COMPUTING THE UNIT GROUP, CLASS GROUP AND COMPACT REPRESENTATIONS IN ALGEBRAIC FUNCTION FIELDS

KIRSTEN EISENTRÄGER AND SEAN HALLGREN

Abstract. Number fields and global function fields have many similar properties. Both have many applications to cryptography and coding theory, and the main computational problems for number fields, such as computing the ring of integers and computing the class group and the unit group, have analogues over function fields. The complexity of the number field problems has been studied extensively and these problems have been the source of some exponential speedups by quantum computation. In this paper we study the analogous problems in function fields. We show that there are efficient quantum algorithms for computing the unit group, the class group and for solving the principal ideal problem in function fields of arbitrary degree. We show that compact representations exist, which allows us to show that the principal ideal problem is in NP. Unlike the number field case, we are also able to show that these compact representations can be computed efficiently.

1. Introduction

Algebraic number theory is concerned with studying finite extensions L of $\mathbb{Q}$ which are called number fields, and investigating properties of the ring of algebraic integers $\mathcal{O}_L$, which is the integral closure of $\mathbb{Z}$ in L. Similarly, we can consider finite algebraic extensions K of $\mathbb{F}_q(t)$, where $\mathbb{F}_q(t)$ is the quotient field of the polynomial ring $\mathbb{F}_q[t]$. These fields are called function fields over finite fields or global function fields. It was noticed early on that the integers have many properties in common with $\mathbb{F}_q[t]$, and similarly, that number fields and global function fields have many similar properties. Often, a problem that is posed for number fields admits an analogous problem for global function fields, and the other way around. For example, the Riemann hypothesis for the classical Riemann zeta function $\zeta(s)$ is still open, while the analogue of this problem for function fields was solved by Weil.

The main computational problems for number fields include computing the ring of integers, the class group, the unit group and solving the principal ideal problem. These problems have been studied extensively and there are a large number of classical algorithms for computing with number fields. Applications include the Number Field Sieve, which is the fastest classical algorithm for factoring [LL93], and the Buchmann-Williams key-exchange system, which is based on the hardness of the principal ideal problem [BW89]. The recent push to make lattice-based cryptography more efficient has been using special lattices that come from number

This work was supported in part by the National Security Agency (NSA) under Army Research Office (ARO) contract number W911NF-08-1-0298. The first author was partially supported by National Science Foundation grant DMS-1056703 and a Sloan Research Fellowship. The second author was partially supported by National Science Foundation grant CCF-0747274.

fields [PR07, LPR10]. Error correcting codes have also been constructed using them [Gur03]. These computational problems for number fields are also a source of many of the known exponential speedups by quantum algorithms. There are efficient quantum algorithms for computing the unit group, class group, and solving the principal ideal problem in constant degree number fields [Hal05, SV05]. Some field extensions have also been computed using quantum algorithms [EH10]. This paper studies these computational problems over function fields.

Function fields have many applications in cryptography and coding theory. There are many cryptographic applications that use elliptic curves or Jacobians of curves of small genus defined over finite fields [CFA$^+$06]. These are based on the assumption that the discrete log problem is difficult to solve in the underlying group associated with these curves. Another way to state this is that the discrete log problem is assumed to be hard in the divisor class group of the function field of the curve. Error correcting codes have also been based on function fields [Gop88]. In a recent paper, Guruswami [Gur09] constructed codes where everything was efficient except computing the basis for the Riemann-Roch space of a certain divisor.

For number fields the problems listed above have been studied extensively and they appear to be computationally hard. For example, computing the ring of integers requires square-free factorization of integers. The best known classical algorithms for computing the unit group, the class group and solving the principal ideal problem are exponentially slower than factoring. On the other hand, computing the class group and unit group is in NP$\cap$coNP for arbitrary degree number fields [Thi95], while the quantum algorithms are only efficient for constant degree number fields. One apparent obstacle is that the only way known to compute with ideals of number fields requires a shortest vector problem in ideal lattices to be solved during computations in order to keep representation sizes small.

In this paper we examine these computational problems over function fields of arbitrary degree. For function fields, computing the ring of integers is computationally equivalent to factoring polynomials over a finite field, which can be done in (classical) polynomial time, so one might hope that much more can be done. In fact, even the analogue of the shortest vector problem has an efficient classical algorithm. But problems such as computing the divisor class group should be hard classically since they include as a special case the discrete log problem on an elliptic curve (a curve of genus one whose function field has degree two). For certain special classes of function fields (where the degree is two and the genus is large) there are subexponential algorithms for computing the class group, which make them less secure for cryptographic purposes: in [ADH94] the authors give a subexponential algorithm for computing the class group of a hyperelliptic curve of large genus, and [MST99] gives a subexponential probabilistic algorithm for computing the class group of a real quadratic congruence function field of large genus. In [Sch00] it is shown that various decision problems for quadratic congruence function fields of large genus are in NP$\cap$coNP. There are also some exponential algorithms known for more general function fields. Another important computational problem that only exists in the function field case is that of computing Riemann-Roch spaces.

In this paper we show that the principal ideal problem over function fields of arbitrary degree is in NP. To do this we show that compact multiplicative representations exist for elements in function fields. This answers a question of Smart [Sma98] and generalizes [Sch96], which showed the existence of compact representations for real quadratic congruence function fields (which have degree 2). Our work adapts work of Thiel who used compact representations in number fields and showed that the principal ideal problem, computing the class number, and computing compact representations of units are in NP∩coNP for number fields [Thi95]. Unlike the number field case, we also show that compact representations can be computed in (classical) polynomial-time for arbitrary degree function fields. The standard representation of an element, e.g. a unit, may take exponentially many bits to represent. Compact representations give a certain factored form of the element which only requires polynomial representation size.

Given this setup, we also show that there are efficient quantum algorithms for computing the unit group, class group, and solving the principal ideal problem in arbitrary degree function fields. This is in contrast to the number field case where currently only the constant degree case has quantum algorithms. These problems are solved by setting up abelian hidden subgroup problems.

One open question related to our work is whether the problems treated by Thiel are also in NP∩coNP for function fields. Compact representations played a key role in the number field case. One issue in the function field case is that it is not known how to deterministically compute generators for the class group efficiently.

Another open question is finding an efficient quantum algorithm for computing class field towers of function fields. Certain towers of function fields called Hilbert class field towers have applications to coding theory. When the tower is infinite one can construct asymptotically good sequences of codes [GS95, p. 212] from the fields in such towers. Infinite towers are known to exist [Sch92], but for applications of such codes in practice, an explicit construction of the fields in the tower is required. Class groups of certain subrings of the function fields in the tower appear as the Galois groups of the field extensions in the tower. Therefore, computing the class groups (and compact representations), as we do in this paper, is required to compute such towers, as it is in the number field case [EH10].

In order to set up our algorithms we need efficient algorithms for doing computations in the infrastructure of a function field. Fontein recently provided these and we prove that his algorithms in [Fon09, Fon11] are polynomial-time. To compute with the infrastructure it is necessary to efficiently compute the Riemann-Roch space of a divisor D. For this we use Hess's algorithm [Hes02] which is a relatively simple, self-contained algorithm. In the appendix we include a complexity analysis of his algorithm. For other references that analyze Hess's algorithm see also [Fon09] (under some additional assumptions) and [Die08]. We also give an algorithm to compute the infinite ideal B from the input divisor D and analyze its complexity. The algorithms above have been implemented, for example in MAGMA. The focus of this paper, however, is on the complexity analysis. Analyzing the Riemann-Roch algorithm addresses the missing piece for the codes in [Gur09] to be efficient.

One technical challenge in our work is adapting the algorithm to compute compact representations from the number field case [Thi95]. This requires showing that we can compute them without searching for minima in a region of exponential size, which is necessary in the number field case. We also analyze the Riemann-Roch space computation. This involves showing that we can efficiently compute the prime ideals of $\mathcal{O}_\infty$ which we use to compute the infinite ideal from the given representation of the divisor D. To do this we factor the ideal $(1/x)\mathcal{O}_\infty$ inside $\mathcal{O}_\infty$

by computing the radical of $(1/x)\mathcal{O}_\infty$ and analyze its complexity. This generalizes the ideal factorization algorithm for number fields [EH10].

