < M, N \le 2\sqrt{d}$. Then*

$$\left|\sum_{M\le n<M+N} \chi_d(n)\right| \le \alpha(r) d^{\frac{r+1}{4r^2}} \big(\log d + \beta(r)\big)^{\frac{1}{2r}} N^{1-\frac{1}{r}}, \tag{18}$$

where $\alpha(r), \beta(r)$ are given by Table 1.

Table 1.

r	$\alpha(r)$	$\beta(r)$	r	$\alpha(r)$	$\beta(r)$
2	1.8221	8.9077	9	1.4548	0.0085
3	1.8000	5.3948	10	1.4231	−0.4106
4	1.7263	3.6658	11	1.3958	−0.7848
5	1.6526	2.5405	12	1.3721	−1.1232
6	1.5892	1.7059	13	1.3512	−1.4323
7	1.5363	1.0405	14	1.3328	−1.7169
8	1.4921	0.4856	15	1.3164	−1.9808

Remark. The restriction on N was chosen to suit our application; a similar bound could be obtained for all N at the expense of slightly worse constants. For N of size $\sqrt{d}$, the constant $\alpha(r)$ is essentially optimal for this method of proof.

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

We will prove (18) by induction on N. During the induction we may allow M to vary outside of the given range; in particular, it may be negative. Assume for now that we have a bound of the form

$$\left|\sum_{M\le n<M+N'}\chi_d(n)\right|\le K(N')^{1-1/r}, \tag{19}$$

valid for all $N'<N$. Note that for $N'\le K^r$ this inequality is trivial. Let $A,B>0$ be parameters, with B an integer. We consider shifts of the sum (18) by ab, where a and b are integers, $|b|\le B$, $|a|\le A$ and $|a|$ is prime. By the induction hypothesis, we have

$$\left|\sum_{M\le n<M+N}\chi_d(n)-\sum_{M\le n<M+N}\chi_d(n+ab)\right|\le 2K|ab|^{1-1/r}. \tag{20}$$

Averaging over a,b we get

$$\begin{aligned}&\left|\sum_{M\le n<M+N}\chi_d(n)-\frac{1}{2\pi(A)(2B+1)}\sum_{a,b}\sum_{M\le n<M+N}\chi_d(n+ab)\right|\\ &\quad\le\frac{2K}{\pi(A)(B+1/2)}\sum_{a\text{ prime}\le A}a^{1-1/r}\sum_{1\le b\le B}b^{1-1/r}\\ &\quad\le\frac{2K}{(2-1/r)^2}(A(B+1/2))^{1-1/r}\frac{\sum_{a\text{ prime}\le A}a^{1-1/r}}{\pi(A)A^{1-1/r}/(2-1/r)}.\end{aligned} \tag{21}$$

Turning now to the average, in the inner sum we write

$$\chi_d(n+ab)=\chi_d(a)\chi_d(\bar{a}n+b), \tag{22}$$

where $\bar{a}$ is a multiplicative inverse of $a\pmod d$; hence the average is bounded by

$$\begin{aligned}&\frac{1}{2\pi(A)(2B+1)}\sum_a\sum_{M\le n<M+N}\left|\sum_b\chi_d(\bar{a}n+b)\right|\\ &\quad=\frac{1}{2\pi(A)(2B+1)}\sum_{x\,(\mathrm{mod}\ d)}\nu(x)\left|\sum_b\chi_d(x+b)\right|,\end{aligned} \tag{23}$$

where $\nu(x)$ is the number of representations of x as $\bar{a}n$.

