MATHEMATICS OF COMPUTATION
Volume 71, Number 238, Pages 863–871
S 0025-5718(01)01341-2
Article electronically published on August 3, 2001

COMPARISON OF ALGORITHMS TO CALCULATE QUADRATIC IRREGULARITY OF PRIME NUMBERS

JOSHUA HOLDEN

ABSTRACT. In previous work, the author has extended the concept of regular and irregular primes to the setting of arbitrary totally real number fields k_0, using the values of the zeta function ζ_{k_0} at negative integers as our "higher Bernoulli numbers". In the case where k_0 is a real quadratic field, Siegel presented two formulas for calculating these zeta-values: one using entirely elementary methods and one which is derived from the theory of modular forms. (The author would like to thank Henri Cohen for suggesting an analysis of the second formula.) We briefly discuss several algorithms based on these formulas and compare the running time involved in using them to determine the index of k_0-irregularity (more generally, "quadratic irregularity") of a prime number.

1. DEFINITIONS

Let k_0 be a totally real number field, and let p be an odd prime. Let $k_1 = k_0(\zeta_p)$, where ζ_{p^n} will denote a primitive p^n-th root of unity. Let $\Delta = \mathrm{Gal}(k_1/k_0)$, and let $\delta = |\Delta|$. Let p^e be the largest power of p such that $\zeta_{p^e} \in k_0(\zeta_p)$.

Definition 1. Let ζ_{k_0} be the zeta function for k_0. We say that p is k_0-*regular* if p is relatively prime to $\zeta_{k_0}(1-2m)$ for all integers m such that $2 \leq 2m \leq \delta - 2$ and also p is relatively prime to $p^e\zeta_{k_0}(1-\delta)$. The number of such zeta-values that are divisible by p will be the *index of* k_0-*irregularity* of p.

According to a well-known theorem of Kummer, p divides the order of the class group of $\mathbf{Q}(\zeta_p)$ if and only if p divides the numerator of a Bernoulli number B_m for some even m such that $2 \leq m \leq p-3$. Such primes are called irregular; the others are called regular. In the setting we have described above, the author proved in his thesis ([7], see also [8]), building on work of Greenberg and Kudo, that under a certain technical condition Kummer's criterion can be extended to give information about whether p divides the class group of $k_0(\zeta_p)$. To be exact, let k_1^+ denote the maximal real subfield of k_1, which is equal to $k_0(\zeta_p + \zeta_p^{-1})$. Let $h(k_1)$ denote the class number of k_1 and $h^+(k_1)$ denote the class number of k_1^+. It is known that $h^+(k_1) \mid h(k_1)$; we let the relative class number $h^-(k_1)$ be the quotient.

Theorem 1 (Greenberg, Holden)**.** *Assume that no prime of the field k_1^+ lying over p splits in k_1. Then p divides $h^-(k_1)$ if and only if p is not k_0-regular.*

Received by the editor July 23, 1999 and, in revised form, August 8, 2000.
2000 *Mathematics Subject Classification.* Primary 11Y40, 11Y60, 11Y16, 11B68; Secondary 11R42, 11R29, 94A60, 11R18.
Key words and phrases. Bernoulli numbers, Bernoulli polynomials, irregular primes, zeta functions, quadratic extensions, cyclotomic extensions, class groups, cryptography.

As an application, we note that one common way of constructing public-key cryptographic systems is to utilize the problem of finding a discrete logarithm in some abelian group. In order to make sure that the discrete logarithm problem is computationally hard, one needs to know something about the structure of the group involved, e.g., that it is divisible by a large prime. Theorem 1 shows that if p is a large k_0-irregular prime and the conditions of the theorem are met, then the class group of $k_0(\zeta_p)$ may be especially suitable for cryptography. (One should see [3] for more on the use of class groups in cryptography.)