There have been other approaches to study some of these problems over function fields. In [HI94] Huang and Ierardi gave a construction of the Riemann-Roch space that is polynomial time, assuming that all the singular points of the plane curve defining the function field are ordinary and defined over the base field. For another construction that used the Brill-Noether method see Volcheck [Vol94]. Recently, the authors learned that the (unpublished) Habilitation thesis by Diem [Die08] also studied Hess's algorithm. Kedlaya [Ked06] showed how to compute zeta functions of curves with a quantum algorithm. Part of this required computing the size of the divisor class group $\mathrm{Pic}^0(K)$, and he showed how to compute in the group efficiently. Our work, by contrast, requires the different representation using the infrastructure of Fontein [Fon11] in order to compute in the unit group and the class group, rather than only in the divisor class group $\mathrm{Pic}^0(K)$. The infrastructure also allows us to show the existence of compact representations.

Infrastructures have also been studied in [SW, FW], which give quantum algorithms for computing one-dimensional infrastructures and the period lattice of infrastructures of fixed dimension.

2. Background on algebraic function fields and divisors

Algebraic function fields over finite fields Let k be a finite field with $q = p^m$ elements for some prime $p > 0$. An *algebraic function field* K/k is an extension field $K \supseteq k$ such that K is a finite algebraic extension of $k(x)$ for some $x \in K$ which is transcendental over K. After replacing k with a finite extension we may assume that k is the *constant field* of K, i.e. that k is algebraically closed in K. By [Sti08, p. 144] such an algebraic function field is separably generated, i.e. there exist $x, y \in K$ such that $K = k(x, y)$. The function field K is then specified by a finite field k, an indeterminate x and the minimal polynomial $f \in k(x)[T]$ of y over $k(x)$. Throughout the paper, we assume that K is given to us as $K = k(x, y)$ with x, y as above and let $d := [K : k(x)]$.

A *valuation ring* of the function field K/k is a ring $\tilde{\mathcal{O}} \subseteq K$, such that $k \subsetneqq \tilde{\mathcal{O}} \subsetneqq K$, and for every $z \in K$ we have $z \in \tilde{\mathcal{O}}$ or $z^{-1} \in \tilde{\mathcal{O}}$. A valuation ring is a local ring, i.e. it has a unique maximal ideal [Sti08, p. 2]. A *place* of a function field K/k is defined to be the maximal ideal of some valuation ring of K/k. To each place $\mathfrak{p}$ of K, there is an *associated discrete valuation* $v_\mathfrak{p} : K^* \to \mathbb{Z}$, and there is a one-to-one correspondence between places of K/k and discrete valuations of K/k [Sti08, pp. 5f.]. Denote by $\mathcal{P}_K$ the set of all places of K. If $\mathfrak{p}$ is a place of K with corresponding valuation ring $\mathcal{O}_\mathfrak{p}$, we define the *degree of* $\mathfrak{p}$ to be the degree of the field extension of $\mathcal{O}_\mathfrak{p}/\mathfrak{p}$ over k, $\deg \mathfrak{p} = [\mathcal{O}_\mathfrak{p}/\mathfrak{p} : k]$. If F/K is an extension of algebraic function fields we say that a place $\mathfrak{P} \in \mathcal{P}_F$ *lies above a place* $\mathfrak{p} \in \mathcal{P}_K$ if $\mathfrak{p} \subseteq \mathfrak{P}$.

For the rational function field $k(x)$, the places are completely understood: the places of $k(x)$ correspond to the irreducible polynomials of $k[x]$, together with a "place at infinity", ∞.

Let v_∞ be the discrete valuation corresponding to the infinite place ∞ of the rational function field $k(x)$. Then v_∞ is defined via $v_\infty(f/g) = \deg g - \deg f$, for $f, g \in k[x]$. Let $\mathfrak{o}_\infty := \{a \in k(x) : v_\infty(a) \geq 0\}$. Then $\mathfrak{o}_\infty$ is the valuation ring associated to v_∞ and the unique maximal ideal of $\mathfrak{o}_\infty$ is generated by $1/x$. Let S denote the set of places of K above ∞. Let $\mathcal{O}_\infty := \{a \in K : v_\mathfrak{p}(a) \geq 0$ for all $\mathfrak{p} \in$

$S\}$. Then $\mathcal{O}_\infty$ is the integral closure of $\mathfrak{o}_\infty$ in K, and $\mathcal{O}_\infty$ is a free $\mathfrak{o}_\infty$-module of rank d. The ring $\mathcal{O}_\infty$ is a principal ideal domain whose prime ideals correspond to the elements in S.

Divisors on algebraic function fields A *divisor* on K is a formal sum

$$D = \sum_{\mathfrak{p}\in\mathcal{P}_K} n_\mathfrak{p}\mathfrak{p},$$

in which $n_\mathfrak{p} = 0$ for all but finitely many $\mathfrak{p}$. Let $\operatorname{Div}(K)$ denote the group of divisors on K. For a divisor D which is given as $D = \sum_{\mathfrak{p}\in\mathcal{P}_K} n_\mathfrak{p}\mathfrak{p}$, we define the degree of D to be $\deg D = \sum_{\mathfrak{p}\in\mathcal{P}_K} n_\mathfrak{p} \deg \mathfrak{p}$. The divisors of degree zero form a subgroup of $\operatorname{Div}(K)$, denoted by $\operatorname{Div}^0(K)$. For $f \in K^*$, the *divisor div*(f) *of* f is defined to be $\operatorname{div}(f) = \sum_{\mathfrak{p}\in\mathcal{P}_K} v_\mathfrak{p}(f)\mathfrak{p}$. The set of all divisors of the form $\operatorname{div}(f)$ form the group $\operatorname{Prin}(K)$ of principal divisors on K. Note that if D is a principal divisor then $\deg D = 0$. We define the divisor class group $\operatorname{Pic}^0(K)$ to be the quotient of the group of divisors of degree zero by the group of principal divisors, $\operatorname{Pic}^0(K) = \operatorname{Div}^0(K)/\operatorname{Prin}(K)$. It is a finite group.

A divisor $D = \sum_\mathfrak{p} n_\mathfrak{p}\mathfrak{p}$ is *effective* if $n_\mathfrak{p} \geq 0$ for all $\mathfrak{p}$; we write $D_1 \geq D_2$ to mean that $D_1 - D_2$ is effective. Every divisor D can be written uniquely as $D = D_+ - D_-$ with D^+, D^- effective divisors with disjoint support. We define the *height* $\operatorname{ht}(D)$ *of a divisor* D as $\operatorname{ht}(D) := \max\{\deg(D_+), \deg(D_-)\}$. For a divisor $D \in \operatorname{Div}(K)$ we define the *Riemann-Roch space of* D to be the set

$$L(D) := \{f \in K : \operatorname{div}(f) + D \geq 0\} \cup \{0\}.$$

The set $L(D)$ is a vector space over k, and we denote by $\ell(D)$ its dimension.

Fractional ideals Let $\mathcal{O}$ be the integral closure of $k[x]$ in K. Then $\mathcal{O}$ is a free $k[x]$-module of rank d. By [Chi89], Theorem 1, a $k[x]$-basis for $\mathcal{O}$ can be computed in time polynomial in d and $\log q$. If $S = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_{n+1}\}$ is the set of places above the infinite place ∞ of $k(x)$, then we also have $\mathcal{O} = \{a \in K : v_\mathfrak{p}(a) \geq 0\ \forall \mathfrak{p} \notin S\}$. Note that for any finite, non-empty set S of places of K one can find an $x \in K$ such that S is the set of infinite places above x. Throughout the paper we assume that $\deg \mathfrak{p}_{n+1} = 1$. This can always be achieved by passing to a finite extension of the constant field k.