To bound the sum, we apply Hölder's inequality in the form

$$\left[\sum_{x\,(\mathrm{mod}\ d)}\nu(x)\right]^{1-1/r}\left[\sum_{x\,(\mathrm{mod}\ d)}\nu(x)^2\right]^{1/2r}\left[\sum_{x\,(\mathrm{mod}\ d)}\left|\sum_b\chi_d(x+b)\right|^{2r}\right]^{1/2r}. \tag{24}$$

Now the first sum is simply $2\pi(A)N$. The second is the number of quadruples (a_1,n_1,a_2,n_2) with $a_1n_2\equiv a_2n_1\pmod d$. The size of the numbers will insure (to be checked later) that this implies

$$a_1n_2=a_2n_1. \tag{25}$$

If $a_1=a_2$, then $n_1=n_2$; these terms give the diagonal contribution $2\pi(A)N$. For the remaining terms, for fixed $a_1\ne a_2$, the number of solutions to (25) is at most

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

$\frac{N}{\max(|a_1|,|a_2|)}+1$. Thus, these terms contribute at most

$$8\sum_{a_1 \text{ prime}\le A}\ \sum_{a_2 \text{ prime}<a_1}\left(\frac{N}{a_1}+1\right) < 4\pi(A)^2 + 8N\sum_{a \text{ prime}\le A}\frac{\pi(a)-1}{a}. \tag{26}$$

Altogether, we have that the second sum is bounded by

$$2\pi(A)N\left(1+\frac{2\pi(A)}{N}+\frac{4}{\pi(A)}\sum_{a \text{ prime}\le A}\frac{\pi(a)-1}{a}\right). \tag{27}$$

It will be the case that A is large, but small with respect to N, so that the factor in parentheses above will not be too much larger than 1. For now, we write simply f for this factor.

The third sum we expand directly:

$$\sum_{|b_1|,\dots,|b_{2r}|\le B}\ \sum_{x \pmod d}\chi_d((x+b_1)\cdots(x+b_{2r})). \tag{28}$$

Note that the inner sum is related to the number of points over $\mathbb{F}_d$ of the curve

$$y^2=(x+b_1)\cdots(x+b_{2r}). \tag{29}$$

If the right-hand side of (29) is not a square in $\mathbb{F}_d[x]$, then by Weil's bound for curves, the inner sum is at most $2(r-1)\sqrt{d}$. Otherwise, we can do little better than the trivial bound d. The number of ways that the right-hand side can be a square is $(2r-1)!!\cdot(2B+1)^r$, so we see that (28) is less than

$$2(r-1)\sqrt{d}(2B+1)^{2r}+(2r-1)!!\cdot d(2B+1)^r. \tag{30}$$

Putting everything together, we have the upper bound

$$\frac{(2\pi(A)N)^{1-1/2r}f^{1/2r}\big(2(r-1)\sqrt{d}(2B+1)^{2r}+(2r-1)!!\cdot d(2B+1)^r\big)^{1/2r}}{2\pi(A)(2B+1)}. \tag{31}$$

Now set $A(B+1/2)=cN$, for some c to be determined later, so that

$$2\pi(A)=\frac{\pi(A)}{A}\frac{4cN}{2B+1}. \tag{32}$$

This yields

$$\left(\frac{Af}{4c\pi(A)}\right)^{1/2r}N^{1-1/r}\big(2(r-1)\sqrt{d}(2B+1)+(2r-1)!!\cdot d(2B+1)^{1-r}\big)^{1/2r}. \tag{33}$$

We choose $2B+1$ optimally to be the smallest odd integer $\ge\left[\frac{1}{2}(2r-1)!!\right]^{1/r}d^{1/2r}$ so that

$$2(r-1)\sqrt{d}(2B+1)+(2r-1)!!\cdot d(2B+1)^{1-r}\approx 4(r-1)\left[\frac{1}{2}(2r-1)!!\right]^{1/r}d^{1/2+1/2r}. \tag{34}$$

By Taylor's theorem, this can be off by no more than

$$4r(r-1)\left[\frac{1}{2}(2r-1)!!\right]^{-1/r}d^{1/2-1/2r}. \tag{35}$$

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

On the other hand, we were a bit wasteful in the bound (30); in particular, we get a savings of at least $2\big(\sqrt{d}-1\big)$ in Weil's bound each time that the right-hand side of (29) is divisible by a square. A conservative estimate for the total savings is

$$\begin{aligned} &2\big(\sqrt{d}-1\big)\left(2B+1-\frac{(2B+1)!}{(2B+1-2r)!}(2B+1)^{1-2r}\right) \\ &\qquad\geq 2\big(\sqrt{d}-1\big)\left[1-\exp\left(-\frac{r(r-1)}{\big[\frac{1}{2}(2r-1)!!\big]^{1/r}d^{1/2r}+2}\right)\right]. \end{aligned} \tag{36}$$

