hexagon:2611.00086v2

Class Numbers of Imaginary Quadratic Orders in the Polynomial Hierarchy

Frank Vega

Information Physics Institute, 840 W 67th St, Hialeah, FL 33012, USA

Abstract

For $d \geq 3$ with $-d \equiv 0, 1 \pmod 4$, let $h(-d)$ be the number of classes of primitive positive definite binary quadratic forms of discriminant $-d$. We prove that $d \mapsto h(-d)$ is computable in deterministic polynomial time with an oracle for a language in the second level Σ_2^{P} of the polynomial hierarchy. The proof uses only reduction theory of binary quadratic forms, Lagrange's theorem and Hermite normal forms; it does not assume the generalized Riemann hypothesis, does not take a generating set of the class group as input, and uses no randomness. A Σ_2^{P} oracle finds a generating set of $O(\log d)$ forms, and an NP oracle then determines the lattice of relations, whose index is $h(-d)$. Consequently, if P = NP, class numbers of imaginary quadratic fields are computable in polynomial time. As an application, the number D_n of representations of n by the ternary form $8x^2 + 2y^2 + 16z^2$ in Tunnell's congruent number criterion equals $2h(-4n)$ for square-free $n \equiv 2 \pmod 8$ and vanishes for square-free $n \equiv 6 \pmod 8$; hence D_n lies in $\mathrm{FP}^{\Sigma_2^{\mathrm{P}}}$ on even square-free n, and an exact counting reduction from #SAT to D_n would collapse the polynomial hierarchy to Δ_3^{P}.

Keywords: class numbers, binary quadratic forms, polynomial hierarchy, counting complexity, Tunnell's theorem, congruent numbers, Hermite normal form, black-box groups
2020 MSC: 11Y40, 11R29, 68Q15, 11E16, 11E20

1. Introduction

For an integer $d \geq 3$ with $-d \equiv 0, 1 \pmod 4$, let $h(-d)$ be the number of $\mathrm{SL}_2(\mathbb{Z})$-classes of primitive positive definite binary quadratic forms of discriminant $-d$; when $-d$ is a fundamental discriminant, $h(-d)$ is the class number of $\mathbb{Q}(\sqrt{-d})$. Computing $h(-d)$ from the binary encoding of d is a classical problem of computational number theory. Shanks's baby-step giant-step method and its refinements compute it in time exponential in $\log d$ [1]; under the generalized Riemann hypothesis (GRH), Hafner and McCurley gave a subexponential algorithm [2]; and under GRH the class group is generated by forms of norm $O(\log^2 d)$ [3], so that Watrous's quantum algorithm for the order of a solvable black-box group [4] computes $h(-d)$ in quantum polynomial time.

The form class group $\mathrm{Cl}(-d)$ is an abelian group whose elements have unique polynomial-size encodings, the reduced forms, and whose operation is computable in polynomial time. For black-box groups given by a list of generators, Babai and Szemerédi [5] proved that membership lies in NP, and Babai [6] proved that order verification lies in $\mathrm{AM} \cap \mathrm{coAM}$. These results take a generating set as part of the input. No small generating set of $\mathrm{Cl}(-d)$ is known to be computable without GRH.

In this paper we prove that $d \mapsto h(-d)$ is computable in deterministic polynomial time with an oracle for a language in Σ_2^{P} (Theorem 3.1). The proof is elementary: it uses only reduction theory of binary quadratic forms, Lagrange's theorem, and Hermite normal forms of integer matrices. It does not assume GRH, does not take a generating set as input, and uses no interactive proofs or randomness. Its consequences are the following.

1. If P = NP, then class numbers of imaginary quadratic orders are computable in deterministic polynomial time (Corollary 3.2).

Email address: vega.frank@gmail.com (Frank Vega)

2. Let D_n be the number of representations of n by the ternary form $8x^2 + 2y^2 + 16z^2$ that appears in Tunnell's congruent number criterion [7]. For square-free $n \equiv 2 \pmod 8$ one has $D_n = 2h(-4n)$ (Proposition 4.2), and $D_n = 0$ for square-free $n \equiv 6 \pmod 8$ (Lemma 4.1). Hence $n \mapsto D_n$ on even square-free n lies in $\mathrm{FP}^{\Sigma_2^{\mathrm{P}}}$ (Corollary 4.3).
3. An exact counting reduction from #SAT to D_n, computable in polynomial time with an NP oracle, would imply $\#\mathrm{P} \subseteq \mathrm{FP}^{\Sigma_2^{\mathrm{P}}}$ and, by Toda's theorem [8], the collapse of the polynomial hierarchy to Δ_3^{P} (Theorem 4.5). Thus, unless the polynomial hierarchy collapses, Tunnell's count D_n cannot carry #P-hardness, although it is a #P function of n.

Section 2 fixes notation. Section 3 proves the main theorem and its first consequences. Section 4 applies it to Tunnell's count. Section 5 relates the results to prior work, and Section 6 concludes.

2. Preliminaries

For a positive integer m we write $\ell(m) = \lfloor \log_2 m \rfloor + 1$ for its bit length, and $\ell(0) = 1$; $|x|$ denotes the length of a string x, $|b|$ the absolute value of an integer b, and $|G|$ the order of a finite group G. The empty string is denoted by ε. $\Sigma_2^{\mathrm{P}} = \mathrm{NP}^{\mathrm{NP}}$ and $\Delta_3^{\mathrm{P}} = \mathrm{P}^{\Sigma_2^{\mathrm{P}}}$ are levels of the polynomial hierarchy PH; for a class C of languages, FP^C is the class of functions computable in deterministic polynomial time with an oracle for a language in C. If P = NP, then PH = P, so $\mathrm{FP}^{\mathrm{NP}} = \mathrm{FP}^{\Sigma_2^{\mathrm{P}}} = \mathrm{FP}$. A function $f : \{0,1\}^* \to \mathbb{N}$ is in #P if $f(x)$ is the number of strings y of length $p(|x|)$ accepted by a polynomial-time verifier $V(x, y)$, for some polynomial p [9, 10]; #SAT, the number of satisfying assignments $\#\varphi$ of a Boolean formula φ, is #P-complete under parsimonious reductions [11].