For the case we consider, k_0 will be a real quadratic field $\mathbf{Q}(\sqrt{D})$, with D a positive fundamental discriminant. For such a k_0, we will say that primes are D-regular or have given index of D-irregularity, and we will let the zeta function ζ_{k_0} be also denoted by ζ_D. (More generally, we may refer to the concept as "quadratic irregularity".) In this case δ will be equal to $p-1$ unless $D=p$, in which case $\delta=(p-1)/2$. Also, e is always equal to 1 when p does not divide the order of k_0 over $\mathbf{Q}$, which is true in this case since p is odd. For the condition in Theorem 1 that no prime of the field k_1^+ lying over p splits in k_1 to be satisfied, it is sufficient that p should not divide D, and we should also note that since p does not divide the degree of $k_0=\mathbf{Q}(\sqrt{D})$ over $\mathbf{Q}$, a theorem of Leopoldt shows that p divides $h(k_1)$ if and only if p divides $h^-(k_1)$.

In general, we will consider three cost models for the time of multiplication: first using naive multiplication ($O(tt')$), second using Schönhage-Strassen fast multiplication or a similar method ($O(t(\lg t')^{O(1)})$), and third using a model where multiplication (or addition) takes constant time regardless of the size of the factors ($O(1)$). We do not expect constant time multiplication to occur asymptotically in the real world, but it can provide useful estimates in situations where the size of the numbers involved is small compared to the word size of the actual computer in question. (In these running time bounds, t is the number of bits in the larger multiplicand and t' the number of bits in the smaller.)

2. First formula

Siegel's first formula to compute $\zeta_D(1-2m)$ for $m\geq 1$ an integer is analogous to the formula $\zeta(1-2m)=-B_{2m}/(2m)$. Using elementary methods, Siegel showed that similarly

$$\zeta_D(1-2m)=\frac{B_{2m}}{4m^2}D^{2m-1}\sum_{j=1}^{D}\chi(j)B_{2m}(j/D). \tag{1}$$

Here $\chi(j)=\left(\frac{D}{j}\right)$, the Kronecker symbol, and $B_{2m}(j/D)$ indicates the $2m$-th Bernoulli polynomial evaluated at the fraction j/D. The Bernoulli polynomial $B_r(x)$ can be computed from the Bernoulli numbers as

$$B_r(x)=\sum_{s=0}^{r}\binom{r}{s}B_{r-s}x^s.$$

It is not difficult to estimate the sizes of the numbers involved. We will assume throughout that B_m, $1\leq m\leq M$, are precomputed over the common denominator of the final result, and stored in this fashion each has size $O(m(\lg m+\lg D))$ bits for a total table size of $O(M^2(\lg M+\lg D))$ bits. (The precomputation does not

in fact add to the asymptotic running time.) The rational numbers $B_{2m}(j/D)$ can also be stored in $O(m(\lg m + \lg D))$ bits, as can the total. See [6] for more details.

A first attempt at an algorithm based on (1) might compute $B_0(\alpha), \dots, B_M(\alpha)$ naively from the formula. The time taken for this would be dominated by the powerings. For $\alpha = a/b$ some rational number, the total time with naive multiplication would be $O(M^4(\lg M + \lg a + \lg b)^{O(1)})$. Using fast multiplication instead of naive multiplication would improve this to $O(M^3(\lg M + \lg a + \lg b)^{O(1)})$, while with constant time multiplication we need only time $O(M^2 \lg M)$ regardless of a and b.

However we can do better than this, using a cross between Horner's method of evaluating polynomials and an algorithm used by Brent to calculate Bernoulli numbers, as was previously discussed by the author in [6]. This method gives a total time of $O(M^3(\lg M + \lg a + \lg b)^{O(1)})$ using either constant or fast multiplication, and time $O(M^2)$ using constant time multiplication.

Using either of these algorithms to compute $\zeta_D(1-2m)$, $2 \le 2m \le M$, is then relatively straightforward. Note that the Kronecker symbol $\chi(j)$ can be computed in time $O(\lg^2 D)$. The slower version of the algorithm has time $O(M^4 D(\lg M + \lg D)^{O(1)})$ with naive multiplication, $O(M^3 D(\lg M + \lg D)^{O(1)})$ with fast multiplication, and $O(M^2 D(\lg M + \lg D)^{O(1)})$ with constant time multiplication. The faster version runs in time $O(M^3 D(\lg D + \lg M)^{O(1)})$ with either naive or fast multiplication (the $O(1)$ factor is different, of course) and again in time $O(M^2 D(\lg M + \lg D)^{O(1)})$ with constant time multiplication.

