# Efficient Verification of Tunnell's Criterion

Eric Bach[a,*] and Nathan C. Ryan[b,*]

[a] Computer Sciences Department, University of Wisconsin
1210 W. Dayton St., Madison, WI 53706–1685
E-mail: bach@cs.wisc.edu

[b] Department of Mathematics, Bucknell University
Lewisburg, PA 17870
E-mail: nathan.ryan@bucknell.edu

Received October 4, 2006
Revised January 15, 2007

An integer $n$ is congruent if there is a triangle with rational sides whose area is $n$. In the 1980s Tunnell gave an algorithm to test congruence which relied on counting integral points on the ellipsoids $2x^2 + y^2 + 8z^2 = n$ and $2x^2 + y^2 + 32z^2 = n$. The correctness of this algorithm is conditional on the conjecture of Birch and Swinnerton-Dyer. The known methods for testing Tunnell's criterion use $O(n)$ operations. In this paper we give several methods with smaller exponents, including a randomized algorithm using time $n^{1/2+o(1)}$ and space $n^{o(1)}$, and a deterministic algorithm using space and time $n^{1/2+o(1)}$.

*Key words*: congruent numbers, quadratic forms, class field theory, algorithms, complexity

## 1. Introduction

A positive integer $n$ is said to be congruent if there exists a right triangle with rational sides whose area is $n$. Since this property is unaffected when we multiply $n$ by a square, we may as well assume that $n$ is squarefree. A natural way to study this property is to consider the elliptic curve $E_n$, whose Weierstrass equation is $y^2 = x^3 - n^2x$. Let $E_n(\mathbb{Q})$ denote the group of rational points on $E_n$. Its torsion group has order 4 (the point at infinity, together with three points at which $y = 0$). From this observation, one can see that $n$ is congruent if and only if $E_n(\mathbb{Q})$ has positive rank.

In the 1980s, Tunnell [24], using results of Coates and Wiles, and the Shimura correspondence, showed that when $n$ is congruent, the number of integral points on a specific ellipsoid was half the number of points on another specific one (see Section 2 for more details). Assuming the Birch and Swinnerton-Dyer conjecture (BSD), he also showed the converse.

Proofs of these assertions, as well as a statement of this conjecture, may be found in the book of Koblitz [13].

[*] The RMMC Summer School in Computational Number Theory and Cryptography was funded by NSF grant DMS-0612103, the Rocky Mountain Mathematics Consortium, The Number Theory Foundation, The Institute for Mathematics and its Applications (IMA), The Fields Institute, The Centre for Information Security and Cryptography (CISaC) and iCORE of Canada. As this paper is an outgrowth of this conference, the authors would like to thank all these sponsors. We would also like to thank the University of Wyoming for hosting the conference. Thanks to Dieter van Melkebeek for quantum advice. Also the authors are indebted to the referees for their comments that made the exposition more clear.

Thus, assuming BSD, there is an algorithm to test congruence. The purpose of this article is to investigate efficient realizations of this algorithm. We will present methods for verifying the congruence of an integer $n$ that are more efficient than those present in the literature. In particular, our methods use $n^{1/2+o(1)}$ arithmetic operations, whereas those previously known use $O(n)$. Even with BSD, it is not clear whether congruence can be tested using $O(n^{\alpha})$ arithmetic operations, with $\alpha < 1/2$.

The paper is organized as follows. In Section 2 we give Tunnell's criterion, and in Section 3 we review results we will need on representation by quadratic forms. The algorithms themselves are described in Sections 4 and 5, and their experimental tests in Section 6. In Section 7, we review the other algorithms known to us, and in Section 8, we conclude with an investigation of complexity classes containing the set of congruent numbers.

## 2. Tunnell's criterion

The main result of [24] is the following theorem.