$H(d)$ denotes the Hurwitz class number, which counts all, not necessarily primitive, classes of positive definite binary forms of discriminant $-d$, weighted by $1/2$ and $1/3$ for classes of forms equivalent to multiples of $x^2 + y^2$ and $x^2 + xy + y^2$ respectively, and which is zero unless $-d \equiv 0, 1 \pmod 4$ [12]. For a fundamental discriminant $-d < -4$ one has $H(d) = h(-d)$.

Lemma 2.1 (Prefix search)**.** *Let $N \geq 0$ be an integer and $W \subseteq \{0,1\}^N$. Suppose that an oracle decides, for every binary string w of length at most N, whether some element of W has prefix w. Then $N + 1$ oracle queries suffice either to determine that $W = \emptyset$ or to output an element of W.*

Proof. Query $w = \varepsilon$; if the answer is negative, then $W = \emptyset$. Otherwise repeat N times: query $w0$, and replace w by $w0$ if the answer is positive and by $w1$ if it is negative. Throughout, some element of W has prefix w: this holds at the start, and an element of W of length N with prefix w, where $|w| < N$, has prefix $w0$ or prefix $w1$, so if no element of W has prefix $w0$ then some element has prefix $w1$. After N rounds $|w| = N$, and the only string of length N with prefix w is w itself, so $w \in W$. □

Lemma 2.2 (Triangular reduced basis)**.** *Let $t \geq 1$ and let $\Lambda \subseteq \mathbb{Z}^t$ be a subgroup of finite index h. Then Λ has a basis $b_1, \ldots, b_t$ such that, writing $b_i = (b_{i1}, \ldots, b_{it})$, one has $b_{ij} = 0$ for $j > i$, $b_{ii} = \delta_i \geq 1$, $0 \leq b_{ij} < \delta_j$ for $j < i$, and $\delta_1 \cdots \delta_t = h$. In particular every entry b_{ij} lies in $[0, h]$.*

Proof. This is the row-style, lower triangular Hermite normal form of Λ: the basis vectors are the rows of the matrix (b_{ij}), and each entry below the diagonal is reduced modulo the diagonal entry of its column. For $0 \leq i \leq t$ let Λ_i be the set of vectors of Λ whose coordinates of index greater than i vanish, so $\Lambda_0 = \{0\}$ and $\Lambda_t = \Lambda$. Since Λ has index h, $he_i \in \Lambda_i$ for the i-th unit vector e_i, so the set of i-th coordinates of the vectors of Λ_i is a nonzero subgroup $\delta_i\mathbb{Z}$ of $\mathbb{Z}$ with $\delta_i \geq 1$. Choose $b_i \in \Lambda_i$ with i-th coordinate δ_i. If $x \in \Lambda_i$, its i-th coordinate is $q\delta_i$ for some integer q, and $x - qb_i \in \Lambda_{i-1}$; by induction on i, $b_1, \ldots, b_i$ generate Λ_i, and they are linearly independent because the matrix with rows $b_1, \ldots, b_t$ is lower triangular with nonzero diagonal. Hence $b_1, \ldots, b_t$ is a basis of Λ, and $h = [\mathbb{Z}^t : \Lambda]$ equals the absolute value of the determinant of this matrix, which is $\delta_1 \cdots \delta_t$. Now, for $i = 2, \ldots, t$ and then for $j = i - 1, i - 2, \ldots, 1$, replace b_i by $b_i - qb_j$, where $q = \lfloor b_{ij}/\delta_j \rfloor$. This brings b_{ij} into $[0, \delta_j)$ and changes only the coordinates of b_i of index at most j, so the coordinates of index greater than j that have already been reduced, and the diagonal entry δ_i, are unchanged. The vectors still form a basis of Λ, because each step adds an integer multiple of one basis vector to another. Finally $1 \leq \delta_i \leq h$ for every i, since the δ_i are positive integers with product h, so $0 \leq b_{ij} \leq h$ for all i, j. □

3. Class Numbers in the Polynomial Hierarchy

We use the standard facts [1, Section 5.4] that every class of primitive positive definite forms of discriminant $-d$ contains exactly one reduced form (a, b, c), with $|b| \le a \le c$, $b \ge 0$ if $|b| = a$ or $a = c$, and $a \le \sqrt{d/3}$; that reduced forms have encoding length $O(\log d)$; and that composition followed by reduction, the group operation of the form class group $\mathrm{Cl}(-d)$, is computable in time polynomial in $\log d$. Consequently $\prod_i f_i^{v_i}$ can be computed by repeated squaring in time polynomial in $\log d$ and in the bit lengths of the exponents. The inverse of the class of a reduced form (a, b, c) is the class of $(a, -b, c)$, so negative exponents are covered as well. The identity 1_G of $\mathrm{Cl}(-d)$ is the class of the principal form, which is $(1, 0, d/4)$ if $d \equiv 0 \pmod 4$ and $(1, 1, (d+1)/4)$ if $d \equiv 3 \pmod 4$. We also use the polynomial-time computability of Hermite normal forms of integer matrices [13].

A reduced form (a, b, c) of discriminant $-d$ is determined by the pair (a, b), because $c = (b^2 + d)/(4a)$. With $A = \lfloor \sqrt{d/3} \rfloor$, the number of integer pairs (a, b) with $1 \le a \le \sqrt{d/3}$ and $|b| \le a$ is $\sum_{a=1}^{A}(2a+1) = A^2 + 2A$, and therefore

$$h(-d) \;\le\; A^2 + 2A \;\le\; \tfrac{d}{3} + 2\sqrt{d/3} \;\le\; d \qquad (d \ge 3). \tag{1}$$

The last inequality is equivalent to $\sqrt{d/3} \le d/3$, that is, to $d \ge 3$.