A *fractional ideal* of $\mathcal{O}$ is a finitely generated $\mathcal{O}$-submodule of K. Since $\mathcal{O}$ is a Dedekind domain, the non-zero fractional ideals $\operatorname{Id}(\mathcal{O})$ of $\mathcal{O}$ form a (free) abelian group under multiplication. There is a natural homomorphism $\phi : \operatorname{Div}(K) \to \operatorname{Id}(\mathcal{O})$ defined by $\sum n_\mathfrak{p}\mathfrak{p} \mapsto \prod_{\mathfrak{p}\notin S}(\mathfrak{p} \cap \mathcal{O})^{-n_\mathfrak{p}}$. This map has a right inverse, namely $\operatorname{div} : \operatorname{Id}(\mathcal{O}) \to \operatorname{Div}(K)$ which sends a fractional ideal $B = \prod_{\mathfrak{p}\notin S}(\mathfrak{p} \cap \mathcal{O})^{n_\mathfrak{p}}$ to $\operatorname{div}(B) := -\sum_{\mathfrak{p}\notin S} n_\mathfrak{p}\mathfrak{p}$. Hence each divisor can be represented by a pair $(A, \sum t_i\mathfrak{p}_i)$ where A is a fractional ideal of $\mathcal{O}$ and $\{\mathfrak{p}_1, \ldots, \mathfrak{p}_{n+1}\}$ are the places in S, i.e. the primes above ∞. This is how we will represent divisors throughout the paper.

The class group $\operatorname{Cl}(\mathcal{O})$ of $\mathcal{O}$ is defined to be the group of fractional ideals of $\mathcal{O}$ modulo the principal fractional ideals of $\mathcal{O}$. The group $\operatorname{Cl}(\mathcal{O})$ is a finite abelian group, and the map $\phi : \operatorname{Div}(K) \to \operatorname{Id}(\mathcal{O})$ extends to a homomorphism

$$\phi : \operatorname{Pic}^0(K) \to \operatorname{Cl}(\mathcal{O}), \quad \left[\sum n_\mathfrak{p}\mathfrak{p}\right] \mapsto \left[\prod_{\mathfrak{p}\notin S}(\mathfrak{p} \cap \mathcal{O})^{-n_\mathfrak{p}}\right].$$

When $\deg \mathfrak{p}_{n+1} = 1$ this map fits into an exact sequence

$$0 \longrightarrow \mathrm{Ker} \longrightarrow \mathrm{Pic}^0(K) \xrightarrow{\phi} \mathrm{Cl}(\mathcal{O}) \longrightarrow 1.$$

Here Ker is the subgroup of $\mathrm{Pic}^0(K)$ which is generated by all degree zero divisors with support in S, so the first map $\mathrm{Ker} \to \mathrm{Pic}^0(K)$ is just the inclusion map. Since k is a finite field, Ker is finite by [Ros02, p. 243, Proposition 14.2].

3. Computing efficiently in the unit group

In this section we show how to compute efficiently classically in the unit group of $\mathcal{O}$. Recall that $S = \{\mathfrak{p}_1, \dots, \mathfrak{p}_{n+1}\}$ are the places of K above ∞ and that $\mathcal{O} = \{a \in K : v_\mathfrak{p}(a) \geq 0 \; \forall \mathfrak{p} \notin S\}$. Also, we assume that $\mathfrak{p}_{n+1}$ is a place of degree 1.

To compute in the unit group consider the map $\mathrm{val}_\infty : K^* \to \mathbb{Z}^n$, given by $\mathrm{val}_\infty(a) = (-v_{\mathfrak{p}_1}(a), \dots, -v_{\mathfrak{p}_n}(a))$. The image of $\mathcal{O}^*$ under val_∞ is a lattice Λ in $\mathbb{Z}^n$. By an analogue of Dirichlet's Unit Theorem for function fields, the *unit rank*, i.e. the rank of Λ, equals $n = \#S - 1$. Since units can have exponentially many bits in the standard representation, computing the unit group means to compute a basis of that lattice, or to compute compact representations for a fundamental set of units as in Definition 4.3. In Lemma 4.5 we show that the compact representation of an element can be computed from its valuation vector, so it follows that these two problems are polynomial time equivalent in function fields.

Fontein [Fon11] showed that it is possible to compute in a finite abelian group which he denotes $\mathrm{Rep}^{f*}(\mathcal{O})$ and which is isomorphic to $\mathbb{Z}^n/\Lambda$. We discuss his approach in the next section. We then show that these computations are efficient. From the group structure of $\mathbb{Z}^n/\Lambda$ we can obtain the basis for the lattice Λ.

3.1. Minima and reduced ideals in function fields.

We now give the definitions of minima and reduced ideals and define $\mathrm{Rep}^{f*}(\mathcal{O})$ (see Fon11). In the following, by an ideal of $\mathcal{O}$ we will always mean a *fractional* ideal of $\mathcal{O}$.

For each of the places $\mathfrak{p}_1, \dots, \mathfrak{p}_{n+1} \in S$ and their associated discrete valuations $v_{\mathfrak{p}_1}, \dots, v_{\mathfrak{p}_{n+1}}$ there is a corresponding *absolute value* $|\alpha|_i$ which is defined as $|\alpha|_i := q^{-v_{\mathfrak{p}_i}(\alpha) \deg \mathfrak{p}_i}$, $1 \leq i \leq n+1$. For an ideal A and integers $t_1, \dots, t_{n+1} \in \mathbb{Z}$ define

$$B(A, (t_1, \dots, t_{n+1})) := \{\alpha \in A : |\alpha|_i \leq q^{t_i \deg \mathfrak{p}_i}, \text{ for all } i,\ 1 \leq i \leq n+1\}.$$

This is a Riemann-Roch space, $B(A, (t_1, \dots, t_{n+1})) = L\left(\mathrm{div}(A) + \sum_{i=1}^{n+1} t_i \mathfrak{p}_i\right)$.

For an ideal A and $\alpha \in K^*$, let $B(A, \alpha) := B(A, (-v_{\mathfrak{p}_1}(\alpha), \dots, -v_{\mathfrak{p}_{n+1}}(\alpha)))$.

Definition 3.1. *(Minima and reduced ideals)*

(1) *Let A be an ideal of $\mathcal{O}$ and $\mu \in A$ non-zero. The element μ is a* minimum *of A if for every $\alpha \in B(A, \mu)$ either $\alpha = 0$ or $|\alpha|_i = |\mu|_i$ for $1 \leq i \leq n+1$.*

(2) *An ideal A is* reduced *if 1 is a minimum of A.*

Denote by $\mathrm{Red}(A)$ the set of reduced ideals of $\mathcal{O}$ which are in the same ideal class as A in $\mathrm{Cl}(\mathcal{O})$. There is a close connection between the set of minima of an ideal A and the set of reduced ideals equivalent to A. First, if μ is a minimum of A and $\epsilon \in \mathcal{O}^*$, then $\epsilon\mu$ is also a minimum of A. This action of $\mathcal{O}^*$ on the set of minima gives rise to a bijection

$$\{\text{minima of } A\}/\mathcal{O}^* \to \mathrm{Red}(A), \quad \mu\mathcal{O}^* \mapsto \frac{1}{\mu}A.$$

So every element of $\mathrm{Red}(A)$ is of the form $\frac{1}{\mu}A$ with μ a minimum of A. Next define a map from the set of reduced ideals equivalent to A to $\mathbb{Z}^n/\Lambda$ by defining

$$d : \mathrm{Red}(A) \to \mathbb{Z}^n/\Lambda, \quad \frac{1}{\mu}A \mapsto \mathrm{val}_\infty(\mu) + \Lambda.$$

This map is well defined since $\deg \mathfrak{p}_{n+1} = 1$ (see [Fon11, Corollary 5.3]) and injective [Fon11, Proposition 5.5]. Now we can define the group $\mathrm{Rep}^{f*}(\mathcal{O})$ that is isomorphic to $\mathbb{Z}^n/\Lambda$.

Definition 3.2. *Let A be an ideal of $\mathcal{O}$. An $f*$-representation is a tuple*

$$(I, (t_1, \dots, t_n)) \in \mathrm{Red}(A) \times \mathbb{Z}^n$$

such that $B(I, (t_1, \dots, t_n, 0)) = k$. Denote the set of all $f*$-representations in $\mathrm{Red}(A) \times \mathbb{Z}^n$ by $\mathrm{Rep}^{f*}(A)$.