Theorem 2.1 (Tunnell's criterion [24]). *Assume the conjecture of Birch and Swinnerton-Dyer. Then a squarefree positive integer $n$ is congruent if and only if*

$$\begin{aligned}&\left|\left\{(x,y,z)\in\mathbb{Z}^3\colon 2ax^2+y^2+8z^2=\frac{n}{a}\right\}\right|\\ &\quad=2\left|\left\{(x,y,z)\in\mathbb{Z}^3\colon 2ax^2+y^2+32z^2=\frac{n}{a}\right\}\right| \qquad (2.1)\end{aligned}$$

*where*

$$a=\begin{cases}1 & \text{if } n \text{ is odd,}\\ 2 & \text{if } n \text{ is even.}\end{cases}$$

We will refer to the quadratic form $2x^2+y^2+8z^2$ as $Q_1$.

A couple of results can be checked immediately—they are both found in [13]. First, if $n \equiv 5,6,7 \pmod 8$ we deduce that Tunnell's criterion holds. Second,

Proposition 2.2. (2.1) *holds for odd squarefree $n$ if and only if*

$$\begin{aligned}&\left|\left\{(x,y,z)\in\mathbb{Z}^3\colon Q_1(x,y,z)=n,\ z \text{ odd}\right\}\right|\\ &\quad=\left|\left\{(x,y,z)\in\mathbb{Z}^3\colon Q_1(x,y,z)=n,\ z \text{ even}\right\}\right|. \qquad (2.2)\end{aligned}$$

*For even squarefree $n$, replace $n$ by $n/2$ and $Q_1$ by the quadratic form $Q_3(x,y,z)=4x^2+y^2+8z^2$.*

A brute force way to count the number of solutions on either side of (2.1) is to iterate $x$, $y$ and $z$ from 0 to $\sqrt{n}$ and track the number of solutions to either side. This requires examining $O(n^{3/2})$ triples, as remarked by Tunnell. In what follows we reduce the exponent to 1/2.

## 3. Some algebraic number theory

For a quadratic form (binary or ternary), let $r_Q(n)$ denote the number of integer solutions to $Q(x,y,z)=n$ (or $Q(x,y)=n$). The number $r_Q(n)$ is called the representation number of $n$.

Let $Q(x,y)=x^2+2y^2$. To produce our algorithm in Section 4 we will need to determine (a) for which odd $n$ we have $r_Q(n)>0$ and, if possible, (b) the number $r_Q(n)$ itself.

Let $K=\mathbb{Q}(\sqrt{-2})$. Let $p \in \mathbb{Z}$ be an odd prime. Look at the factorization of $p$ in $\mathcal{O}_K=\mathbb{Z}[\sqrt{-2}]$, the ring of integers of $K$. Let $\left(\frac{a}{p}\right)$ be the Legendre symbol. Then we have the following standard result (see e.g., [12]):

Lemma 3.1.

(a) *$p \equiv 1,3 \pmod 8$ iff $\left(\frac{-2}{p}\right)=1$ iff $p$ splits in $K$.*

(b) *$p \equiv 5,7 \pmod 8$ iff $\left(\frac{-2}{p}\right)=-1$ iff $p$ is inert in $K$.*

(c) *2 ramifies.*

Using Lemma 3.1 we can answer question (a) as posed above for $Q(x,y)=x^2+2y^2$. Our argument is a special case of an argument that can be found in, for example, [2]. Suppose the odd $n$ factors as

$$\prod_{p_j \equiv 1,3 \pmod 8} p_j^{\alpha_j} \prod_{p_k \equiv 5,7 \pmod 8} p_k^{\beta_k}.$$

First, we observe that the $\beta_k$ are even. If $p_k \mid x^2+2y^2$, then $p_k \mid x$ and $p_k \mid y$. If this were not the case, then $(-2 \,|\, p_k)=-1$ would be contradicted. Canceling the $p_k^2$ on both sides we can apply the same argument inductively and deduce that if $p_k \mid x^2+2y^2$, then it must do so with an even exponent.