We fix the following encodings, with $\ell = \ell(d)$. A reduced form $f = (a, b, c)$ of discriminant $-d$ is encoded by the binary string $\mathrm{enc}(f)$ of length 2ℓ that consists of the ℓ-bit binary expansions, with leading zeros, of a and of $a + b$; this is well defined, for the following reason. Since $d \ge 3$, we have $\sqrt{d/3} \le d/3$, as noted after (1). Hence $1 \le a \le \sqrt{d/3} \le d/3 \le d$, and, because $-a \le b \le a$, $0 \le a + b \le 2a \le 2\sqrt{d/3} \le 2d/3 \le d$. Both integers therefore lie in $[0, d]$, and $d < 2^\ell$, so each of them has an ℓ-bit binary expansion. Given a string $u \in \{0,1\}^{2\ell}$, one decides in time polynomial in $\log d$ whether $u = \mathrm{enc}(f)$ for a reduced form f of discriminant $-d$, and computes f in that case: decode (a, b) from u, and check that $a \ge 1$, that $|b| \le a$, that $4a$ divides $b^2 + d$, that $c = (b^2 + d)/(4a)$ satisfies $a \le c$ and $\gcd(a, b, c) = 1$, and that $b \ge 0$ if $|b| = a$ or $a = c$. All integers involved have $O(\ell)$ bits, and the test consists of a constant number of comparisons, multiplications and divisions with remainder and of two greatest common divisor computations by the Euclidean algorithm, so it runs in time polynomial in ℓ. A vector $v = (v_1, \ldots, v_t) \in [-d, d]^t$ of integers is encoded by the binary string $\mathrm{enc}(v)$ of length $t(\ell + 1)$ that consists of the $(\ell + 1)$-bit binary expansions of the integers $v_i + d \in [0, 2d]$; a string of length $t(\ell + 1)$ is of the form $\mathrm{enc}(v)$ if and only if each of its t blocks represents an integer at most $2d$. A list of reduced forms or of vectors is encoded by the concatenation of the encodings of its members.

Theorem 3.1 (Class numbers in $\mathrm{FP}^{\Sigma_2^P}$)**.** *The function $d \mapsto h(-d)$, defined for $d \ge 3$ with $-d \equiv 0, 1 \pmod 4$, lies in $FP^{\Sigma_2^P}$. In particular, if $P = NP$, then $d \mapsto h(-d)$ is computable in deterministic polynomial time.*

Proof. Let $G = \mathrm{Cl}(-d)$, $\ell = \ell(d)$, and $h = |G|$, so that $h \le d$ by (1). The algorithm does not know h; it uses only the bound $h \le d$. If $h = 1$ the procedure below terminates immediately with the empty generating set and outputs 1.

Step 1: a small generating set. Let $S = (s_1, \ldots, s_t)$ be a list of reduced forms of discriminant $-d$, with $t \ge 0$. Since G is abelian and $s_i^h = 1_G$ by Lagrange's theorem, every element of $\langle S \rangle$ equals $\prod_i s_i^{v_i}$ for some v with $0 \le v_i < h \le d$, and every such product lies in $\langle S \rangle$. Hence a reduced form f satisfies $f \notin \langle S \rangle$ if and only if $\prod_i s_i^{v_i} \ne f$ for all $v \in \{0, \ldots, d\}^t$. Let L_1 be the set of all triples (d, S, w), where $S = (s_1, \ldots, s_t)$ is a list of reduced forms of discriminant $-d$ and w is a binary string of length at most 2ℓ, such that there exists a string $u \in \{0,1\}^{2\ell}$ that has prefix w, is of the form $u = \mathrm{enc}(f)$ for a reduced form f of discriminant $-d$, and satisfies

$$\forall v \in \{0, \ldots, d\}^t : \quad \prod_i s_i^{v_i} \ne f.$$

Thus w is a prefix of the encoding of the reduced form f, not of an exponent vector. The string u has length 2ℓ, and v is written with $t\ell$ bits; both lengths are bounded by the length of the input (d, S, w), in which S occupies $2t\ell$ bits. The universal quantifier ranges over $(d+1)^t$ vectors and is not checked by enumeration; it is delegated to an NP oracle, as follows. The statement $f \in \langle S \rangle$ has a short certificate, namely an exponent vector v with $\prod_i s_i^{v_i} = f$, so it is an NP statement; the statement $f \notin \langle S \rangle$ is its negation, and one query to an NP oracle decides it. The existential quantifier over f is the nondeterministic guess of the machine, and the universal quantifier over v is the negative answer of the oracle. Let M be the set of all triples (d, S, u), where $S = (s_1, \ldots, s_t)$ is a list of reduced forms of discriminant $-d$ and $u = \mathrm{enc}(f)$ for a reduced form f of discriminant $-d$, such that $\prod_i s_i^{v_i} = f$ for some $v \in \{0, \ldots, d\}^t$. Then $M \in \mathrm{NP}$: the

witness is v, written with $t\ell$ bits, which is at most the input length; the verifier checks $0 \le v_i \le d$, computes each $s_i^{v_i}$ by repeated squaring with at most 2ℓ compositions followed by reduction, multiplies the t results, and compares the reduced form obtained with f, in time polynomial in the input length. Consider the nondeterministic polynomial-time oracle machine that, on input (d, S, w), verifies with the decoding test that S is a list of reduced forms of discriminant $-d$, guesses $u \in \{0,1\}^{2\ell}$, rejects unless w is a prefix of u and $u = \mathrm{enc}(f)$ for a reduced form f of discriminant $-d$, makes the single query (d, S, u) to M, and accepts if and only if the answer is negative. It accepts (d, S, w) if and only if $(d, S, w) \in L_1$, so $L_1 \in \mathrm{NP}^{\mathrm{NP}} = \Sigma_2^{\mathrm{P}}$.