When A and B are two ideals that are in the same ideal class in $\mathrm{Cl}(\mathcal{O})$, then clearly $\mathrm{Rep}^{f*}(A) = \mathrm{Rep}^{f*}(B)$. Let

$$\Phi_A : \mathrm{Rep}^{f*}(A) \to \mathbb{Z}^n/\Lambda$$

be defined by

$$\Phi_A\left(\frac{1}{\mu}A, t\right) = \mathrm{val}_\infty(\mu) + t + \Lambda.$$

Here $t = (t_1, \dots, t_n) \in \mathbb{Z}^n$. In [Fon11, Theorem 6.8] it is proved that this map is a bijection. In particular, $\mathrm{Rep}^{f*}(\mathcal{O})$ is isomorphic to $\mathbb{Z}^n/\Lambda$. So to each element (I, t) of $\mathrm{Rep}^{f*}(A)$, there is an associated point in $\mathbb{Z}^n/\Lambda$, and if $I = \frac{1}{\mu}A$, we say that (I, t) *represents* $\mathrm{val}_\infty(\mu) + t + \Lambda \in \mathbb{Z}^n/\Lambda$. Let $[A]$ be the set of ideals equivalent to A in the class group. It is possible to extend Φ_A to a well defined (no longer injective) map $\Phi_A : [A] \times \mathbb{Z}^n \to \mathbb{Z}^n/\Lambda$ by letting $\Phi_A\left(\frac{1}{\alpha}A, f\right) = \mathrm{val}_\infty(\alpha) + f + \Lambda$.

In [Fon11, Proposition 8.1] the following is shown:

Proposition 3.3. *Let $(A, (t_1, \dots, t_n)) \in \mathrm{Rep}^{f*}(B)$ for some ideal B. Then $div(A) \geq 0$ and $t_i \geq 0$ for $1 \leq i \leq n$. Moreover,*

$$0 \leq \deg div(A) + \sum_{i=1}^{n} t_i \deg \mathfrak{p}_i \leq g.$$

Here g denotes the genus of the function field.

We want to compute a basis for the n-dimensional lattice Λ. Since $\mathbb{Z}^n/\Lambda$ is isomorphic to $\mathrm{Rep}^{f*}(\mathcal{O})$, it is enough to obtain generators and relations for the finite group $\mathrm{Rep}^{f*}(\mathcal{O})$.

3.2. **Reduction and obtaining generators for** $\mathrm{Rep}^{f*}(\mathcal{O})$.

Let $\Phi := \Phi_{\mathcal{O}} : \mathrm{Rep}^{f*}(\mathcal{O}) \to \mathbb{Z}^n/\Lambda$ and its extension to $[\mathcal{O}] \times \mathbb{Z}^n \to \mathbb{Z}^n/\Lambda$ be the maps defined above. The group $\mathbb{Z}^n/\Lambda$ is generated by the standard basis vectors e_i $(1 \leq i \leq n)$, so in order to find generators for $\mathrm{Rep}^{f*}(\mathcal{O})$ we need to find elements $(\frac{1}{\mu_i}\mathcal{O}, f_i)$ such that $\Phi(\frac{1}{\mu_i}\mathcal{O}, f_i) = e_i + \Lambda$. To obtain such elements we consider the elements $(\mathcal{O}, e_i)$, $1 \leq i \leq n$. These elements are not in $\mathrm{Rep}^{f*}(\mathcal{O})$, but they do have the property that $\Phi(\mathcal{O}, e_i) = e_i + \Lambda$. So to obtain the right elements in $\mathrm{Rep}^{f*}(\mathcal{O})$ we reduce the elements $(\mathcal{O}, e_i)$ to elements $(\frac{1}{\mu}\mathcal{O}, f_i) \in \mathrm{Rep}^{f*}(\mathcal{O})$ with Algorithm 1 and use the fact that under Φ, the element $(\mathcal{O}, e_i)$ and its reduction have the same image (see Remark 3.5 below).

The general reduction algorithm that we are describing next works for $\mathrm{Rep}^{f*}(I)$ for any ideal I of $\mathcal{O}$.

Algorithm 1. Reduce
Input: Ideal A, vector $t = (t_1, \ldots, t_n) \in \mathbb{Z}^n$.
Output: Minimum μ of A, reduced ideal $\frac{1}{\mu}A$, vector $t - \mathrm{val}_\infty(\mu)$ where $(\frac{1}{\mu}A, t - \mathrm{val}_\infty(\mu)) \in \mathrm{Rep}^{f*}(A)$.

1. *Find the minimum ℓ in $[-\deg \mathrm{div}(A) - \sum_{i=1}^n t_i \deg \mathfrak{p}_i,\ \ g - \deg \mathrm{div}(A) - \sum_{i=1}^n t_i \deg \mathfrak{p}_i]$ such that $\dim B(A, (t_1, \ldots, t_n, \ell)) > 0$.*
2. *Set $u_1, \ldots, u_n = 0$. For each $1 \le i \le n$, increase u_i to find the largest value u_i with $\dim B(A, (t_1 - u_1, \ldots, t_n - u_n, \ell)) > 0$.*
3. *Let μ be a nonzero element of $B(A, (t_1 - u_1, \ldots, t_n - u_n, \ell))$. Output $(\mu, \frac{1}{\mu}A, (u_1, \ldots, u_n))$.*

Proposition 3.4. *Algorithm 1 is correct and returns $(\mu, \frac{1}{\mu}A, (u_1, \ldots, u_n))$ in time polynomial in d, $\log q$, $\mathrm{ht}(div(A))$ and $||t||_\infty$.*

Proof. Let ℓ be minimal such that $\dim B(A, (t_1, \ldots, t_n, \ell)) > 0$. By [Fon09, Theorem 4.4.3], $\ell \in [-\deg \mathrm{div}(A) - \sum_{i=1}^n t_i \deg \mathfrak{p}_i, g - \deg \mathrm{div}(A) - \sum_{i=1}^n t_i \deg \mathfrak{p}_i]$, so the first step of the algorithm requires at most g Riemann-Roch computations. Each of these computations $B(A, (t_1, \ldots, t_n, \ell)) = L(\mathrm{div}(A) + \sum_{i=1}^n t_i \cdot \mathfrak{p}_i + \ell \cdot \mathfrak{p}_{n+1})$ can be performed in time polynomial in d, $\log q$, $\mathrm{ht}(\mathrm{div}(A))$ and $||t||_\infty$ by Theorem B.7 because ℓ is at most a polynomial in g, $\mathrm{div}(A)$ and $||t||_\infty$, and g is a polynomial in d.

The second step computes the valuation that μ has in the third step. For coordinate i, there are at most t_i Riemann-Roch computations, so in total there are at most $n \max |t_i|$, which is polynomial in d and $||t||_\infty$ since $n \le d$. The correctness of steps 2 and 3 follows from the correctness proof of Algorithm 5.4.2 in [Fon09]. □

Remark 3.5. *Let A be an ideal of $\mathcal{O}$, and let $t = (t_1, \ldots, t_n) \in \mathbb{Z}^n$. Then $(A, (t_1, \ldots, t_n))$ represents the same point in $\mathbb{Z}^n/\Lambda$ as its reduction $(\frac{1}{\mu}A, t - \mathrm{val}_\infty(\mu)) \in \mathrm{Rep}^{f*}(A)$, since $\Phi_A(A, t) = t + \Lambda = \mathrm{val}_\infty(\mu) + (t - \mathrm{val}_\infty(\mu)) + \Lambda = \Phi_A(\frac{1}{\mu}A, t - \mathrm{val}_\infty(\mu))$.*

Denote by Reduce(A, e) the element of $\mathrm{Rep}^{f*}(A)$ that is computed by Algorithm 1. By the above discussion we have $\Phi_A(Reduce(A, e)) = e + \Lambda$, and if $e' = e + v$ with $v \in \Lambda$, then $\Phi_A(Reduce(A, e')) = e' + \Lambda = e + \Lambda$. Since $\Phi_A : \mathrm{Rep}^{f*}(A) \to \mathbb{Z}^n/\Lambda$ is injective this implies that Reduce$(A, e) =$ Reduce(A, e') whenever $e - e' \in \Lambda$.

Definition 3.6. *When $\alpha \in K$, the norm of α can be expressed uniquely as $N(\alpha) = f/h$, with $f, h \in k[x]$, coprime, h monic. We define $\mathrm{dg}(N(\alpha))$ to be $\mathrm{dg}(N(\alpha)) = \max\{\deg f, \deg h\}$.*

Remark 3.7. *When $A = \alpha\mathcal{O}$ then being polynomial in $\mathrm{ht}(div(A))$ is the same as being polynomial in $\mathrm{dg}\, N(\alpha)$ (see [Fon11, p. 28]).*

3.3. Composition and computing inverses in $\mathrm{Rep}^{f*}(\mathcal{O})$ and bounding the representation size of elements.