Comparing (35) and (36), one may see explicitly that for r not too large, this overcompensates for the error in the approximation (34). Thus, we get

$$\left(\frac{(r-1)[\frac{1}{2}(2r-1)!!]^{1/r}Af}{c\pi(A)}\right)^{1/2r}N^{1-1/r}d^{\frac{r+1}{4r^2}}. \tag{37}$$

Combining this with the terms from (21), we have

$$\left(\frac{(r-1)[\frac{1}{2}(2r-1)!!]^{1/r}Af}{c\pi(A)}\right)^{1/2r}N^{1-1/r}d^{\frac{r+1}{4r^2}}+\frac{2K}{(2-1/r)^2}(cN)^{1-1/r}g, \tag{38}$$

where we write

$$g=\frac{\sum_{a\text{ prime}\leq A}a^{1-1/r}}{\pi(A)A^{1-1/r}/(2-1/r)}. \tag{39}$$

To continue the induction, we want (38) to be at most $KN^{1-1/r}$, i.e.,

$$\frac{K}{d^{\frac{r+1}{4r^2}}}\geq\left(\frac{(r-1)[\frac{1}{2}(2r-1)!!]^{1/r}Af}{\pi(A)}\right)^{1/2r}\frac{c^{-1/2r}}{1-\frac{2}{(2-1/r)^2}c^{1-1/r}g}. \tag{40}$$

Note that $g\sim 1$ as $A\to\infty$. Thus, the optimal value of c is $\big(\frac{2r-1}{2r^2}\big)^{r/(r-1)}$. This gives

$$\frac{K}{d^{\frac{r+1}{4r^2}}}\geq\alpha(r)\left(\frac{2r}{r-1}\right)^{1/2r}\left(\frac{Af}{\pi(A)}\right)^{1/2r}\frac{2r-2}{2r-1-g}, \tag{41}$$

where

$$\alpha(r)=\left(\frac{(r-1)^2}{2r}\right)^{1/2r}\left[\frac{1}{2}(2r-1)!!\right]^{1/2r^2}\left(\frac{2r^2}{2r-1}\right)^{1/(2r-2)}\frac{2r-1}{2r-2}. \tag{42}$$

We also check that c is sufficiently small so that the shifts ab are strictly smaller than N, and that the assumption (25) is valid.

Next, recalling the definition of f, we have

$$\begin{aligned} \frac{Af}{\pi(A)}&=\frac{2A}{N}+\frac{A}{\pi(A)}\left(1+\frac{4}{\pi(A)}\sum_{a\text{ prime}\leq A}\frac{\pi(a)-1}{a}\right) \\ &=\frac{2A}{N}+\log A+3+O\left(\frac{1}{\log A}\right), \end{aligned} \tag{43}$$

as $A\to\infty$. Similarly, for g we have the asymptotic

$$g=1-\frac{\frac{r-1}{2r-1}}{\log A}+O\left(\frac{1}{\log^2 A}\right). \tag{44}$$

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

Further, A is bounded by

$$A \le \frac{2\left(\frac{2r-1}{2r^2}\right)^{r/(r-1)} N}{\left[\frac{1}{2}(2r-1)!!\right]^{1/r} d^{1/2r}} \le \frac{4\left(\frac{2r-1}{2r^2}\right)^{r/(r-1)}}{\left[\frac{1}{2}(2r-1)!!\right]^{1/r}} d^{1/2-1/2r}. \tag{45}$$

Thus,

$$\frac{2A}{N} + \log A \le \left(\frac{1}{2} - \frac{1}{2r}\right)\log d + \log\left(\frac{4\left(\frac{2r-1}{2r^2}\right)^{r/(r-1)}}{\left[\frac{1}{2}(2r-1)!!\right]^{1/r}}\right) + \frac{2A}{N}. \tag{46}$$

Also, for $d > 10^{20}$ we see that $\frac{2A}{N} \le \frac{1}{200}$.