For fixed d and S, put $W_S = \{\mathrm{enc}(f) : f \text{ reduced of discriminant } -d,\ f \notin \langle S \rangle\} \subseteq \{0,1\}^{2\ell}$. By the definition of L_1, $(d, S, w) \in L_1$ if and only if some element of W_S has prefix w. Starting from $S = ()$, for which $\langle S \rangle = \{1_G\}$, we apply Lemma 2.1 with $N = 2\ell$ and $W = W_S$; if it outputs $\mathrm{enc}(f)$, we append f to S and repeat, and if it reports $W_S = \emptyset$, we stop. Each appended f lies outside $\langle S \rangle$, so $\langle S \rangle$ is a proper subgroup of $\langle S, f \rangle$ and $|\langle S, f \rangle| \ge 2|\langle S \rangle|$ by Lagrange's theorem. After t forms have been appended, $2^t \le |\langle S \rangle| \le h$, so $t \le \log_2 h \le \log_2 d$ throughout. When the loop stops, $W_S = \emptyset$, that is, every reduced form lies in $\langle S \rangle$, so $\langle S \rangle = G$. Step 1 makes at most $(\log_2 d + 1)(2\ell + 1)$ queries to L_1.

Step 2: the relation lattice. Let $\varphi : \mathbb{Z}^t \to G$, $\varphi(v) = \prod_i s_i^{v_i}$, and $\Lambda = \ker \varphi$. The map φ is a homomorphism because G is abelian, and it is surjective by Step 1, so $[\mathbb{Z}^t : \Lambda] = h$. Moreover $h\mathbb{Z}^t \subseteq \Lambda$, so Λ has rank t. By Lemma 2.2, Λ has a basis $b_1, \ldots, b_t$ all of whose entries lie in $[0, h] \subseteq [0, d]$. This basis is used only in the analysis of Step 3; the algorithm does not compute it in advance.

Step 3: lattice completion. For a finite list B of integer vectors, membership in the lattice $\mathcal{L}(B)$ spanned by B is decidable in polynomial time via Hermite normal forms [13]. Let L_2 be the set of all tuples (d, S, B, w), where $S = (s_1, \ldots, s_t)$ is a list of reduced forms of discriminant $-d$, B is a finite list of vectors of $[-d, d]^t$, and w is a binary string of length at most $t(\ell + 1)$, such that there exists a string $u \in \{0,1\}^{t(\ell+1)}$ that has prefix w, is of the form $u = \mathrm{enc}(v)$ for some $v \in [-d, d]^t$, and satisfies

$$\varphi(v) = 1_G \quad \text{and} \quad v \notin \mathcal{L}(B).$$

The witness u is no longer than the input, $\varphi(v)$ is computed by repeated squaring, and $v \notin \mathcal{L}(B)$ is tested in polynomial time, so $L_2 \in \mathrm{NP}$. For fixed d, S and B, put $W_B = \{\mathrm{enc}(v) : v \in \Lambda \cap [-d, d]^t,\ v \notin \mathcal{L}(B)\} \subseteq \{0,1\}^{t(\ell+1)}$; then $(d, S, B, w) \in L_2$ if and only if some element of W_B has prefix w. Starting from $B = ()$, we apply Lemma 2.1 with $N = t(\ell + 1)$ and $W = W_B$; if it outputs $\mathrm{enc}(v)$, we append v to B and repeat, and if it reports $W_B = \emptyset$, we stop. We prove three claims.

Claim (a): every round makes progress. By Lemma 2.1, as long as $W_B \neq \emptyset$ the search outputs an element of W_B, that is, a vector $v \in \Lambda \cap [-d, d]^t$ with $v \notin \mathcal{L}(B)$. Hence $\mathcal{L}(B) \subsetneq \mathcal{L}(B, v) \subseteq \Lambda$ after every round, whichever element of W_B the search returns. Write r for the rank of $\mathcal{L}(B)$. The rank of $\mathcal{L}(B, v)$ is either $r + 1$ or r; no other case occurs, and the argument below uses no further property of v.

Claim (b): the number of rounds is at most $t + t^2 \log_2(d\sqrt{t}) = O(\log^3 d)$. For a list B of vectors of $\Lambda \cap [-d, d]^t$ of rank r, let V_B be the rational subspace spanned by B and put

$$I(B) = [\Lambda \cap V_B : \mathcal{L}(B)].$$

Both $\mathcal{L}(B)$ and $\Lambda \cap V_B$ are lattices of rank r in V_B, and $\mathcal{L}(B) \subseteq \Lambda \cap V_B$, so $I(B)$ is a positive integer and $I(B) = \det \mathcal{L}(B) / \det(\Lambda \cap V_B)$, where the determinant of a lattice is the square root of the Gram determinant of any of its bases. The Gram determinant of a basis of a nonzero sublattice of $\mathbb{Z}^t$ is a positive integer, so $\det(\Lambda \cap V_B) \ge 1$ and $I(B) \le \det \mathcal{L}(B)$. Choose linearly independent vectors $b_1, \ldots, b_r$ among the members of B. Then $\mathcal{L}(b_1, \ldots, b_r)$ is a sublattice of $\mathcal{L}(B)$ of the same rank, so $\det \mathcal{L}(B) \le \det \mathcal{L}(b_1, \ldots, b_r) \le \prod_j \|b_j\| \le (d\sqrt{t})^r$ by Hadamard's inequality, because each $b_j \in [-d, d]^t$ has Euclidean norm at most $d\sqrt{t}$. Therefore

$$1 \ \le \ I(B) \ \le \ (d\sqrt{t})^r \tag{2}$$

at every moment of the loop. Consider a round that appends v without increasing the rank. Then $V_{(B,v)} = V_B$, and

$$I(B, v) = \frac{I(B)}{[\mathcal{L}(B, v) : \mathcal{L}(B)]} \le \frac{I(B)}{2},$$

because $\mathcal{L}(B) \subsetneq \mathcal{L}(B, v)$ by Claim (a). The rank never decreases. Hence, during the rounds in which the rank equals a fixed value r, the integer $I(B)$ starts at most at $(d\sqrt{t})^r$ by (2), is at least halved by each round that keeps the rank equal

to r, and never drops below 1; so at most $r \log_2(d\sqrt{t})$ rounds keep the rank equal to r. For $r = 0$ this number is 0, in accordance with the fact that every vector outside $\mathcal{L}(B) = \{0\}$ increases the rank. At most t rounds increase the rank. The total number of rounds is therefore at most

$$t + \sum_{r=1}^{t} r \log_2(d\sqrt{t}) \;\le\; t + t^2 \log_2(d\sqrt{t}) \;=\; O(\log^3 d),$$

since $t \le \log_2 d$. In particular the loop terminates, and each of its rounds makes $t(\ell+1)+1 = O(\log^2 d)$ queries to L_2.