By [Fon11, Proposition 8.2], elements in $\mathrm{Rep}^{f*}(\mathcal{O})$ can be represented by $O(d^2 g \log q)$ bits. Here g denotes the genus of the function field, which is of size polynomial in d.

Composition of two elements $(A, f), (A', f')$ of $\mathrm{Rep}^{f*}(\mathcal{O})$ is done by multiplying the ideals, adding the two vectors, and then applying Algorithm 1 to $(AA', f + f')$.

To compute the inverse of $(A, f_1, \dots, f_n)$, compute the inverse A^{-1}, find ℓ minimal such that $\dim B(A^{-1}, (-f_1, \dots, -f_n, \ell)) > 0$ and then reduce using Algorithm 1 [Fon09, Proposition 4.3.4]. The ideal arithmetic in $\mathcal{O}$ is polynomial in $\log q$, d and $\mathrm{ht}(\mathrm{div}(A)), \mathrm{ht}(\mathrm{div}(A'))$ [Die08, Proposition 2.66, and Proposition 2.69(b)] and $\mathrm{ht}(\mathrm{div}(A))$ is of size polynomial in d and $\log q$ when $(A, f) \in \mathrm{Rep}^{f*}(\mathcal{O})$. Hence Proposition 3.4 implies that composition of two elements and computing inverses in $\mathrm{Rep}^{f*}(\mathcal{O})$ are both polynomial in $\log q$ and d.

4. Compact representations in global function fields

In this section we show how to efficiently compute compact representations of elements $\alpha \in K$ classically. This allows us to show that the principal ideal problem is in NP and to compute compact representations of units. We adapt the definitions and approach for number fields given in [Thi95, page 82] to the function field case. The sizes are adapted to match the parameters that are appropriate for number fields and that come from our algorithms. In the function field case we show that exponential search for minima is no longer necessary.

Definition 4.1. *For $\alpha \in K$ and $s \in \mathbb{Q}^n$ we say that α is* close to *s if*

$$||\,\mathrm{val}_\infty(\alpha) - s||_1 \leq n + g.$$

Definition 4.2. *A* multiplicative representation *of an element $\alpha \in K$ is a pair*

$$M = ((\beta_1, \dots, \beta_\ell), (e_1, \dots, e_\ell)),$$

where $\beta_i \in K$, $e_i \in \mathbb{Z}$, $\ell \in \mathbb{N}$, and such that $\alpha = \prod_{i=1}^{\ell} \beta_i^{e_i}$.

A binary multiplicative representation (BMR) *of an element $\alpha \in K$ is a multiplicative representation where $((\beta_1, \dots, \beta_i), (e_1, \dots, e_i))$ is a minimum of $\mathcal{O}$ for all $1 \leq i \leq \ell$ and $e_i = 2^{\ell - i}$ for all $1 \leq i \leq \ell$. Since the exponents are determined, a BMR can be represented as $(\beta_1, \dots, \beta_k)$.*

Definition 4.3. *A* compact representation *of $\alpha \in K$ is a pair $B = (\gamma, (\beta_1, \dots, \beta_\ell))$ where $\gamma \in K$ such that $\alpha = \frac{\gamma}{\beta}$ where β is a minimum of $\mathcal{O}$ represented by the BMR $(\beta_1, \dots, \beta_\ell)$, and*

$$\ell \leq \log(||\,\mathrm{val}_\infty(\alpha)||_\infty + g),$$
$$\mathit{size}(\gamma) \leq \mathit{poly}(\log q, d, \mathrm{dg}\, N(\alpha)), \text{ and}$$
$$\mathit{size}(\beta_i) \leq \mathit{poly}(\log q, d).$$

Here size denotes the number of bits to represent the element.

The bound on ℓ is chosen to handle the length of the generator after reducing $\alpha\mathcal{O}$, which is $\mathrm{val}_\infty(\gamma/\alpha)$. The factor γ comes from ideal reduction, so γ has size polynomial in d, $\log q$, and $\mathrm{dg}\, N(\alpha)$.

Claim 4.4. *Given a BMR $(\beta_1, \dots, \beta_\ell)$ of a minimum β of $\mathcal{O}$, the ideal $\frac{1}{\beta}\mathcal{O}$ can be efficiently computed.*

Proof. At the first step, $\frac{1}{\beta_1}\mathcal{O}$, which is a reduced ideal by the definition of BMR, can be efficiently computed. In general, let $\beta_i' = \prod_{j=1}^{i} \beta_j^{2^{i-j}}$. By the definition of BMR, β_i' is a minimum of $\mathcal{O}$ for all i. Given the reduced ideal $\frac{1}{\beta_i'}\mathcal{O}$, the reduced ideal $\frac{1}{\beta_{i+1}'}\mathcal{O} = \frac{1}{\beta_{i+1}(\beta_i')^2}\mathcal{O}$ can be efficiently computed by squaring $\frac{1}{\beta_i'}\mathcal{O}$ and multiplying by $1/\beta_{i+1}$. □

Algorithm 2. Compact Representation
Input: $\mathrm{val}_\infty(\alpha)$, $A = \alpha\mathcal{O}$.
Output: A compact representation of α.

```
(1) Call Reduce$(A, 0)$ to get a reduced ideal $\frac{1}{\gamma}A$ and element $\gamma \in K$.
(2) Let $(\beta_1, \dots, \beta_\ell)$=Close$(\mathcal{O},\ \mathrm{val}_\infty(\frac{\gamma}{\alpha}))$.
(3) Output $(\gamma, (\beta_1, \dots, \beta_\ell))$.
```

Lemma 4.5. *Compact Representation (Algorithm 2) returns a compact representation of $\alpha \in K$ in time polynomial in $\log q, d, \deg N(\alpha)$ and $\log(||\mathrm{val}_\infty(\alpha)||_\infty)$.*

Proof. By Proposition 3.4 the element γ in Step 1 can be computed with Algorithm 1 in time at most polynomial in d, $\log q$, and $\deg N(\alpha)$ by Proposition 3.4 and Remark 3.7. Therefore the size is bounded by the same amount. Also γ is a minimum of $A = \alpha\mathcal{O}$, so $\beta := \gamma/\alpha$ is a minimum of $\mathcal{O}$. By Corollary 4.9, $\mathrm{Close}(\mathcal{O}, \mathrm{val}_\infty(\frac{\gamma}{\alpha}))$ returns the BMR $(\beta_1, \dots, \beta_\ell)$ of the minimum $\beta = \gamma/\alpha$ of $\mathcal{O}$ (not just the BMR of a minimum close to γ/α). Hence the algorithm computes the compact representation $(\gamma, (\beta_1, \dots, \beta_\ell))$ of $\alpha = \gamma/\beta$. In Step 2 of the algorithm, Algorithm Close is called, which executes $\ell = \log(||\mathrm{val}_\infty(\gamma/\alpha)||_\infty) + 1$ calls of Algorithm Double. Each call of Double calls Reduce once on $(B, \lfloor u \rceil)$ where B is the square of a reduced ideal and hence small, and $||\lfloor u \rceil||_1 \leq 5n/2 + 2g$, where $n/2$ is from rounding, and we double $||t - \mathrm{val}_\infty(\mu)||_1 \leq n + g$ to get u. By Proposition 3.4 this is polynomial time. By the bound on ℓ, Algorithm 1 is polynomial in time d, $\log q$, $\deg N(\alpha)$ and $\log(||\mathrm{val}_\infty(\alpha)||_\infty)$. □

Algorithm 3. Close
Input: reduced ideal A, vector $s \in \mathbb{Q}^n$
Output: BMR $(\beta_1, \dots, \beta_\ell)$ of a minimum $\beta \in A$ which is close to s where $\ell = \log(||s||_\infty) + 1$

```
(1) Let $\beta_0 = 1$, $\ell = \log(||s||_\infty) + 1$ and $t = s/2^\ell$.
(2) For $k$ from 1 to $\ell$
    (a) $(\beta_1, \dots, \beta_k) :=$ Double$(A, t, (\beta_0, \beta_1, \dots, \beta_{k-1}))$
    (b) $t := 2t$
(3) Return $(\beta_1, \dots, \beta_\ell)$.
```