3. Second formula

Siegel's second formula is, as I said, derived from the theory of modular forms. In general, for k_0 a totally real number field as above, it says that

$$\zeta_{k_0}(1-2m) = -2^n c_{2mn}^{-1} \sum_{l=1}^{r} c_{2mn,l} s_l^{k_0}(2m),$$

where $n = [k_0 : \mathbf{Q}]$, $c_{2mn} = c_{2mn,0}$ and $c_{2mn,l}$ are rational integers depending only on $2mn$ and l (given by explicit formulas which we will discuss),

$$r = \begin{cases} \lfloor mn/6 \rfloor & \text{if } 2mn \equiv 2 \text{ modulo } 12, \\ \lfloor mn/6 \rfloor + 1 & \text{otherwise,} \end{cases}$$

and $s_l^{k_0}$ is a sum over norms of ideals in the ring of integers of k_0, namely,

$$s_l^{k_0}(2m) = \sum_{\nu \in (\mathfrak{d})^{-1},\ \nu \gg 0,\ \operatorname{tr}(\nu) = l} \sigma_{2m-1}((\nu)\mathfrak{d}),$$

where

$$\sigma_{2m-1}(\mathfrak{A}) = \sum_{\mathfrak{B} | \mathfrak{A}} N(\mathfrak{B})^{2m-1}$$

is a generalization of the usual sum of powers function and $\mathfrak{d}$ is the different of k_0. In the quadratic case this all becomes much easier:

$$\zeta_D(1-2m) = -4 c_{4m}^{-1} \sum_{l=1}^{r} c_{4m,l} s_l^D(2m), \qquad r = \lfloor m/3 \rfloor + 1, \tag{2}$$

$$s_l^D(2m) = \sum_{\nu \in (\sqrt{D})^{-1},\ \nu \gg 0,\ \mathrm{tr}(\nu)=l} \sigma_{2m-1}((\nu\sqrt{D})),$$

and $s_l^D(2m)$ can also be expressed in terms of a purely arithmetic function $e_{2m-1}(n)$, as follows:

$$s_l^{k_0}(2m) = \sum_{j|l} \chi_D(j) j^{2m-1} e_{2m-1}((l/j)^2 D)$$

and

$$e_{2m-1}(n) = \sum_{\substack{x^2 \equiv n \pmod 4 \\ |x| \le \sqrt{n}}} \sigma_{2m-1}\left(\frac{n-x^2}{4}\right)$$

where

$$\sigma_{2m-1}(n) = \sum_{d|n} d^{2m-1}$$

is the usual sum-of-powers function. (See [10], [11], [4] and [5] for more detailed descriptions of these formulas, and for their derivations.) The coefficients $c_{4m,l}$ are most easily expressed as the coefficients of a certain power series, and can be computed as needed without adding to the asymptotic running time. We will give explicit formulas for the power series and discuss its computation in Section 4. It is not hard to prove that in this form $c_{4m,l}$ is of size $O(m)$. The running time for calculating the function $e_{2m-1}(n)$ is complicated by the need for factoring; we use here an estimate based on the elliptic curve factoring method (we would expect something very similar with any of the other standard subexponential methods) to get an expected running time involving the function $\mathrm{L}(x) = e^{\sqrt{\log x \log\log x}}$. Given this, we get an expected running time to compute $e_{2m-1}(n)$ of

$$O(\sqrt{n}\,\mathrm{L}(n)^{1+o(1)} + (2m-1)^2\sqrt{n}\lg^2 n)$$

using naive multiplication. If we now applied (2) as it was originally stated to compute all $\zeta_D(1-2m)$, $2 \le 2m \le M$, we would get a running time of

$$O(M^3\sqrt{D}\,\mathrm{L}(M)^{O(1)}\,\mathrm{L}(D)^{O(1)}\lg M + M^5\sqrt{D}\lg M(\lg M + \lg D)^{O(1)}),$$

again using naive multiplication.