Claim (c): when the loop stops, $\mathcal{L}(B) = \Lambda$. The loop stops when $W_B = \emptyset$, that is, when $\Lambda \cap [-d,d]^t \subseteq \mathcal{L}(B)$. By Step 2, every vector b_i of the basis of Λ given by Lemma 2.2 has all its entries in $[0,d]$, hence lies in $\Lambda \cap [-d,d]^t \subseteq \mathcal{L}(B)$. Therefore $\Lambda \subseteq \mathcal{L}(B)$, and $\mathcal{L}(B) \subseteq \Lambda$ by Claim (a), so $\mathcal{L}(B) = \Lambda$; in particular $\mathcal{L}(B)$ has rank t.

We compute a basis of $\mathcal{L}(B) = \Lambda$ in polynomial time from the list B [13] and output the absolute value of the determinant of the $t \times t$ matrix formed by this basis; this equals $[\mathbb{Z}^t : \Lambda] = h$ by Step 2.

Running time. By Step 1, S has $t \le \log_2 d$ members of 2ℓ bits each. By Claim (b), B has at most $t + t^2 \log_2(d\sqrt{t}) = O(\log^3 d)$ members, each a vector of $[-d,d]^t$ encoded with $t(\ell+1) = O(\log^2 d)$ bits. Hence every query has length polynomial in $\log d$, the number of queries is polynomial in $\log d$, and the computations between queries, namely the bookkeeping of prefix search and the final determinant, take time polynomial in $\log d$. The procedure is therefore a deterministic polynomial-time oracle Turing machine. The languages L_1 and L_2 are fixed: they do not depend on d, which is part of each query. Since $L_1 \in \Sigma_2^{\mathrm{P}}$ and $L_2 \in \mathrm{NP} \subseteq \Sigma_2^{\mathrm{P}}$, the single language $\{0x : x \in L_1\} \cup \{1x : x \in L_2\}$ is in Σ_2^{P} and serves as the oracle. Under P = NP both languages are in P. □

Corollary 3.2 (Class numbers under P = NP)**.** *If P = NP, then $d \mapsto h(-d)$ is computable in deterministic polynomial time; in particular, the class number of every imaginary quadratic field $\mathbb{Q}(\sqrt{-d})$ is computable in time polynomial in $\log d$.*

Proof. Under P = NP, $\Sigma_2^{\mathrm{P}} = \mathrm{P}$, and Theorem 3.1 applies. □

Remark 3.3 (The oracle queries)**.** Step 1 of the proof makes at most $(\log_2 d + 1)(2\ell(d)+1) = O(\log^2 d)$ queries to the Σ_2^{P} language L_1, and Step 3 makes $O(\log^5 d)$ queries to the NP language L_2, since each of its $O(\log^3 d)$ prefix searches makes $t(\ell(d)+1)+1 = O(\log^2 d)$ queries. The only Σ_2^{P} queries are those that certify that a list of reduced forms generates $\mathrm{Cl}(-d)$. Given a generating set of $O(\log d)$ forms, for instance the one provided under GRH by [3], Steps 2 and 3 alone compute $h(-d)$ in $\mathrm{FP}^{\mathrm{NP}}$. More generally, for every list S of at most $\log_2 d$ reduced forms of discriminant $-d$, generating or not, Steps 2 and 3 compute $|\langle S \rangle|$ in $\mathrm{FP}^{\mathrm{NP}}$: their analysis uses only that φ maps $\mathbb{Z}^t$ onto $\langle S \rangle$ and that $|\langle S \rangle| \le d$. The integer d enters L_1 and L_2 only as an upper bound for h: Steps 1 to 3 remain valid when d is replaced, in the exponent range $\{0,\dots,d\}^t$ and in the box $[-d,d]^t$, by any integer U with $h \le U \le d$ that is computable from d in polynomial time, for instance $U = A^2 + 2A$ from (1); this replaces $\ell(d)$ by $\ell(U)$ in the lengths of the exponent and relation vectors and leaves the bounds $O(\log^2 d)$ and $O(\log^5 d)$ unchanged. Theorem 3.1 is a statement in the oracle model: each query is counted as one step, and no bound is claimed for the time needed to answer the queries without an oracle.

Remark 3.4 (Comparison with the counting bound)**.** By the decoding test for enc, $h(-d)$ is the number of strings $u \in \{0,1\}^{2\ell(d)}$ such that $u = \mathrm{enc}(f)$ for a reduced form f of discriminant $-d$, so $d \mapsto h(-d)$ is a #P function and lies in $\mathrm{FP}^{\#\mathrm{P}}$. By Toda's theorem [8], $\Sigma_2^{\mathrm{P}} \subseteq \mathrm{P}^{\#\mathrm{P}}$, hence $\mathrm{FP}^{\Sigma_2^{\mathrm{P}}} \subseteq \mathrm{FP}^{\#\mathrm{P}}$, and Theorem 3.1 places $d \mapsto h(-d)$ in the smaller class.

4. Application to Tunnell's Count

For a positive integer n let

$$C_n = \#\{(x,y,z) \in \mathbb{Z}^3 : n = 8x^2 + 2y^2 + 64z^2\}, \qquad D_n = \#\{(x,y,z) \in \mathbb{Z}^3 : n = 8x^2 + 2y^2 + 16z^2\}.$$

Tunnell [7] proved that if an even square-free n is congruent, then $2C_n = D_n$, and that the converse holds if the Birch–Swinnerton-Dyer conjecture holds for the curve $y^2 = x^3 - n^2 x$. Both counts are #P functions of n: a witness is a triple (x,y,z) of bit length $O(\ell(n))$, checked in polynomial time. Let $\mathcal{N} = \{n \in \mathbb{N} : n \text{ square-free}, \; n \equiv 2 \pmod 8\}$.