Since $\mathbb{Z}[\sqrt{-2}]$ is a unique factorization domain, up to units $n$ factors in $\mathcal{O}_K$ as

$$n=\prod_j P_j^{\alpha_j} P_j'^{\alpha_j} \prod_k p_k{}^{\beta_k}$$

where the $P_j'$ is the complex conjugate of $P_j$ and the $\beta_k$ are as above. Moreover, we can let $P_j = x_j+\sqrt{-2}y_j$ where $x_j$ and $y_j$ are the unique positive integers so that $p_j=x_j^2+2y_j^2$.

The unit group of $O_K$ is $\{\pm 1\}$ and so we note that $\frac{1}{2}r_Q(n)$ is a multiplicative function. To determine $\frac{1}{2}r_Q(n)$, we count elements of norm $p^\alpha \parallel n$ in $\mathcal{O}_K$. The local factors are determined in this way: if $p=2$, there is evidently only one element of norm $2^\alpha$ since 2 ramifies. If $p \equiv 1,3 \pmod 8$, we note that an element of norm $p^\alpha$ factors as $P^\gamma(P')^{\alpha-\gamma}$ for $0 \le \gamma \le \alpha$. Thus, for such primes, the local factor is $\alpha+1$. If $p \equiv 5,7 \pmod 8$ and $\alpha$ is odd we observe, by the above, that there are no solutions to $x^2+2y^2=n$ and so in this case, the local factor is 0. In the final case, if $p \equiv 5,7 \pmod 8$ and $\alpha$ is even, since such $p$ are inert, the only element of norm $p^\alpha$ is $p^\alpha$ itself.

To test congruence of even numbers we will need to determine the number of solutions to

$$4x^2 + y^2 = n \tag{3.1}$$

from the factorization of $n$. This can be reduced to studying sums of squares, as follows. We note that (3.1) is the same as $n = z^2 + y^2$, with $z$ even. So we consider the parity of $z$ and $y$. If $n$ is odd, exactly one of $z, y$ is even, so the number of solutions to (3.1) is half the number of solutions to $z^2 + y^2 = n$. If $n$ is even, then $z$ and $y$ are both odd or both even. In the first case, we don't get a solution to (3.1), and in the second case, $n$ must be divisible by 4. Hence, when $n \equiv 2 \pmod 4$, there are no solutions to (3.1), and when $n \equiv 0 \pmod 4$, the number of solutions equals the number of solutions to $x^2 + y^2 = n/4$.

To count the number of solutions to $x^2 + y^2 = m$, we first count the ideals of norm $m$ in $\mathbb{Z}[i]$, similarly to what was done above for $x^2 + 2y^2 = n$. This number is the product of local factors, as follows. Suppose that $p^\alpha \parallel m$. Then the local factor is 1 for $p = 2$, $1 + \alpha$ for $p \equiv 1 \pmod 4$, and 1 (resp. 0) for $p \equiv 3 \pmod 4$ and $\alpha$ even (resp. odd). To get the number of representations, we multiply together the local factors for each prime dividing $m$, and multiply by 4 since $\mathbb{Z}[i]$ has 4 units. For example

$$2340 = 22 \cdot 5 \cdot 13 \cdot 32$$

has $4 \cdot 1 \cdot 2 \cdot 2 \cdot 1 = 16$ representations.

The local factors can be justified as follows. In $\mathbb{Z}[i]$, 2 is ramified, so there is only one ideal of norm $2^\alpha$. An odd prime $p$ is inert if it is 3 (mod 4), and split otherwise. In the first case, there will be an ideal of norm $p^\alpha$ exactly when $\alpha$ is even, and only one such. In the second case, write $p = PP'$ for prime ideals $P$ and $P'$. By unique ideal factorization, an ideal $Q$ has norm $p^\alpha$ iff

$$Q = P^\gamma (P')^{\alpha-\gamma}, \quad 0 \le \gamma \le \alpha.$$

As before, there are $\alpha + 1$ choices for $\gamma$.