However, it is more efficient to rearrange the terms of the formula as follows:

$$\begin{aligned}
\zeta_D(1-2m) &= -4c_{4m}^{-1}\sum_{l=1}^{r} c_{4m,l}\, s_l^{k_0}(2m) \\
&= -4c_{4m}^{-1}\sum_{l=1}^{r} c_{4m,l}\sum_{j|l}\chi_D(j) j^{2m-1} e_{2m-1}((l/j)^2 D) \\
&= -4c_{4m}^{-1}\sum_{k=1}^{r}\left(\sum_{j=1}^{\lfloor r/k\rfloor}\chi_D(j) j^{2m-1} c_{4m,jk}\right) e_{2m-1}(k^2 D). \qquad (3)
\end{aligned}$$

This rearrangement of the formula requires fewer calls to compute e_{2m-1} by a factor of $\lg m$. Using this version of the formula, the time necessary to compute all $\zeta_D(1-2m)$, $2 \le 2m \le M$, using naive multiplication is

$$O(M^3\sqrt{D}\,\mathrm{L}(M)^{O(1)}\,\mathrm{L}(D)^{O(1)} + M^5\sqrt{D}(\lg M + \lg D)^{O(1)}).$$

This is much worse than the best algorithm based on (1) in terms of M, but it is better in terms of D. Also, except for one final division by c_{4m}^{-1}, all of the arithmetic in this formula deals only with rational integers, unlike the previous formulas. Note that the first term comes from the factoring process, while the second term comes from multiplications.

It should be noted that the asymptotic running time of this algorithm is greatly improved by using Schönhage-Strassen fast multiplication or constant time multiplication, in which cases the second term becomes smaller than the first and the running time becomes

$$O(M^3\sqrt{D}\,\mathrm{L}(M)^{O(1)}\,\mathrm{L}(D)^{O(1)}).$$

This is still worse than using (1) in terms of M, but only by a subexponential factor.

It should also be noted that (2) and (3) also present opportunities for time savings when computing zeta-values for multiple D in the same range of M, at a sacrifice of memory space. The controlling factor in the speed of the algorithm is the number of times that $\sigma_{2m-1}(n)$ must be calculated. Note that in computing all $\zeta_d(1-2m)$, $5 \le d \le D$, there can only be $O(m^2D)$ different values of n. However, following the algorithm strictly, we would ordinarily make $O(m^2D^{3/2})$ calls to the subroutine that calculates this function.

Thus if we compute all $\zeta_d(1-2m)$, $5 \le d \le D$, storing values of $\sigma_{2m-1}(n)$ as we compute them, and then repeat this process for each m in the range $2 \le 2m \le M$, the running time should be $O(D\,\mathrm{L}(D)^{O(1)})$ in terms of D, rather than $O(D^{3/2}\,\mathrm{L}(D)^{O(1)})$ as one would obtain following the algorithm strictly. This compares very favorably with the time of $O(D^2)$ in terms of D which holds for algorithms using (1).

Since the exponent $2m-1$ used in the $\sigma_{2m-1}(n)$ function changes as m does, we can dispose of the table when we change m. The table that we need to keep requires at most $O(M^3D(\lg M + \lg D))$ bits of storage, which could be a significant barrier. More efficient storage of the important information may be valuable here; we will discuss this somewhat more in Section 5.

4. Computing the numbers $c_{4m,l}$

The integers $c_{4m,l}$ are defined as follows. Let

$$G_k = 1 - \frac{2k}{B_k}\sum_{n=1}^{\infty}\sigma_{k-1}(n)\,q^n$$

be the (normalized) Eisenstein series of order k for $k = 6, 10$, and 14. (For the general $c_{2mn,l}$ one also needs $k = 0, 4$, and 8.) Let

$$\begin{aligned}\Delta &= q\prod_{n=1}^{\infty}(1-q^n)^{24}\\ &= q\left(\sum_{n=0}^{\infty}(-1)^n(2n+1)\,q^{n(n+1)/2}\right)^8\end{aligned}$$

be the discriminant series. Let $r = \lfloor m/3\rfloor + 1$ as before, and let

$$T_{4m} = G_{12r-4m+2}\Delta^{-r} = \sum_{n=-r}^{\infty} c_{4m,-n}\,q^n.$$

Then $c_{4m} = c_{4m,0}$, and the other $c_{4m,l}$ for $1 \le l \le r$ can also be read off as coefficients of T_{4m}. Luckily, the expression $12r - 4m + 2$ only takes on the values 6, 10, and 14. (In the more general case we can define T_{2mn} similarly; the expression $12r - 2mn + 2$ can take on the values 0, 4, and 8 in addition to those above.)