Lemma 4.1 (Local vanishing)**.** *Let n be an even square-free positive integer. If $n \equiv 6 \pmod 8$, then $C_n = D_n = 0$; in particular Tunnell's identity $2C_n = D_n$ holds trivially.*

Proof. Write $n = 2m$ with m odd; $n \equiv 6 \pmod 8$ means $m \equiv 3 \pmod 4$. A representation $n = 8x^2 + 2y^2 + 16z^2$ gives $m = 4x^2 + y^2 + 8z^2 \equiv y^2 \pmod 4$, while $y^2 \equiv 0$ or $1 \pmod 4$, a contradiction. The same argument applies to $m = 4x^2 + y^2 + 32z^2$. □

Proposition 4.2 (Class-number formula)**.** *For every $n \in \mathcal{N}$,*

$$D_n = 2h(-4n),$$

where $h(-4n)$ is the class number of $\mathbb{Q}(\sqrt{-n})$. Consequently $D_n > 0$, $D_n = O(\sqrt{n}\log n)$, and for $n \in \mathcal{N}$ Tunnell's identity $2C_n = D_n$ is equivalent to $C_n = h(-4n)$.

Proof. Write $n = 2m$, so $m \equiv 1 \pmod 4$ is odd and square-free. Dividing $n = 8x^2 + 2y^2 + 16z^2$ by 2 shows that $D_n = r_{(1,4,8)}(m)$, the number of triples $(y, x, z) \in \mathbb{Z}^3$ with $m = y^2 + 4x^2 + 8z^2$. Let $r_3(n)$ be the number of triples $(a, b, c) \in \mathbb{Z}^3$ with $n = a^2 + b^2 + c^2$. By Gauss's three-square theorem, in the form stated in [12, Equation (1.1)], $r_3(n) = 12H(4n)$ for every positive integer $n \equiv 1, 2 \pmod 4$. A negative integer $D \equiv 0 \pmod 4$ is a fundamental discriminant if and only if $D/4$ is square-free and $D/4 \equiv 2$ or $3 \pmod 4$ [1]. Here $-4n/4 = -n$ is square-free and $-n \equiv 2 \pmod 4$, so $-4n$ is a fundamental discriminant, and $-4n \le -8 < -4$, so $H(4n) = h(-4n)$ and

$$r_3(n) = 12h(-4n). \tag{3}$$

We now prove that $r_3(n) = 6D_n$. Let $n = a^2 + b^2 + c^2$. A square is congruent to 0, 1 or 4 modulo 8, according to whether its root is divisible by 4, odd, or congruent to 2 modulo 4. A sum of three such residues is congruent to 2 modulo 8 only if two of them are 1 and the third is 0: an odd number of odd squares gives an odd sum, no odd square gives a sum in $\{0, 4\}$ modulo 8, and two odd squares together with a square congruent to 4 give 6. Hence exactly two of a, b, c are odd and the third is divisible by 4. Let R be the set of triples $(a, b, c) \in \mathbb{Z}^3$ with $n = a^2 + b^2 + c^2$, and for $k \in \{1, 2, 3\}$ let R_k be the subset of those whose k-th coordinate is even. The sets R_1, R_2, R_3 are pairwise disjoint with union R, and moving the k-th coordinate to the third position is a bijection $R_k \to R_3$, so $r_3(n) = |R| = 3|R_3|$. Writing $c = 4z$ shows that $|R_3| = T$, where T is the number of triples $(a, b, z) \in \mathbb{Z}^3$ with a, b odd and $n = a^2 + b^2 + 16z^2$; hence $r_3(n) = 3T$. The map $(a, b) \mapsto (u, v) = ((a + b)/2, (a - b)/2)$ is a bijection from the pairs of odd integers onto the pairs of integers of opposite parity, with inverse $(u, v) \mapsto (u + v, u - v)$, and $a^2 + b^2 = 2(u^2 + v^2)$. Hence T is the number of triples $(u, v, z) \in \mathbb{Z}^3$ with u, v of opposite parity and $m = u^2 + v^2 + 8z^2$. Each of these triples has either u odd and v even, or u even and v odd, and $(u, v, z) \mapsto (v, u, z)$ is a bijection between the two kinds, so exactly half of these triples have u odd and v even. Writing $u = y$, $v = 2x$ identifies the latter with the triples (y, x, z) such that y is odd and $m = y^2 + 4x^2 + 8z^2$. Since m is odd, every representation $m = y^2 + 4x^2 + 8z^2$ has y odd. Therefore $T = 2r_{(1,4,8)}(m) = 2D_n$ and $r_3(n) = 6D_n$, and (3) gives $D_n = 2h(-4n)$.

This agrees with the identity of Bringmann and Kane [12, Lemma 4.2], which states that $r_{\mathbf{a}}(m) = c_{\mathbf{a}}(m)(H(8m) - 2H(2m))$ for every positive integer m and every $\mathbf{a}$ in a list of six diagonal ternary forms that contains $\mathbf{a} = (1, 4, 8)$, with constants $c_{\mathbf{a}}(m)$ that depend only on a residue class of m and are tabulated in [12, Appendix B]. For our m, $-2m \equiv 2 \pmod 4$ is not a discriminant, so $H(2m) = 0$, and $H(8m) = H(4n) = h(-4n)$; the identity $D_n = 2h(-4n)$ proved above is the case $c_{(1,4,8)}(m) = 2$ of that lemma. Dirichlet's class number formula $h(-4n) = \pi^{-1}\sqrt{4n}\,L(1, \chi_{-4n})$ and the elementary bound $L(1, \chi_{-4n}) = O(\log n)$ give $D_n = O(\sqrt{n}\log n)$, and $h(-4n) \ge 1$ gives $D_n > 0$. The last assertion is immediate. □

Corollary 4.3 (Complexity of Tunnell's count)**.** *The function $n \mapsto D_n$ on even square-free n lies in $FP^{\Sigma_2^P}$. If $P = NP$, it is computable in deterministic polynomial time.*