We conclude this section with a historical note. Certainly, counting representations of integers by quadratic forms is a classical problem. For the forms of interest to us, these counts were often expressed by tallying divisors in various congruence classes. A number of such results appear in Hua [11, pp. 307–309]; see also Dickson [4, vol. 2, p. 248; vol. 3, p. 38]. Gauss [10, art. 182] counted the solutions to $x^2 + y^2$ as we have, by grouping prime factors into arithmetic progressions. Nowadays, we recognize this as a superior method, because factoring is generally more efficient than listing divisors.

## 4. An improvement

Our improvement over the brute force method replaces a triple loop (over $x, y, z$) by a single one that iterates $z$ from 0 to $\sqrt{n/8}$. By Proposition 2.2 we need only count solutions to $Q(x, y, z) = n$, and tally these separately for even and odd $z$. The solution-counts use the theorems in Section 3.

Using Algorithm 1 we can derive Algorithm 2, implicit in Theorem 2.1.

Algorithm 1. Counts representations by the quadratic forms in (2.1)

```
Require: $n \in \mathbb{Z}_+$ odd
  $evenCount \leftarrow 0$
  $oddCount \leftarrow 0$
  for $z = 0$ to $\sqrt{n/8}$ do
    Factor $n - 8z^2$ as $\prod_{p_j \equiv 1,3 \pmod 8} p_j^{\alpha_j} \prod_{p_k \equiv 5,7 \pmod 8} p_k^{\beta_k}$.
    if all $\beta_k$ odd then
      $count \leftarrow count + 2\prod_j (1 + \alpha_j)$
    end if
    if $z$ even then
      $evenCount \leftarrow evenCount + count$
    else
      $oddCount \leftarrow oddCount + count$
    end if
  end for
  return $oddCount$, $evenCount$
```

Algorithm 2. Determines if odd squarefree $n$ is congruent

```
Require: $n \in \mathbb{Z}_+$ odd squarefree
  Apply Algorithm 1 to $n$.
  if $oddCount(n) = evenCount(n)$ then
    return true
  else
    return false
  end if
```

In broad strokes, Algorithm 2 proceeds as follows: factor all $m$ of the form

$$n - 8z^2 \quad \text{where} \quad 0 \leq z \leq \sqrt{n/8}.$$

Using the formula for the number of solutions to $x^2 + 2y^2 = m$ derived in Section 3, we count the solutions for all such solutions. If the total for even $z$ equals the total for odd $z$, then we know Tunnell's criterion is satisfied. Note that Proposition 2.2 allows us to consider only one quadratic form, rather than two.

What is the cost of this algorithm? With present knowledge, the complexity of factoring depends on what kind of algorithm is allowed. More precisely, to factor a number less than $n$ uses $n^{c+o(1)}$ operations, where

$$c = \begin{cases} \frac{1}{4} & \text{unconditionally [15],} \\ \frac{1}{5} & \text{assuming ERH [18],} \\ 0 & \text{if random choices are allowed [5].} \end{cases}$$

Therefore, Algorithm 2 will determine if $n$ satisfies Tunnell's criterion using a number of operations that is

$$\begin{cases} n^{\frac{3}{4}+o(1)} & \text{unconditionally,} \\ n^{\frac{7}{10}+o(1)} & \text{assuming ERH,} \\ n^{\frac{1}{2}+o(1)} & \text{with randomness.} \end{cases}$$

Note that the first two results refer to deterministic algorithms. Since probabilistic factoring algorithms can be made errorless [23], the third algorithm is of Las Vegas type (always correct, probably fast).

Since Algorithm 1 actually factors $n$, it can be used to tell if $n$ is squarefree. If $n$ is not squarefree, the factorizations it computes are still useful, since if $n = n'r^2$, and $z = z'r$, we have

$$n - 8z^2 = r^2(n' - 8z'^2).$$

We remark that the above algorithm will find representation numbers for any ternary quadratic form

$$x^2 + by^2 + cz^2$$

so that the class number of quadratic forms with discriminant $-4b$ class number of $x^2 + by^2$ is 1; in other words, if $b$ is a Heegner number.