Combining these estimates, it suffices to take

$$\frac{K}{d^{\frac{r+1}{4r^2}}} \ge \alpha(r)\left\{\left[\log d + \beta(r) + O\left(\frac{1}{\log A}\right)\right]\left[1 - \frac{\frac{r}{2r-1}}{\log A} + O\left(\frac{1}{\log^2 A}\right)\right]\right\}^{\frac{1}{2r}}, \tag{47}$$

where

$$\beta(r) = \frac{2r}{r-1}\left(\log\left(\frac{4\left(\frac{2r-1}{2r^2}\right)^{r/(r-1)}}{\left[\frac{1}{2}(2r-1)!!\right]^{1/r}}\right) + 3.005\right). \tag{48}$$

Finally, thanks to the coefficient $-\frac{r}{2r-1}$ above, we see that for d large, (47) is bounded by its limiting value as $A \to \infty$. To see that $d > 10^{20}$ is sufficient we rely on numerical computation for small A and explicit error terms for large A. In particular, we use the following estimates from [15] (see also [29] for general results of this type):

$$\begin{aligned} &\frac{x}{\log x - 1} < \pi(x) < \frac{x}{\log x - 1.1} \text{ for } x \ge 60184 \quad \text{and} \\ &|\theta(x) - x| < \frac{0.2x}{\log^2 x} \text{ for } x \ge 3594641. \end{aligned} \tag{49}$$

We omit the technical details.

4. Implementation

In this section we describe, by way of example, the implementation of our algorithm for prime discriminants, using Proposition 2. We consider the prime number $d = 10^{31} + 33$. First, we use PARI/GP [34] to compute the class number $h_0 = 43$ and regulator $R(d) \approx 84328477135202.25641$. Since PARI uses a variant of the subexponential algorithm, a priori these values come with no certificate of correctness without GRH. However, it is easy to verify that the supplied generator has order 43, so h_0 at least divides the true class number. Likewise, we may check the regulator by reduction theory; see, e.g., the PARI tutorial [33, Sec. 10]. These computations take only a few minutes to complete.

Now let $S(n)$ be as in Section 2.3. Since F is monotonically decreasing and convex, combining equations (2), (7) and (15), we have

$$\begin{aligned} \frac{h(d)}{h_0} \le \frac{1}{h_0 R(d)}\Bigg(&\sum_{n \le X} \chi_d(n) F\left(\frac{n}{\sqrt{d}}\right) + \frac{1}{\sqrt{d}} \sum_{X < n < 2\sqrt{d}} \left|S(n) F'\left(\frac{n}{\sqrt{d}}\right)\right| \\ &+ \left|S\left(2\sqrt{d}\right)\right| F(2) + \left|\sum_{n > 2\sqrt{d}} \chi_d(n) F\left(\frac{n}{\sqrt{d}}\right)\right|\Bigg). \end{aligned} \tag{50}$$

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

To handle the terms for $n > 2\sqrt{d}$ we apply the trivial bound (11) with $X = \lfloor 2\sqrt{d} \rfloor$. The error term (14) shows that for d of practical size these terms in general will be negligible, although that must still be checked; in our case we get $\frac{1}{h_0 R(d)} \frac{\sqrt{d}}{16\pi^2 e^{4\pi}} < 2 \times 10^{-8}$. Similarly, to handle the $\left|S\left(2\sqrt{d}\right)\right| F(2)$ term we may apply either (11) or Proposition 2 with $r = 2$.