Proof. The residue n mod 8 is read from the three low-order bits of n. An even square-free n is of the form $2m$ with m odd, so $n \equiv 2$ or $6 \pmod 8$. If $n \equiv 6 \pmod 8$ output 0 (Lemma 4.1); if $n \equiv 2 \pmod 8$ output $2h(-4n)$ (Proposition 4.2), computed by Theorem 3.1 with $d = 4n$. The domain itself is decidable with one NP query: the language $\{n : \exists e \ge 2,\ e^2 \mid n\}$ of non-square-free integers is in NP, because a witness e satisfies $\ell(e) \le \ell(n)$ and is verified by one multiplication and one division; n is square-free if and only if the oracle rejects n. □

Remark 4.4 (Factorization with an NP oracle)**.** The complete factorization of an integer $m \geq 2$ is also computable in $\mathrm{FP}^{\mathrm{NP}}$. The language $F = \{(m,k) : \exists e,\ 2 \leq e \leq k,\ e \mid m\}$ is in NP: the nondeterministic machine guesses e with $\ell(e) \leq \ell(k)$ and verifies $2 \leq e \leq k$ and $e \mid m$ by one division. For fixed $m \geq 2$ the predicate $(m,k) \in F$ is monotone in k, false for $k = 1$ and true for $k = m$, and the least k with $(m,k) \in F$ is the least divisor $p > 1$ of m, which is prime. Binary search on $k \in [2,m]$ finds p with at most $\ell(m)$ queries: keep an interval $[k_0,k_1]$ with $(m,k_0-1) \notin F$ and $(m,k_1) \in F$, initially $[2,m]$, query its midpoint k, and replace k_1 by k if $(m,k) \in F$ and k_0 by $k+1$ otherwise, until $k_0 = k_1 = p$. Replacing m by m/p and repeating, at most $\log_2 m$ times, yields all prime factors with multiplicity using at most $\ell(m)^2$ queries; m is square-free if and only if no prime occurs twice.

Theorem 4.5 (No exact counting reduction to D_n)**.** *Suppose there are functions R, α, β, computable in polynomial time with an NP oracle, such that for every Boolean formula φ, $R(\varphi)$ is an even square-free integer, $\alpha(\varphi) \geq 1$ and $\beta(\varphi)$ are integers, and $D_{R(\varphi)} = \alpha(\varphi)\,\#\varphi + \beta(\varphi)$. Then $\#P \subseteq FP^{\Sigma_2^P}$ and $PH = \Delta_3^P$.*

Proof. On input φ, compute $n = R(\varphi)$, $\alpha(\varphi)$ and $\beta(\varphi)$ with the NP oracle, compute D_n by Corollary 4.3, and output $(D_n - \beta(\varphi))/\alpha(\varphi) = \#\varphi$. Every oracle used is in Σ_2^{P}, so $\#\mathrm{SAT} \in \mathrm{FP}^{\Sigma_2^{\mathrm{P}}}$. Every $f \in \#\mathrm{P}$ reduces parsimoniously to #SAT, so $\#\mathrm{P} \subseteq \mathrm{FP}^{\Sigma_2^{\mathrm{P}}}$, and hence $\mathrm{P}^{\#\mathrm{P}} \subseteq \mathrm{P}^{\Sigma_2^{\mathrm{P}}} = \Delta_3^{\mathrm{P}}$. By Toda's theorem [8], $\mathrm{PH} \subseteq \mathrm{P}^{\#\mathrm{P}} \subseteq \Delta_3^{\mathrm{P}} \subseteq \mathrm{PH}$. □

Corollary 4.6 (Quantum computation of Tunnell's count)**.** *Assume GRH. Then $n \mapsto D_n$ on even square-free n is computable with bounded error in quantum polynomial time.*

Proof. Under GRH, $\mathrm{Cl}(-d)$ is generated by the classes of prime forms of norm less than $6\log^2 d$ [3], which can be listed in polynomial time. Reduced forms give a unique encoding of group elements and the group operation is polynomial-time, so Watrous's quantum algorithm for the order of a solvable black-box group [4] computes $h(-d)$ with bounded error in polynomial time. Square-freeness is decided with Shor's factoring algorithm [14], n mod 8 is read from the three low-order bits of n, and Lemma 4.1 and Proposition 4.2 give D_n. □

A numerical check of Lemma 4.1 and Proposition 4.2 is reported in Appendix Appendix A.

5. Relationship to Prior Work

The bounds for black-box groups recalled in the Introduction take a generating set as part of the input [5, 6]. Theorem 3.1 concerns the class group of a single discriminant, given only d; Step 1 of its proof finds a generating set of $O(\log d)$ forms with a Σ_2^{P} oracle, and Steps 2 and 3 compute the order of the group they generate with an NP oracle, deterministically. The subexponential bound of [2] is proved under GRH, and the algorithms that do not assume GRH run in time exponential in $\log d$ [1]. Theorem 3.1 is of a different kind: it is not a bound on the running time of an algorithm without oracles, but an upper bound on the complexity class of the function $d \mapsto h(-d)$, proved without GRH and without a generating set in the input. The upper bound that follows from counting reduced forms is $\mathrm{FP}^{\#\mathrm{P}}$, and Theorem 3.1 lowers it to $\mathrm{FP}^{\Sigma_2^{\mathrm{P}}}$ (Remark 3.4). For computation without oracles, Theorem 3.1 has two consequences: a polynomial-time algorithm if P = NP (Corollary 3.2), and, under GRH, the computation of $h(-d)$ in $\mathrm{FP}^{\mathrm{NP}}$ by Steps 2 and 3 applied to the generating set of [3] (Remark 3.3). Tunnell's criterion has been studied computationally on a large scale [15]. The class-number formula for D_n is derived in Proposition 4.2 from Gauss's three-square theorem, and it is an instance of the class-number identities for ternary forms of Bringmann and Kane [12].