## 5. Factoring with sieves

In this section we discuss a modification to Algorithm 1 that improves the running time, at the cost of more space. Recall that Algorithm 1 factors $n - 8z^2$ for all $z$, $0 \le z < \sqrt{n/8}$. Because these are successive values of a fixed polynomial, we can factor them together using a sieve. This is similar to what is done in the quadratic sieve factoring algorithm [16].

We will assume that $n$ is odd. (Modifications for $n$ even are left to the reader.) It will be enough to partially factor $n - 8z^2$ over the primes up to $\sqrt{n}$, since any unfactored part left must be prime. We use the sieve for these partial factorizations, following with a "clean up" phase to obtain complete factorizations.

In the sieving process, $p = 2$ can be skipped, since $n - 8z^2$ is odd if $n$ is. We note also that for odd $p$, we have $p \mid n - 8z^2$ iff $z^2 \equiv n/8 \pmod{p}$. This is only possible if $p \mid n$, or if the Legendre symbol $(2n \,|\, p) = +1$.

We now analyze the complexity of this procedure. By Mertens's theorem, the cost of marking the $z$'s will be about

$$\sqrt{n} \sum_{p \le \sqrt{n}} \frac{2}{p} = O(n^{1/2} \log\log n)$$

operations. We need to solve $\pi(\sqrt{n}) - 1 \sim 2\sqrt{n}/\log n$ quadratic equations mod $p$. Such equations are solvable in polynomial time, using the Tonelli–Shanks algorithm [19], provided a quadratic nonresidue for $p$ is available. Since we will do this for many $p$, it will suffice to test $a = 2, 3, 5, \dots$ until we find a number with $(a \,|\, p) = -1$. Erdős [7] proved that if $n(p)$ is the least quadratic nonresidue for $p$, then

$$\frac{1}{\pi(x) - 1} \sum_{3 \le p \le x} n(p) = O(1).$$

Thus, the number of Legendre symbols that will be computed in this process is at most a constant times

$$\pi(\sqrt{n}) = O(n^{1/2}/(\log n)).$$

Each Legendre symbol can be computed in polynomial time, using a variant of the Euclidean algorithm [3]. Processing each $z$ in the last stage will use $O(\log n)$ operations, since a number bounded by $n$ can have at most $\log_2 n$ prime factors. Putting everything together, we see that Algorithm 3 uses $n^{1/2+o(1)}$ operations.

Algorithm 3. Factors all positive $n - 8z^2$

```
Require: $n \in \mathbb{Z}_+$ odd
  for primes $p$, $3 \le p \le \sqrt{n}$ do
    if $p \mid n$, mark all $z \equiv 0 \pmod{p}$ with the factor $p$.
    if $(2n \,|\, p) = -1$, skip this prime.
    if $(2n \,|\, p) = +1$, let $z_1, z_2$ be the roots of $8z^2 \equiv n \pmod{p}$, with $0 < z_1 < z_2 < p$. Mark
        $z_1, z_1 + p, z_1 + 2p, \dots, z_1 + k_1 p$
    and
        $z_2, z_2 + p, z_2 + 2p, \dots, z_2 + k_2 p$
    with the factor $p$. It will be necessary to go up to $k_i \le (\sqrt{n/8} - z_i)/p$.
  end for
  for $z = -\sqrt{n/8}$ to $\sqrt{n/8}$ do
    let $p_1, \dots, p_r$ be the factors of $n - 8z^2$ found above.
    for each $i$, find $e_i$ such that $p_i^{e_i} \parallel n - 8z^2$.
    if $q = n/(p_1^{e_1} \cdots p_r^{e_r})$ is greater than 1, let
        $p_{r+1} = q, \quad e_{r+1} = 1.$
  end for
```

In practice, one might want to use a randomized algorithm to solve the quadratic equations, and there are several choices available for this. As we have stated it, however, Algorithm 3 is deterministic and does not rely on ERH or any other heuristic assumption. We thus get an improvement on the running times of the last section.