Proposition 4.6. *Close (Algorithm 3) is correct.*

Proof. This follows from the fact that in Step 1 of the algorithm $\beta_0 = 1$ is a minimum of A which is close to $t = s/2^\ell$ and Proposition 4.7. □

Algorithm 4. Double
Input: reduced ideal A, $t \in \mathbb{Q}^n$, BMR $(\beta_1, \dots, \beta_{k-1})$ of a minimum β of A which is close to t.
Output: BMR $(\beta_1, \dots, \beta_{k-1}, \beta_k)$ of a minimum of A which is close to $2t$ with β_k a minimum of $\frac{1}{\beta^2}A$ and β_k has size polynomial in d, $\log q$, $\mathrm{ht}(\mathrm{div} A)$.

```
(1) Let $B := \frac{1}{\beta^2}A$ and $u := 2t - \mathrm{val}_\infty(\beta^2)$.
(2) Reduce $(B, \lfloor u \rceil)$ to get a minimum $\beta_k$ of $B$ that is close to $u$.
(3) Return $(\beta_1, \dots, \beta_{k-1}, \beta_k)$. (This is the BMR of $\beta^2 \cdot \beta_k$.)
```

Proposition 4.7. *Double (Algorithm 4) is correct.*

Proof. First we show that there exists a minimum β_k of B such that $\mathrm{val}_\infty(\beta_k) - u$ has ℓ_1 norm $\leq n/2 + g$. When we reduce the pair $(B, \lfloor u \rceil)$ we get a pair $(\frac{1}{\beta_k}B, \lfloor u \rceil -$

$\mathrm{val}_\infty(\beta_k)) \in \mathrm{Rep}^{f*}(B)$. Let $t = (t_1, \dots, t_n) = \lfloor u \rceil - \mathrm{val}_\infty(\beta_k)$. By Proposition 3.3, we have $\sum_{i=1}^n t_i \deg \mathfrak{p}_i \leq g$, where g is the genus of the function field K. Rounding moved u by at most $n/2$ from the target point, so $\mathrm{val}_\infty(\beta_k) - u$ has ℓ_1-norm bounded by $n/2+g$. So there exists a minimum β_k of B that is close to $u = 2t - \mathrm{val}_\infty(\beta^2)$, and this minimum is computed in Step 3 of Double. Moreover, by Proposition 3.4, the minimum β_k has size polynomial in d, $\log q$, $\mathrm{ht}(\mathrm{div}(B))$, and $||u||_\infty$. Then, since β_k is close to $2t - \mathrm{val}_\infty(\beta^2)$, we have that $\beta^2\beta_k$ is close to $2t$, since $||2t - \mathrm{val}_\infty(\beta^2\beta_k)||_1 = ||(2t - (\mathrm{val}_\infty(\beta^2)) - \mathrm{val}_\infty(\beta_k)||_1$. □

In the next proposition we will show that if there is a minimum μ of A whose valuation vector equals $2t$, then Double returns the BMR of this minimum μ.

Proposition 4.8. *Let A be a reduced ideal. Suppose there is a minimum μ of A such that $\mathrm{val}_\infty(\mu) = 2t$. Then Double$(A, t, \beta = (\beta, \dots, \beta_{k-1}))$ returns the BMR $(\beta_1, \dots, \beta_k)$ of this minimum, i.e. $\mu = \beta^2 \cdot \beta_k$.*

Proof. In Step 3 of Double the algorithm reduces the pair $(A/(\beta^2), 2t - \mathrm{val}_\infty(\beta^2))$, where β is the given minimum of A which is close to t. Since $2t = \mathrm{val}_\infty(\mu)$, $2t$ has integer coordinates and it is not necessary to round $u = 2t - \mathrm{val}_\infty(\beta^2)$.

After reducing $(A/(\beta^2), 2t - v(\beta^2))$ we obtain an element $(A/(\beta_k\beta^2), 2t - \mathrm{val}_\infty(\beta^2) - \mathrm{val}_\infty(\beta_k))$ of $\mathrm{Rep}^{f*}(\mathcal{O})$, where β_k is a minimum of $A/(\beta^2)$. Since reduction produces a unique element in $\mathrm{Rep}^{f*}(\mathcal{O})$ and elements of $\mathrm{Rep}^{f*}(\mathcal{O})$ have unique representatives, this implies that β_k is uniquely determined (up to multiplication by an element of $\mathbb{F}_q^*$). Now since μ is a minimum of A, we have $(1/\mu \cdot A, 0) \in \mathrm{Rep}^{f*}(\mathcal{O})$. We also have that $\nu := \mu/(\beta^2)$ is a minimum of $A/(\beta^2)$. Then

$$\left(\frac{1}{\nu}A/(\beta^2), 2t - \mathrm{val}_\infty(\beta^2) - \mathrm{val}_\infty(\nu)\right) = (1/\mu \cdot A, 0) \in \mathrm{Rep}^{f*}(\mathcal{O}).$$

Hence we must have $\beta_k = \mu/(\beta^2)$, i.e. Double returns the BMR of $\mu = \beta_k\beta^2$. □

Corollary 4.9. *If the input in Close (Algorithm 3) is a reduced ideal A and a vector $s \in \mathbb{Q}^n$ such that $s = \mathrm{val}_\infty(\mu)$ for a minimum μ of A then Close outputs the BMR of this minimum μ of A.*

Proof. At the last step of the for-loop in Step 2 of Close, we have the BMR of a minimum β of A that is close to $s/2$ and the last call of Double produces the BMR of a minimum β' of A that is close to s. By Proposition 4.8, Double outputs the BMR of μ, so Algorithm Close returns the BMR of μ with $s = \mathrm{val}_\infty(\mu)$. □

Corollary 4.10. *The principal ideal problem is in NP.*

Proof. Given a function field and an ideal I of $\mathcal{O}$ represented in HNF, if the ideal is principal, then the proof is a compact representation $B = (\gamma, (\beta_1, ..., \beta_\ell))$ for α, where $I = \alpha\mathcal{O}$. By Definition 4.3, the compact representation B has size bounded by $\log(||\,\mathrm{val}_\infty(\alpha)||_\infty + g)$ and $\mathit{poly}(\log q, d, \mathrm{dg}\, N(\alpha))$. The field parameters are $\log q$, d, and g. By Remark 3.7, being polynomial in $\mathrm{dg}(N(\alpha))$ is the same as being polynomial in $\mathrm{ht}(\mathrm{div}(A))$, which is the size of the ideal $A = \alpha\mathcal{O}$. We have that $||\log \mathrm{val}_\infty(\alpha)||_\infty$ is bounded by Proposition 3.3 and Proposition 3.4.

The verifier must efficiently test whether $A = \frac{\gamma}{\beta}\mathcal{O}$, where $\beta = \prod \beta_i^{2^{n-i}}$. The verifier can efficiently compute the ideal as follows. By Claim 4.4, $\frac{1}{\beta}\mathcal{O}$ can be efficiently computed. Multiplication by γ is efficient. Finally, comparing the HNF of A and $\frac{\gamma}{\beta}\mathcal{O}$ is efficient since the representation of an ideal is unique. □

5. Quantum algorithms for the unit group, principal ideal problem and the class group

In this section we give efficient quantum algorithms for computing the unit group, solving the principal ideal problem and computing the class group. Recall from Section 3 that for the unit group and the principal ideal problem this means the objects are computed in the val_∞ embedding, and that compact representations can then be computed.

The basic approach is to show that these problems reduce to abelian hidden subgroup problems (HSP) which are known to have efficient quantum algorithms [CM01]. The class group case is slightly more general since the HSP instances will take values that are quantum states.