6. Conclusion

The class number $h(-d)$ of every imaginary quadratic order is computable in $\mathrm{FP}^{\Sigma_2^{\mathrm{P}}}$, without GRH and without a given generating set, and hence in polynomial time if P = NP. For square-free $n \equiv 2 \pmod 8$, Tunnell's count D_n equals $2h(-4n)$, so it lies in $\mathrm{FP}^{\Sigma_2^{\mathrm{P}}}$ on all even square-free n, and an exact counting reduction from #SAT to D_n would collapse the polynomial hierarchy to Δ_3^{P}. Two questions remain open: whether $h(-d)$ is computable in $\mathrm{FP}^{\mathrm{NP}}$ without GRH; and whether the other Tunnell count C_n, which differs from $h(-4n)$ by a coefficient of a weight-3/2 cusp form, lies in the polynomial hierarchy. Concerning the first question, by Remark 3.3 the order $|\langle S\rangle|$, a divisor of $h(-d)$, is computable in $\mathrm{FP}^{\mathrm{NP}}$ for every list S of at most $\log_2 d$ reduced forms. The equality $\langle S\rangle = \mathrm{Cl}(-d)$ is the statement that for every reduced form f there is an exponent vector v with $\prod_i s_i^{v_i} = f$, a statement with a universal quantifier followed

by an existential one. The proof of Theorem 3.1 uses its Σ_2^{P} queries only to decide the negation of this statement, and it would give $h(-d) \in \mathrm{FP}^{\mathrm{NP}}$ if a list S with $\langle S \rangle = \mathrm{Cl}(-d)$ could be computed in $\mathrm{FP}^{\mathrm{NP}}$.

Acknowledgements

The author would like to thank Iris, Marilin, Sonia, Yoselin, and Arelis for their support.

Statements and Declarations

Declaration of interests

The author declares no competing interests.

Funding sources

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of generative AI in scientific writing

The author designed the results of this article, and used Claude Opus 5.5 to improve the readability of this manuscript and Claude Fable 5.2 to write the verification script `verify_tunnell.py`; the author ran all the computations. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of the publication.

Data availability

All numerical data used in this article are reported in Appendix Appendix A, and the script `verify_tunnell.py` is included in the submission.

Appendix A. Numerical Verification

The identities of Lemma 4.1 and Proposition 4.2 have been checked numerically, in exact integer arithmetic. Square-freeness of n is tested by trial division. The count D_n is computed by enumerating all pairs of integers (x, z) with $8x^2 + 16z^2 \le n$ and counting, for each pair, the integers y with $2y^2 = n - 8x^2 - 16z^2$, by means of an integer square root. The class number $h(-4n)$ is computed by enumerating all pairs of integers (a, b) with $1 \le a \le \sqrt{4n/3}$, $-a < b \le a$, b even and $4a \mid b^2 + 4n$, and counting those for which $c = (b^2 + 4n)/(4a)$ satisfies $c \ge a$, $\gcd(a, b, c) = 1$, and $b \ge 0$ if $a = c$; these are exactly the reduced forms of discriminant $-4n$. The equality $D_n = 2h(-4n)$ holds for all $n \in \mathcal{N}$ with $n \le 3 \cdot 10^4$ (3038 integers) and for 1500 pseudorandom $n \in \mathcal{N}$ with $n \le 2 \cdot 10^5$, and $D_n = 0$ for every square-free $n \equiv 6 \pmod 8$ with $n \le 2 \cdot 10^5$ (20267 integers). These computations are performed by the script `verify_tunnell.py`, which was written with Claude and run by the author, and which is available in the submission.

References

[1] H. Cohen, A Course in Computational Algebraic Number Theory, Vol. 138 of Graduate Texts in Mathematics, Springer-Verlag, Berlin, 1993. doi:10.1007/978-3-662-02945-9.

[2] J. L. Hafner, K. S. McCurley, A rigorous subexponential algorithm for computation of class groups, Journal of the American Mathematical Society 2 (4) (1989) 837–850. doi:10.1090/S0894-0347-1989-1002631-0.

[3] E. Bach, Explicit bounds for primality testing and related problems, Mathematics of Computation 55 (191) (1990) 355–380. doi:10.1090/S0025-5718-1990-1023756-8.

[4] J. Watrous, Quantum algorithms for solvable groups, in: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, ACM, New York, NY, USA, 2001, pp. 60–67. doi:10.1145/380752.380759.

[5] L. Babai, E. Szemerédi, On the complexity of matrix group problems I, in: 25th Annual Symposium on Foundations of Computer Science, IEEE, 1984, pp. 229–240. doi:10.1109/SFCS.1984.715919.

[6] L. Babai, Bounded round interactive proofs in finite groups, SIAM Journal on Discrete Mathematics 5 (1) (1992) 88–111. doi:10.1137/0405008.

[7] J. B. Tunnell, A classical Diophantine problem and modular forms of weight 3/2, Inventiones Mathematicae 72 (2) (1983) 323–334. doi:10.1007/BF01389327.
[8] S. Toda, PP is as hard as the polynomial-time hierarchy, SIAM Journal on Computing 20 (5) (1991) 865–877. doi:10.1137/0220053.
[9] L. G. Valiant, The complexity of computing the permanent, Theoretical Computer Science 8 (2) (1979) 189–201. doi:10.1016/0304-3975(79)90044-6.
[10] L. G. Valiant, The complexity of enumeration and reliability problems, SIAM Journal on Computing 8 (3) (1979) 410–421. doi:10.1137/0208032.
[11] S. Arora, B. Barak, Computational Complexity: A Modern Approach, Cambridge University Press, Cambridge, UK, 2009. doi:10.1017/CBO9780511804090.
[12] K. Bringmann, B. Kane, Class numbers and representations by ternary quadratic forms with congruence conditions, Mathematics of Computation 91 (333) (2022) 295–329. doi:10.1090/mcom/3648.
[13] R. Kannan, A. Bachem, Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix, SIAM Journal on Computing 8 (4) (1979) 499–507. doi:10.1137/0208040.
[14] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Journal on Computing 26 (5) (1997) 1484–1509. doi:10.1137/S0097539795293172.
[15] W. B. Hart, G. Tornaría, M. Watkins, Congruent number theta coefficients to 10^{12}, in: Algorithmic Number Theory (ANTS-IX), Vol. 6197 of Lecture Notes in Computer Science, Springer, Berlin, 2010, pp. 186–200. doi:10.1007/978-3-642-14518-6_17.