For $n < 2\sqrt{d}$, Proposition 1 says that

$$|S(n)| \le \alpha(r) d^{\frac{r+1}{4r^2}} (\log d + \beta(r))^{\frac{1}{2r}} (n - X)^{1 - \frac{1}{r}}. \tag{51}$$

Since we do not yet know X, we start with a candidate value, say $X = \lfloor \sqrt{d} \rfloor$. To estimate the terms for $X < n < 2\sqrt{d}$, we compute in advance several linear upper bound approximations to $|F'(x)|$. Again by convexity of F, it suffices to compute sample points for, say, 10000 values of x between 0 and 2, and interpolate between them. We use geometrically spaced sample points, so that there are many more samples near 0; there the function is hardest to model because of the x^{-2} singularity and, consequently, those terms matter the most. We then divide the sum up into intervals between the sample points, starting from $2\sqrt{d}$ and working towards 0. Over any given interval, the sum is easily estimated by an integral.

Of course we want to use the best value of r for each interval. What typically happens is that the $r = 2$ bound is best for n near $2\sqrt{d}$. Then, as n decreases, the successive bounds for $r = 3, 4, \ldots$ become better. This continues until we reach a value of n for which the trivial bound is best.

After computing the sum, we see whether the total, including $\left|S\left(2\sqrt{d}\right)\right| F(2)$ and the terms from $n > 2\sqrt{d}$, exceeds $0.995 h_0 R(d)$, say. If it does, our chosen value of X is too small, otherwise too large. Repeating this procedure, we can quickly locate, by a bisection algorithm, the smallest X which yields an answer $\le 0.995 h_0 R(d)$. Applying this to our example, we find that $X = 750 \times 10^9$ terms are sufficient. For purposes of comparison, that is over 14000 times smaller than the $\sqrt{\frac{d}{2\pi} \log \frac{d}{2\pi}}$ terms needed for the approximate functional equation alone.

Finally we compute the first sum in (50). The error in our approximation is typically much less than the rigorous bounds indicate, so we are likely to get an answer less than $1.005 h_0 R(d)$. When this happens, we know for sure that $h(d)/h_0 < 2$ and hence $h(d) = h_0$. If the answer exceeds $1.005 h_0 R(d)$, then we replace 0.995 by a smaller value in the above and try again.

The computation of the sum up to X remains, by far, the most time consuming part of our algorithm, so it is worth some effort to optimize. Note that since the argument to the function F is small, we may replace F by the first few terms of its power series

$$F(x) = \frac{1}{2x} - \log x - \frac{2 + \gamma + \log \pi}{2} + \sum_{k=1}^{\infty} \frac{(-1)^{k-1} \pi^k}{k!} x^{2k} \left(\frac{1}{2k} + \frac{1}{2k+1} \right), \tag{52}$$

with only a small error. In fact, for our example we see by a trivial bound that the terms after $1/2x$ contribute less than $0.002 h_0 R(d)$. Thus, it is enough to compute

$$\frac{\sqrt{d}}{2} \sum_{n \le X} \frac{\chi_d(n)}{n}. \tag{53}$$

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

To do this efficiently, we split the sum over smooth and nonsmooth numbers:

$$\sum_{n\le X}\frac{\chi_d(n)}{n}=\sum_{\substack{n\le X\\ p|n\Rightarrow p\le\sqrt{X}}}\frac{\chi_d(n)}{n}+\sum_{\sqrt{X}<p\le X}\frac{\chi_d(p)}{p}\sum_{k\le X/p}\frac{\chi_d(k)}{k}. \tag{54}$$

Since $\chi_d(n)/n$ is multiplicative, the first sum above, over smooth numbers, may be evaluated recursively. For the second, it suffices to tabulate the partial sums $\sum_{k\le x}\chi_d(k)/k$ for $x<\sqrt{X}$. In this way, the relatively expensive evaluation of the character χ_d need only be carried out at prime numbers. As a result, the total number of arithmetic operations is $O(X)$, which gives a significant practical savings over the $O(X\log X)$ operations needed using in-order summation. (Note that the primes up to X may be computed in time $o(X)$, see, e.g., [1].)

Computing (53) for our example, we get less than $1.000001h_0R(d)$, and hence conclude that indeed $h(d)=43$. This required approximately 95 hours of computation on a 500MHz UltraSparc II.