Algorithm 3 will need space for the factorizations of $O(\sqrt{n})$ numbers bounded by $\sqrt{n}$, that is, $O(\sqrt{n}\log n)$ bits. Thus it will require more space than a version of Algorithm 1 that "outsources" factorization. The space required by most recent factoring algorithms, including the quadratic sieve [16], is $n^{o(1)}$.

## 6. Implementations of the algorithms

Both Algorithm 2 and Algorithm 3 were implemented and timed using Victor Shoup's NTL [20]. NTL had all the required functionality except an integer factorization algorithm. The second author implemented Richard Brent's [1] variant of Pollard's rho algorithm. We tested each algorithm against the odd squarefree integers less than a million. We tested each algorithm a number of times, and on average Algorithm 3 took fifty percent longer than Algorithm 2. Tunnell conjectured that for non-congruent squarefree numbers, the modular form coefficient described in the main theorem of [24] is always an integral multiple of the number of divisors of $n$. We found no exceptions to this conjecture.

## 7. Other methods

In this section we review the other known methods for testing Tunnell's criterion.

As we noted above, a brute force computation uses $O(n^{3/2})$ operations. The exponent can be reduced to 1 by a "collision" method. We first make a list of $O(n)$ triples $(x^2+2y^2, x, y)$ and another list of $O(\sqrt{n})$ pairs $(n-8z^2, z)$. Sorting these on the first component provides us with all solutions to $2x^2+y^2+8z^2=n$.

Another possibility is to use an analytic formula to compute $L(E_n, 1)$ [13, p. 95]. This value is essentially the square of the difference count in Proposition 2.2, as the ratio of these two values is an easily computed constant times $n^{-1/2}$. Thus, $L(1)$ is either 0 or $\Omega(n^{-1/2})$. By the estimates in [13] the coefficients in the series for $L(1)$ are such that we would need $O(n\log n)$ terms in this series to tell these two cases apart. This is thus another method with cost $n^{1+o(1)}$.

Results of Elkies [6] suggest a point of infinite rank on the curve $y^2=x^3-n^2x$ can be found in time $O(n^{1/2})$. These results hold only for $n\equiv 5,7 \pmod 8$, which we know satisfy Tunnell's criterion. So this method is not an improvement over our methods.

Yet another method was suggested to the second author by William Stein and also requires looking at the curve $E_n\colon y^2=x^3-n^2x$. The curve $E_n$ is the quadratic twist of $E_1$, by a quadratic character of conductor dividing $4n$. Thus $L(E_n,1)\neq 0$ if and only if the $\chi$-twisted winding element in the 1-dimensional space $H_1(X_0(32),\mathbb{Q})^+$ of modular symbols of level 32 and weight 2 for $\Gamma_0(32)$ is nonzero. Computing this $\chi$-twisted winding element takes $O(n)$ arithmetic operations as it requires one to sum over the $a$ in $\mathbb{Z}/n\mathbb{Z}$. Our methods have exponents less than 1, and so are asymptotically faster than this method.

In practice, elliptic curve ranks are often computed by methods using descent. (See Chapter X of Silverman [21] for an overview.) There is no guarantee, however, that a computation of this kind will terminate.

## 8. Complexity-theoretic aspects of Tunnell's criterion

For this section, we assume knowledge of standard complexity classes, as can be found in [17]. At the end we assume some facts about quantum computation, for which [14] is a good reference.

Unconditionally, the best we can say is that the set of congruent numbers is recursively enumerable. Indeed, one could enumerate congruent numbers by either systematically considering all Pythagorean triples, or by searching for non-torsion points on $E_n(\mathbb{Q})$, $n \geq 1$. (Since all square multiples of $n$ give $\mathbb{Q}$-isomorphic curves, the existence of such points is oblivious to square factors.) If Lang's conjecture [21, p. 235] is true, this search could be restricted to rational points whose $x$-coordinates have bit length $\leq n^{3+o(1)}$, as the discriminant of $E_n$ is $O(n^6)$. (Technically, Lang referred to canonical height, but there are explicit bounds relating this to naive height; see [22].) This would give an algorithm of cost $\exp(n^{3+o(1)})$ to test congruence.