The best algorithm known to the author for calculating these coefficients goes roughly as follows. At the start of the computations for $\zeta_D(1-2m)$, $2 \le 2m \le M$, calculate G_6, G_{10}, G_{14}, and Δ^{-1} with the maximum number of coefficients necessary (about $M/12$). Instead of trying to compute all of the needed series Δ^{-r} at once, we calculate it as a running product which only needs to be updated when r changes. Then, whenever m changes, we multiply truncated versions $G_{12r-4m+2}$ and Δ^{-r} (with about $m/6$ coefficients each) to find the required coefficients of T_{4m}.

The series Δ^{-1} can also be expressed as

$$\begin{aligned}\Delta^{-1} &= q \prod_{n=1}^{\infty} (1-q^n)^{-24} \\ &= q \left(\prod_{n=1}^{\infty} \frac{1}{(1-q^n)} \right)^{24} \\ &= q \left(\sum_{n=0}^{\infty} p(n)\, q^n \right)^{24}\end{aligned}$$

where $p(n)$ takes on integer values and is well-known as the partition function from additive number theory. Hardy and Ramanujan proved an asymptotic expression for $p(n)$ which shows that $\lg p(n)$ is of order $\sqrt{n}$. (See, for example, Chapter 14 of [1].) From this it is easy to show that the coefficients of Δ^{-r} take at most $O(M^{1.5})$ bits of storage each, as do the coefficients of $G_{12r-4m+2}$ and T_{4m}. Thus the storage required for all of the computation of $c_{4m,l}$ necessary for a fixed m is $O(M^{2.5})$. The resulting table, which is of course independent of d, can be used to compute $\zeta_d(1-2m)$ for any range of d and then disposed of when m is changed.

As far as the time for this algorithm is concerned, computing the tables necessary for all $\zeta_d(1-2m)$, $2 \le 2m \le M$, $5 \le d \le D$, takes time $O(M^5)$ with naive multiplication of integers and also of polynomials. By using FFT methods to multiply polynomials the time can be reduced to $O(M^{3.5} \lg M \lg\lg M)$ with fast multiplication of integers and $O(M^2 \lg M)$ with constant time multiplication of integers. (The use of FFT methods here was suggested by A.O.L. Atkin and Will Galway.) This is within our previously established time bounds in the naive and constant time cases; however in the fast multiplication case it could add to the total asymptotic time in terms of M, which for the previous parts of the algorithm was established as $O(M^3\, \mathrm{L}(M)^{O(1)})$ in terms of M. On the other hand it should be noted that these calculations only need to be done once per value of m no matter how many values of d one is examining. Also the constant involved in the $O(M^{3.5} \lg M \lg\lg M)$ seems to be quite good in practice compared to that in the $O(M^3\, \mathrm{L}(M)^{O(1)})$.

5. Summary and future work

Table 1 presents the various algorithms, for comparison. We present the asymptotic order of the running time to compute $\zeta_D(1-2m)$, $2 \le 2m \le M$, for a given D, using the three methods of multiplication discussed earlier.