Acknowledgments

This paper grew out of a comment by Andrew Granville; I thank him for his encouragement. Thanks also to the referee, Kannan Soundararajan, and Hugh Williams for helpful suggestions.

References

[1] A. O. L. Atkin and D. J. Bernstein. Prime sieves using binary quadratic forms. *Math. Comp.*, 73(246):1023–1030 (electronic), 2004. MR2031423 (2004i:11147)

[2] Ingrid Biehl and Johannes Buchmann. Algorithms for quadratic orders. In *Mathematics of Computation 1943–1993: a half-century of computational mathematics (Vancouver, BC, 1993)*, volume 48 of *Proc. Sympos. Appl. Math.*, pages 425–449. Amer. Math. Soc., Providence, RI, 1994. MR1314882 (95m:11146)

[3] Andrew R. Booker and Andreas Strömbergsson. Numerical computations with the trace formula and the Selberg eigenvalue conjecture, *Crelle* (to appear).

[4] J. Buchmann. *Algorithms for Binary Quadratic Forms*. Springer-Verlag, to appear.

[5] J. Buchmann and S. Düllmann. A probabilistic class group and regulator algorithm and its implementation. In *Computational number theory (Debrecen, 1989)*, pages 53–72. de Gruyter, Berlin, 1991. MR1151855 (92m:11150)

[6] Johannes Buchmann. A subexponential algorithm for the determination of class groups and regulators of algebraic number fields. In *Séminaire de Théorie des Nombres, Paris 1988–1989*, volume 91 of *Progr. Math.*, pages 27–41. Birkhäuser Boston, Boston, MA, 1990. MR1104698 (92g:11125)

[7] Johannes Buchmann and Stephan Düllmann. Distributed class group computation. In *Informatik*, volume 1 of *Teubner-Texte Inform.*, pages 69–79. Teubner, Stuttgart, 1992. MR1182565 (93e:11153)

[8] Johannes Buchmann, Michael J. Jacobson, Jr., Stefan Neis, Patrick Theobald, and Damian Weber. Sieving methods for class group computation. In *Algorithmic algebra and number theory (Heidelberg, 1997)*, pages 3–10. Springer, Berlin, 1999. MR1672089 (2000a:11177)

[9] Johannes Buchmann, Michael J. Jacobson, Jr., and Edlyn Teske. On some computational problems in finite abelian groups. *Math. Comp.*, 66(220):1663–1687, 1997. MR1432126 (98a:11185)

[10] D. A. Burgess. The distribution of quadratic residues and non-residues. *Mathematika*, 4:106–112, 1957. MR0093504 (20:28)

[11] H. Cohen, F. Diaz y Diaz, and M. Olivier. Calculs de nombres de classes et de régulateurs de corps quadratiques en temps sous-exponentiel. In *Séminaire de Théorie des Nombres, Paris, 1990–91*, volume 108 of *Progr. Math.*, pages 35–46. Birkhäuser Boston, Boston, MA, 1993. MR1263522 (94m:11151)

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use

[12] H. Cohen and H. W. Lenstra, Jr. Heuristics on class groups of number fields. In *Number theory, Noordwijkerhout 1983 (Noordwijkerhout, 1983)*, volume 1068 of *Lecture Notes in Math.*, pages 33–62. Springer, Berlin, 1984. MR0756082 (85j:11144)

[13] Henri Cohen. *A course in computational algebraic number theory*, volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 1993. MR1228206 (94i:11105)

[14] R. De Haan, M. J. Jacobson, Jr., and H. C. Williams. A fast, rigorous technique for verifying the regulator of a real quadratic field. *preprint*, 2004.

[15] P. Dusart. Autour de la fonction qui compte le nombre de nombres premiers. *Université de Limoges*, Ph.D. thesis, 1998.