Let us now revert to assuming BSD. First, the non-congruent squarefree numbers and their square multiples can also be enumerated, by testing Tunnell's criterion. Running both enumeration procedures in parallel, we would eventually classify every number as congruent or not. It is worth observing that any determination resulting from this procedure is unconditionally correct; we only need BSD to guarantee that no number escapes.

Under BSD, the set of congruent numbers also belongs to PSPACE. To prove this we can use Algorithm 1, together with any factoring algorithm that uses polynomial space (e.g. trial division).

One interesting subclass of PSPACE is $\mathrm{C}_=\mathrm{P}$, defined as follows [25]. A language (set of integers in binary) $L$ belongs to $\mathrm{C}_=\mathrm{P}$ when there is a nondeterministic polynomial-time Turing machine $M$ such that $n \in L$ precisely when the number of accepting paths for $n$ equals the number of rejecting paths. We can show that the set of squarefree congruent numbers belongs to $\mathrm{C}_=\mathrm{P}$, as follows. First, the set of squarefree numbers is in co-NP, which is a subclass of $\mathrm{C}_=\mathrm{P}$ [9, Cor. 4.11]. Second, $\mathrm{C}_=\mathrm{P}$ is closed under intersection, which is easy to prove by observing that GapP is closed under multiplication [9, Lemma 3.9]. By these observations, it is enough to show how $M$ works on squarefree inputs. We will show this for odd $n$, leaving the other case to the reader. On input $n$, guess integers $x, y, z$ with $|x|, |y| \leq n$ and $-n-1 \leq z \leq n$. (The extra $-1$ is there to balance even and odd values of $z$.) Then, evaluate $Q_1(x, y, z)$. Accept the input if $Q_1 = n$ and $z$ is even, or if $Q_1 \neq n$ and $z$ is odd. Reject otherwise.

There is an interesting connection between this class and quantum computation. Fenner, et al. [8] proved that the languages in $\mathrm{C}_=\mathrm{P}$ are exactly those whose complements are in NQP, the quantum analog of NP. This means, then, that there is a polynomial-time quantum algorithm with the following property: when applied to a squarefree integer $n$, the amplitude for the accepting output cancels to zero iff $n$ is congruent. Here we sketch a quantum circuit with this property, again considering only odd $n$. The basic idea is to adapt the Deutsch–Josza algorithm. First, choose $r$ so that the integers of absolute value $\leq \sqrt{n}$ are distinct mod $2^r$. It will suffice to take $r = (1/2)\log_2 n + O(1)$. We dedicate an $r$-qubit register to each of $x, y, z$, using 2's complement notation. (That is, a bit pattern with the leftmost bit equal to 1 represents a negative number.) Note that the number of even and odd $z$'s are automatically balanced. There is an additional qubit $b$, making $\nu = 3r + 1$ qubits in all. Let $f$ be the Boolean function computed in the last paragraph, i.e. $f(x, y, z) = 1$ iff $Q_1 = n$ and $z$ is even, or $Q_1 \neq n$ and $z$ is odd. Let $H$ be a $2 \times 2$ Hadamard matrix, and $U_f$ the unitary matrix that takes $|x, y, z, b\rangle$ to $|x, y, z, b \oplus f(x, y, z)\rangle$; a quantum circuit for $U_f$ can be built as indicated in [14, p. 158]. We begin the algorithm by preparing the state

$$|x, y, z, b\rangle = |0 \cdots 0, 0 \cdots 0, 0 \cdots 0, 1\rangle.$$

Then, apply $H^{\otimes\nu} U_f H^{\otimes\nu}$, observe all qubits, and accept iff $x = y = z = 0$. By the argument in [14, p. 35], the amplitude for this vanishes iff $f$ is balanced, that is, iff $n$ is congruent.