TABLE 1. Comparison of algorithms for $\zeta_D(1-2m)$, $2 \le 2m \le M$

Equation used	Multiplication	Time order
(1)	naive	$M^4 D(\lg M + \lg D)^{O(1)}$
(1)	fast	$M^3 D(\lg M + \lg D)^{O(1)}$
(1)	constant	$M^2 D(\lg M + \lg D)^{O(1)}$
(1) from [6]	naive	$M^3 D(\lg D + \lg M)^{O(1)}$
(1) from [6]	fast	$M^3 D(\lg D + \lg M)^{O(1)}$
(1) from [6]	constant	$M^2 D(\lg M + \lg D)^{O(1)}$
(2)	naive	$M^3\sqrt{D}\,\mathrm{L}(M)^{O(1)}\,\mathrm{L}(D)^{O(1)}\lg M + M^5\sqrt{D}(\lg M + \lg D)^{O(1)}$
(2)	fast	$M^3\sqrt{D}\,\mathrm{L}(M)^{O(1)}\,\mathrm{L}(D)^{O(1)}\lg M$
(2)	constant	$M^3\sqrt{D}\,\mathrm{L}(M)^{O(1)}\,\mathrm{L}(D)^{O(1)}\lg M$
(3)	naive	$M^3\sqrt{D}\,\mathrm{L}(M)^{O(1)}\,\mathrm{L}(D)^{O(1)} + M^5\sqrt{D}(\lg M + \lg D)^{O(1)}$
(3)	fast	$M^3\sqrt{D}\,\mathrm{L}(M)^{O(1)}\,\mathrm{L}(D)^{O(1)}$
(3)	constant	$M^3\sqrt{D}\,\mathrm{L}(M)^{O(1)}\,\mathrm{L}(D)^{O(1)}$

TABLE 2. Calculating $\zeta_D(1-2m)$, $2 \le 2m \le M$

D	M	time (1) from [6]	(stacksize)	time (3)	(stacksize)
5	100	3.155	(10M)	.838	(10M)
101	100	55.561	(10M)	3.758	(10M)
501	100	4:50.282	(10M)	12.480	(10M)
1001	100	10:52.411	(10M)	20.670	(10M)
5001	100	48:27.107	(12M)	1:43.548	(10M)
5	500	2:05.670	(4M)	8:11.603	(4M)
101	500	42:38.612	(4M)	1:18:26.615	(4M)
501	500	3:48:18.615	(8M)	4:45:20.438	(4M)
1001	500	7:49:56.048	(12M)	7:49:00.783	(4M)
5	1000	10:05.903	(4M)	3:55:46.908	(4M)
5	2000	1:14:01.992	(16M)	118:17:46.020	(4M)

The factor of $\lg M$ in the times for algorithms based on (2) has been included to emphasize that these algorithms are slower than those based on (3), even though the factor of $\lg M$ could be absorbed into that of $\mathrm{L}(M)^{O(1)}$.

We also provide, in Tables 2 and 3, tables of actual timings for some of the algorithms, using naive multiplication. The times were measured on a Sun SPARC Ultra-1 computer using the GP-Pari interpreted language. (See [2].) The data computed by these programs will be analyzed in a future paper.

Table 2 measures the time to compute $\zeta_D(1-2m)$, $2 \le 2m \le M$, for a given D, in hours, minutes, and seconds. The number in parentheses indicates the size of the stack used, in bytes. These numbers are only a very rough guide to the actual amount of memory used.

Table 3 measures the time to compute $\zeta_d(1-2m)$, $2 \le 2m \le M$, $5 \le d \le D$. We use the algorithm based on (3), both with and without keeping a table of $\sigma_{2m-1}(n)$

TABLE 3. Calculating $\zeta_d(1-2m)$, $2 \le 2m \le M$, $5 \le d \le D$

D	M	time (3)	(stacksize)	time (3) with table	(stacksize)
100	100	1:19.638	(4M)	1:27.983	(4M)
500	100	18:34.377	(4M)	13:57.166	(8M)
1000	100	1:03:50.464	(4M)	37:52.552	(12M)
5000	100	26:39:46.109	(4M)	7:10:44.870	(64M)

as described earlier. The units and stack size numbers should be interpreted as in Table 2.

As one can see, memory usage for algorithms based on (3) with a table of values $\sigma_{2m-1}(n)$ goes up quite quickly, and even so a lot of redundant work is being done. For one thing, the same numbers n will have to be factored repeatedly for different values of $2m-1$, but they will lead to different values of $\sigma_{2m-1}(n)$ so the actual factorization would need to be stored and not just a function value. This would be even more memory intensive. Carl Pomerance has suggested some approaches to this, including using a cache rather than a complete table, and storing only the largest prime factor rather than a complete factorization. The question of a cache leads naturally to the question of which numbers will appear as integer values of $(k^D - x^2)/4$ as k and D vary, and how often. Henri Cohen, in [5], gives some variations on Siegel's formula which could cut down on the number of times $\sigma_{2m-1}(n)$ needs to be computed. Algorithms based on these might provide a linear speedup over the algorithms presented here, but the asymptotic behavior would probably be the same.