Theorem 5.1. *There is a polynomial-time quantum algorithm for computing the unit group of a function field.*

Proof. A hidden subgroup problem for the unit group can be defined by the function $g : \mathbb{Z}^n \to \mathrm{Rep}^{f*}(\mathcal{O})$ defined as $g(e) = \mathrm{Reduce}(\mathcal{O}, e)$. Here $\mathrm{Reduce}(\mathcal{O}, e)$ is the element of $\mathrm{Rep}^{f*}(\mathcal{O})$ which is computed by Algorithm 1 and it is polynomial-time computable by Proposition 3.4. By Remark 3.5, $\mathrm{Reduce}(\mathcal{O}, e) = \mathrm{Reduce}(\mathcal{O}, e + v)$ for any $v \in \Lambda$, so the function g is constant on cosets. Therefore g is also defined on $\mathbb{Z}^n/\Lambda$, and it gives a bijection between $\mathbb{Z}^n/\Lambda$ and $\mathrm{Rep}^{f*}(\mathcal{O})$, so it is also distinct on different cosets. Using the HSP instance g, a quantum algorithm can compute a basis for Λ efficiently. Compact representations can then be computed if desired. □

In the decision version of the principal ideal problem an ideal I of $\mathcal{O}$ is given in HNF and the problem is to decide if it is principal. If it is principal, then the search version of the problem is to compute a generator, i.e. compute an α such that $I = \alpha\mathcal{O}$. Since generators may take an exponential number of bits to represent in general, we only require computing $\mathrm{val}_\infty(\alpha)$. Knowing $\mathrm{val}_\infty(\alpha)$ and $\alpha\mathcal{O}$ determines α up to a nonzero multiple in the finite field k. So given an arbitrary ideal I that is given to us in HNF, the strategy is to solve the search problem and compute a candidate value for $\mathrm{val}_\infty(\alpha)$, and then to test whether $I = \alpha\mathcal{O}$ to see if the ideal is principal or not. A compact representation of α can then be computed from $\mathrm{val}_\infty(\alpha)$ and I using Algorithm 2.

Theorem 5.2. *There is a polynomial-time quantum algorithm for the principal ideal problem in a function field.*

Proof. Recall that for a vector $v \in \mathbb{Z}^n$, calling Algorithm 1 on $(\mathcal{O}, v)$ results in a pair $(I_v, f_v) \in \mathrm{Rep}^{f*}(\mathcal{O})$. Here I_v is a reduced ideal and f_v is a vector such that f_v has positive coordinates. If $\frac{1}{\mu}\mathcal{O} = I_v$ then $\mathrm{val}_\infty(\frac{1}{\mu}) + f_v = v$ by Remark 3.5.

To solve the principal ideal problem we do the following. Given any ideal I we first call Algorithm 1 on $(I, 0)$ to get a reduced ideal I_v. The reduction algorithm also computes γ such that $\frac{1}{\gamma}I = I_v$. Hence it suffices to solve the principal ideal problem for reduced ideals. If $I_v = \frac{1}{\mu}\mathcal{O}$ is reduced, then I_v represents the point $v + \Lambda \in \mathbb{Z}^n/\Lambda$ with $v = \mathrm{val}_\infty(\mu)$. By the above discussion, solving the principal ideal problem means computing v. First, by Theorem 5.1, a basis B of the unit group (under the embedding val_∞) can be computed efficiently with a quantum algorithm. A hidden subgroup problem can be set up as follows. By abuse of notation we denote by $\mathbb{Z}^n/B$ the quotient of $\mathbb{Z}^n$ by the lattice generated by the elements in B. Let $G = \mathbb{Z}_M \times \mathbb{Z}^n/B$, where $M = |\mathbb{Z}^n/B|$. Define $h : G \to \mathrm{Rep}^{f*}(K) = \bigcup_A \mathrm{Rep}^{f*}(A)$ by defining h to be the following algorithm. On input (a, b), it uses the composition operation in Section 3.3 and repeated doubling to compute a times the group element (this does reductions along the way, giving an element in $\mathrm{Rep}^{f*}(K)$), then composing the result with $(\mathcal{O}, -b)$ and reducing. When the ideal I is principal, then $h(a, b) = (I_{av-b}, f_{av-b})$. The hidden subgroup in this case is $H = \langle(1, v)\rangle$, and $h(H) = (\mathcal{O}, 0)$. A set of coset representatives for H is $\{(0, w) : w \in \mathbb{Z}^n/B\}$. Then $h((0, w) + n(1, v)) = h(n, w + nv) = (I_{-w}, f_{-w})$, and so the different values of w correspond to the set of elements in $\mathrm{Rep}^{f*}(\mathcal{O})$. So it is constant on cosets and distinct on different cosets. The function h can be computed efficiently using a small modification to Close (Algorithm 3). Therefore there is an efficient quantum algorithm for finding generators for H. Given an element $(n, nv) \in \mathbb{Z}_M \times \mathbb{Z}^n/B$ of H, it is easy to compute v. □

Theorem 5.3. *There is a polynomial-time quantum algorithm for computing the ideal class group of a function field.*

Proof. To compute the class group we also reduce to an abelian hidden subgroup problem where the function takes quantum states as values. Since it is not known how to compute unique representatives in the class group we instead create quantum states to represent each element, as a superposition over all elements of $\mathrm{Rep}^{f*}(J)$ for the ideal class of J. Let $g_1, \ldots, g_m$ be a set of generators for $\mathrm{Cl}(\mathcal{O})$, which can be computed by Appendix A. For an ideal J, let $|\phi_J\rangle = \sum_{(I,v)\in\mathrm{Rep}^{f*}(J)} |I, v\rangle$. Define $f : \mathbb{Z}^m \to \mathbb{C}^{|\mathrm{Pic}^0(K)|}$ by $f(e_1, \ldots, e_m) = |\phi_J\rangle$, where J is the ideal resulting from $\mathrm{Reduce}(g_1^{e_1} \cdots g_m^{e_m}, 0)$. The function f can be efficiently evaluated using the algorithm for the principal ideal problem as follows. Given $|e_1, \ldots, e_m\rangle$, compute $|e_1, \ldots, e_m, J\rangle$, where J is the ideal resulting from $\mathrm{Reduce}(g_1^{e_1} \cdots g_m^{e_m}, 0)$. The ideal in the last register, call it J, is now used to create the superposition over reduced ideals. Create $\sum_{v\in\mathbb{Z}^n/B} |J, v\rangle$, then $\sum_{v\in\mathbb{Z}^n/B} |J, v, (J_v, f_v)\rangle$ where (J_v, f_v) is the result of calling $\mathrm{Reduce}(J, v)$. Next use the principal ideal algorithm on $J \cdot J_v^{-1}$, which answers v, to create $\sum_{v\in\mathbb{Z}^n/B} |J, v, (J_v, f_v), v\rangle$. Next uncompute v in the second register using the fourth, then uncompute the fourth register by running the principal ideal algorithm backwards. Finally, uncompute J using $e_1, \ldots, e_m$. □

Appendix A. Computing generators for Cl($\mathcal{O}$)

As usual, let K be an algebraic function field over a finite field of constants $k = \mathbb{F}_q$. As discussed in Section 2, when $S = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_{n+1}\}$ is the set of places at infinity and $\deg \mathfrak{p}_{n+1} = 1$, we have a short exact sequence

$$0 \longrightarrow \mathrm{Ker} \longrightarrow \mathrm{Pic}^0(K) \longrightarrow \mathrm{Cl}(\mathcal{O}) \to 1$$

where the map from $\mathrm{Pic}^0(K) \to \mathrm{Cl}(\mathcal{O})$ is given as

$$\sum_{\mathfrak{p}\in\mathcal{P}_K} n_{\mathfrak{p}}\mathfrak{p} \mapsto \prod_{\mathfrak{p}\in\mathcal{P}_K - S} (\mathfrak{p} \cap \mathcal{O})^{-n_{\mathfrak{p}}}.$$

Given a function field K as above, there is a smooth projective geometrically irreducible curve C whose function field is K. Let g denote the genus of this curve.

In [Ked06] Kedlaya proves that for q with $q^{1/2} > 16g$ there exists a randomized algorithm that produces a generating set for $\mathrm{Pic}^0(K)$ in time polynomial in g and

$\log q$. The genus of the curve C does not change if we increase the size of the base field k. Hence by making the constant field larger, if necessary, we can achieve that $q^{1/2} > 16g$. From the above exact sequence it follows that the image of the generating set for $\mathrm{Pic}^0(K)$ under the map described above gives a generating set of $\mathrm{Cl}(\mathcal{O})$.