## References

[1] R.P. Brent, An improved Monte Carlo factorization algorithm. BIT, **20** (1980), 176–184.

[2] J. Cilleruelo and A. Córdoba, Lattice points on ellipses. Duke Math. J., **76** (1994), 741–750.

[3] G.E. Collins and R.G.K. Loos, The Jacobi symbol algorithm. SIGSAM Bull., **16** (1982), 12–16.

[4] L.E. Dickson, History of the Theory of Numbers, Three Volumes. Chelsea Publishing Co., New York, 1966.

[5] J.D. Dixon, Asymptotically fast factorization of integers. Math. Comp., **36** (1981), 255–260.

[6] N.D. Elkies, Curves $Dy^2 = x^3 - x$ of odd analytic rank. Algorithmic number theory (Sydney, 2002), Lecture Notes in Comput. Sci., **2369**, Springer-Verlag, Berlin, 2002, 244–251.

[7] P. Erdős, Remarks on number theory I. Mat. Lapok, **12** (1961), 10–17.

[8] S.A. Fenner, F. Green, S. Homer and R. Pruim, Determining acceptance possibility for a quantum computation is hard for the polynomial hierarchy. Proc. Royal Society, London (A), **455** (1999), 3953–3966.

[9] L. Fortnow, Counting complexity. Complexity Theory Retrospective II (L. Hemaspaandra and A. Selman, eds.), Lecture Notes in Comput. Sci., Springer-Verlag, Berlin, 1997, 81–107.

[10] C.F. Gauss, Disquisitiones Arithmeticae. Springer-Verlag, New York, 1986.

[11] L.K. Hua, Introduction to Number Theory. Springer-Verlag, Berlin, 1982.

[12] K. Ireland and M. Rosen, A Classical Introduction to Modern Number Theory, second ed. Graduate Texts in Mathematics, **84**, Springer-Verlag, New York, 1990.

[13] N. Koblitz, Introduction to Elliptic Curves and Modular Forms. Graduate Texts in Mathematics, **97**, Springer-Verlag, New York, 1984.

[14] M.A. Nielsen and I.L. Chuang, Quantum Computation and Quantum Information. Cambridge University Press, Cambridge, 2000.

[15] J.M. Pollard, Theorems on factorization and primality testing. Proc. Cambridge Philos. Soc., **76** (1974), 521–528.

[16] C. Pomerance, The quadratic sieve factoring algorithm. Advances in cryptology (Paris, 1984), Lecture Notes in Comput. Sci., **209**, Springer-Verlag, Berlin, 1985, 169–182.

[17] S. Rudich and A. Wigderson (eds.), Computational Complexity Theory. American Mathematical Society, Providence, 2004.

[18] R.J. Schoof, Quadratic fields and factorization. Computational methods in number theory, Part II, Math. Centre Tracts, **155**, Math. Centrum, Amsterdam, 1982, 235–286.

[19] D. Shanks, Five number-theoretic algorithms. Proceedings of the Second Manitoba Conference on Numerical Mathematics (Winnipeg), Congressus Numerantium, No. VII, Utilitas Math., 1973, 51–70.

[20] V. Shoup, NTL: A Library for Doing Number Theory. Available from the author's home page at New York University.

[21] J.H. Silverman, The Arithmetic of Elliptic Curves. Springer-Verlag, New York, 1986.

[22] N.P. Smart, The Algorithmic Resolution of Diophantine Equations. Cambridge U. Press, New York, 1998.

[23] M. Tompa, Probabilistic Factoring Algorithms can be made errorless. Technical Report 83-09-01, Univ. Washington, Dept. of Computer Science, 1983.

[24] J.B. Tunnell, A classical Diophantine problem and modular forms of weight 3/2. Invent. Math., **72** (1983), 323–334.

[25] K. Wagner, The complexity of combinatorial problems with succinct input representation. Acta Inform., **23** (1986), 325–356.