The anonymous reviewer has suggested that some of the arithmetic could possibly be speeded up by the use of modular techniques and the Chinese remainder theorem. This certainly deserves more consideration.

Another prospect for future work is the analysis of "first-hit" versions of these algorithms, namely determining how long we should expect to search before finding, say, the first D-irregular prime larger than a certain bound for D in a given range. This might be particularly useful for cryptographic applications, in which we would be explicitly looking for a class group (or a small number of them) with a hard discrete logarithm problem. This will be explained in more detail in [9].

ACKNOWLEDGMENTS

The author would like to thank A.O.L. Atkin, Will Galway, and the members of the NMBRTHY electronic mailing list for their very helpful suggestions, and Robert Harley for generating some of the precomputed tables used in the computations reported in this paper. He would also like to especially thank Johannes Buchmann, Henri Cohen, and Carl Pomerance for suggestions and encouragement, and Gary Walsh for his encouraging remarks on an early version of this paper. Finally, he would like to thank the anonymous reviewer for encouragement and several helpful suggestions.

REFERENCES

[1] Tom M. Apostol, *Introduction to analytic number theory*, Undergraduate Texts in Mathematics, Springer-Verlag, 1976. MR **55:**7892

[2] C. Batut, K. Belabas, D. Bernardi, H. Cohen, and M. Olivier, *User's guide to PARI-GP*, Laboratoire A2X, Université Bordeaux I, version 2.0.9 ed., May 13, 1998,

`<http://hasse.mathematik.tu-muenchen.de/ntsw/pari/Welcome.html>, <ftp://megrez.-math.u-bordeaux.fr>`.

[3] Johannes Buchmann and Sachar Paulus, *A one way function based on ideal arithmetic in number fields*, Advances in cryptology—CRYPTO '97 (Burton S. Kaliski, Jr, ed.), Lecture Notes in Computer Science, vol. 1294, Springer-Verlag, 1997, pp. 385–394.

[4] Henri Cohen, *Sums involving the values at negative integers of L-functions of quadratic characters*, Math. Ann. **217** (1975), 271–285. MR **52:**10615

[5] ———, *Variations sur un thème de Siegel et Hecke*, Acta Arith. **30** (1976), 63–93. MR **54:**10207

[6] Joshua Holden, *Irregularity of prime numbers over real quadratic fields*, Algorithmic Number Theory: Third International Symposium; Proceedings (J. P. Buhler, ed.), Springer Lecture Notes in Computer Science, vol. 1423, Springer-Verlag, 1998, pp. 454–462. MR **2000m:**11113

[7] ———, *On the Fontaine-Mazur conjecture for number fields and an analogue for function fields*, Ph.D. thesis, Brown University, 1998.

[8] ———, *On the Fontaine-Mazur Conjecture for number fields and an analogue for function fields*, J. Number Theory **81** (2000), 16–47. MR **2001e:**11111

[9] ———, *First-hit analysis of algorithms for computing quadratic irregularity*, (In preparation).

[10] Carl Ludwig Siegel, *Bernoullische Polynome und quadratische Zahlkörper*, Nachr. Akad. Wiss. Göttingen Math.-Phys. Kl. II **2** (1968), 7–38. MR **38:**2123

[11] Don Zagier, *On the values at negative integers of the zeta-function of a real quadratic field*, Enseign. Math. (2) **22** (1976), 55–95. MR **53:**10742

Department of Mathematics and Statistics, University of Massachusetts at Amherst, Amherst, Massachusetts 01003

E-mail address: `holden@math.duke.edu`

Current address: Department of Mathematics, Rose-Hulman Institute of Technology, 5500 Wabash Ave., Terre Haute, Indiana 47803

E-mail address: `holden@rose-hulman.edu`

URL: `http://www.rose-hulman.edu/~holden`