[16] E. Grosswald. On Burgess' bound for primitive roots modulo primes and an application to $\Gamma(p)$. *Amer. J. Math.*, 103(6):1171–1183, 1981. MR0636957 (82k:10059)

[17] James L. Hafner and Kevin S. McCurley. A rigorous subexponential algorithm for computation of class groups. *J. Amer. Math. Soc.*, 2(4):837–850, 1989. MR1002631 (91f:11090)

[18] H. A. Helfgott and A. Venkatesh. Integral points on elliptic curves and 3-torsion in class groups. *preprint*, 2004.

[19] Henryk Iwaniec and Emmanuel Kowalski. *Analytic number theory*, volume 53 of *American Mathematical Society Colloquium Publications*. American Mathematical Society, Providence, RI, 2004. MR2061214 (2005h:11005)

[20] Michael J. Jacobson, Jr., Renate Scheidler, and Hugh C. Williams. The efficiency and security of a real quadratic field based key exchange protocol. In *Public-key cryptography and computational number theory (Warsaw, 2000)*, pages 89–112. de Gruyter, Berlin, 2001. MR1881630 (2003f:94062)

[21] Chun-Gang Ji and Hong-Wen Lu. Lower bound of real primitive L-function at $s = 1$. *Acta Arith.*, 111(4):405–409, 2004. MR2039505 (2004k:11140)

[22] A. K. Lenstra and H. W. Lenstra, Jr. Algorithms in number theory. *Tech. Report 97-008*, 1987.

[23] A. K. Lenstra and H. W. Lenstra, Jr. Algorithms in number theory. In *Handbook of theoretical computer science, Vol. A*, pages 673–715. Elsevier, Amsterdam, 1990. MR1127178

[24] H. W. Lenstra, Jr. On the calculation of regulators and class numbers of quadratic fields. In *Number theory days, 1980 (Exeter, 1980)*, volume 56 of *London Math. Soc. Lecture Note Ser.*, pages 123–150. Cambridge Univ. Press, Cambridge, 1982. MR0697260 (86g:11080)

[25] J.E. Littlewood. On the class number of the corpus $p(\sqrt{-k})$. *Proc. London Math. Soc.*, 27:358–372, 1928.

[26] Stéphane Louboutin. Computation of class numbers of quadratic number fields. *Math. Comp.*, 71(240):1735–1743 (electronic), 2002. MR1933052 (2003i:11163)

[27] K. S. McCurley. Cryptographic key distribution and computation in class groups. In *Proceedings NATO ASI on Number Theory and Applications (Dordrecht)*, volume 265 of *ASI Series C*, pages 459–479. Kluwer, 1989. MR1123090 (92e:11149)

[28] L. B. Pierce. The 3-part of class numbers of quadratic fields. *Oxford University*, MSc. thesis, 2004.

[29] J. Barkley Rosser and Lowell Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois J. Math.*, 6:64–94, 1962. MR0137689 (25:1139)

[30] Daniel Shanks. Class number, a theory of factorization, and genera. In *1969 Number Theory Institute (Proc. Sympos. Pure Math., Vol. XX, State Univ. New York, Stony Brook, N.Y., 1969)*, pages 415–440. Amer. Math. Soc., Providence, R.I., 1971. MR0316385 (47:4932)

[31] Daniel Shanks. The infrastructure of a real quadratic field and its applications. In *Proceedings of the Number Theory Conference (Univ. Colorado, Boulder, Colo., 1972)*, pages 217–224, Boulder, Colo., 1972. Univ. Colorado. MR0389842 (52:10672)

[32] Anitha Srinivasan. Computations of class numbers of real quadratic fields. *Math. Comp.*, 67(223):1285–1308, 1998. MR1468944 (99b:11143)

[33] The PARI Group, Bordeaux. *PARI/GP tutorial*, 2004. available from `http://pari.math.u-bordeaux.fr/doc.html`.

[34] The PARI Group, Bordeaux. *PARI/GP, version* `2.1.5`, 2004. available from `http://pari.math.u-bordeaux.fr/`.

Department of Mathematics, 530 Church Street, University of Michigan, Ann Arbor, Michigan 48109

E-mail address: `arbooker@umich.edu`

License or copyright restrictions may apply to redistribution; see https://www.ams.org/journal-terms-of-use