Appendix B. Computing Riemann-Roch spaces

In this section we analyze the complexity of computing the Riemann-Roch space $L(D) := \{f \in K : \mathrm{div}(f) + D \geq 0\} \cup \{0\}$. The input to the problem is a function field K and a divisor $D = (A, \sum_{i=1}^{n+1} t_i \mathfrak{p}_i)$ of K. The fractional ideal A of $\mathcal{O}$ is given in HNF relative to an $\mathcal{O}$ basis. The second part of D is given by a set of integers $\{t_i : 1 \leq i \leq n+1\}$ that determine the multiplicity of the infinite places, i.e. the places of $S = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_{n+1}\}$, in D.

We follow Hess's [Hes02] algorithm to compute the Riemann-Roch space. In [Hes02] Hess does not include any proofs for the complexity of his algorithm, so in this section we show that the Riemann-Roch space $L(D)$ can be computed in time polynomial in $d, \log q$ and $\mathrm{ht}(D)$. (For the definition of $\mathrm{ht}(D)$ see Section 2.) Hess's algorithm is a relatively simple, self-contained algorithm. We also investigate more closely the complexity of computing $\mathfrak{o}_\infty$-bases of the ideals we are working with.

The main idea in [Hes02] is that the Riemann-Roch space can be computed as the intersection of two ideals that come from the divisor D, where the two ideals are in the rings $\mathcal{O}$ and $\mathcal{O}_\infty$.

First we show that we can compute an $\mathfrak{o}_\infty$-basis for $\mathcal{O}_\infty$ in polynomial time.

Proposition B.1 ([Eis95], Proposition 4.13)**.** *Let $R \subset S$, and let U be a multiplicatively closed subset of R. If S' is the integral closure of R in S, then $S'[U^{-1}]$ is the integral closure of $R[U^{-1}]$ in $S[U^{-1}]$.*

Lemma B.2. *An $\mathfrak{o}_\infty$-basis for $\mathcal{O}_\infty$ can be computed in time poly in d and $\log q$.*

Proof. By Chistov ([Chi89]), Theorem 1, applied to $k[1/x]$, we can compute a basis $\beta_1, \ldots, \beta_d$ of the integral closure of $k[1/x]$ in K. By Proposition B.1, taking integral closures commutes with localization, so when we apply the proposition with $R = k[1/x]$, $S = K$ and U the complement of the prime ideal $(1/x)$ of R, we have $\mathfrak{o}_\infty = k[1/x][U^{-1}]$. Let S' be the integral closure of $k[1/x]$ in K. Then $\mathcal{O}_\infty = S'[U^{-1}]$, which implies that $\beta_1, \ldots, \beta_d$ is an $\mathfrak{o}_\infty$-basis for $\mathcal{O}_\infty$. □

Lemma B.3. *Let A be a fractional ideal of $\mathcal{O}$ given by a $k[x]$-basis, and let B be a fractional ideal of $\mathcal{O}_\infty$ given by an $\mathfrak{o}_\infty$-basis. There exist bases $a_1, \ldots, a_d$ of A and $b_1, \ldots, b_d$ of B and uniquely determined integers λ_i such that $x^{-\lambda_i} b_i = a_i$.*

Proof. Let $a'_1, \ldots, a'_d \in K$ be a $k[x]$-basis of A and $b'_1, \ldots, b'_d \in K$ a $\mathfrak{o}_\infty$-basis of B. Both of these are bases for the vector space $K/k(x)$. Let $M \in k(x)^{d \times d}$ be such that

$$(a'_1, \ldots, a'_d) = (b'_1, \ldots, b'_d)M.$$

By [Hes02, Corollary 4.3] there exists unimodular $T_1 \in \mathfrak{o}_\infty^{d \times d} \subset k[[x^{-1}]]^{d \times d}$ and unimodular $T_2 \in k[x]^{d \times d}$ such that $T_1 M T_2 = (x^{-\lambda_j} \delta_{ij})_{ij}$.

Let $(a_1, \ldots, a_d) = (a'_1, \ldots, a'_d)T_2$ and $(b_1, \ldots, b_d) = (b'_1, \ldots, b'_d)T_1^{-1}$. Then

$$(b_1, \ldots, b_n)T_1 M T_2 = (b'_1, \ldots, b'_d)MT_2 = (a'_1, \ldots, a'_d)T_2 = (a_1, \ldots, a_d).$$

□

Lemma B.4. *If $a_1, \dots, a_d$ and $b_1, \dots, b_d$ are bases as in Lemma B.3, then $A \cap B$ has k-basis $\{x^j a_i : 1 \le i \le d, 0 \le j \le \lambda_j\}$.*

Proof. Assume $\lambda \ge 0$. The elements $x^j a_i \in A$ for $j \ge 0$ since $x \in \mathcal{O}$ so we only have to show $x^j a_i \in B$ iff $0 \le j \le \lambda$. We have $a_i = x^{-\lambda_i} b_i \in B$ since $\frac{1}{x} \in \mathfrak{o}_\infty$, B is an $\mathfrak{o}_\infty$-module and $\lambda_i \ge 0$. Similarly, $x^j a_i = x^{j-\lambda_i} b_i \in B$ iff $j - \lambda_i \le 0$, i.e. for $j \le \lambda_i$. But $x^j a_i \in A$ iff $j \ge 0$, so $x^j a_i \in A \cap B$ for $0 \le j \le \lambda_i$.

To see that this set forms a k-basis note that $A \cap B = \bigcup_{i=1}^{d} (A \cap B \cap k(x) a_i)$, and a k-basis for $A \cap B$ is the union of the k-bases for $A \cap B \cap k(x) a_i$.

But $A \cap B \cap k(x) a_i$ (for i with $\lambda_i \ge 0$) equals $A \cap B \cap a_i k[x]$, so it suffices to determine which monomials $(x^j) a_i$ are in this intersection. By the above argument the only monomials in this intersection are $a_i, x a_i, \dots, x^{\lambda_i} a_i$, and these elements are clearly linearly independent over k, so they form a k-basis for $A \cap B \cap k(x) a_i$ (for i with $\lambda_i \ge 0$) . □

Lemma B.5. *The elements $a_1, \dots, a_d$ and the integers $\lambda_1, \dots, \lambda_d$ above can be computed in polynomial time.*

Proof. We will first show that the matrices M and T_2 above can be computed in polynomial-time. The lemma then follows from the fact that $(a_1, \dots, a_d) = (a'_1, \dots, a'_d) T_2$, and that the maximum degree of elements of the j-th column of $M T_2$ is equal to $-\lambda_j$([Fon09, p. 15], [Hes02, Corollary 4.3]). When elements in K are specified as polynomials in y, i.e., as $\sum_{i=0}^{n} a_i y^i$ for coefficients $a_i \in k(x)$, then writing a element $\alpha \in K$ in terms of a basis $\omega_1, \dots, \omega_n$ is a vector space transformation, with vector space generators $1, y, y^2, \dots, y^{n-1}$. The columns of the matrix $A \in k(x)^{n \times n}$ contain the coefficients of the polynomials $\omega_1, \dots, \omega_n$. Then solving the equation $Az = b$ over $k(x)$ for z gives the coefficients of b in terms of the basis. For M, this can be done for each column.

The matrix T_2 is computed using Paulus's polynomial-time algorithm [Pau98] by keeping track of the operations during the basis reduction. □

Algorithm 5. Ideal intersection for ideals in two different rings
Input: Function field K, $x \in K$; $k[x]$-basis $\omega_1, \dots, \omega_d$ of $\mathcal{O}$, $k[x]$-basis $a'_1, \dots, a'_d$ of the fractional ideal A of $\mathcal{O}$, $\mathfrak{o}_\infty$-basis $v_1, \dots, v_d$ of $\mathcal{O}_\infty$, $\mathfrak{o}_\infty$-basis $b'_1, \dots, b'_d$ of the fractional ideal B of $\mathcal{O}_\infty$.
Output: $(a_1, \dots, a_d; \lambda_1, \dots, \lambda_d)$ such that $\{x^j a_i : 1 \le i \le d, 0 \le j \le \lambda_i\}$ is k-basis of the k-vector space $A \cap B$.

(1) *Compute a matrix M such that $(b'_1, \dots, b'_d) M = (a'_1, \dots, a'_d)$.*
(2) *Do a basis reduction on M. Keep track of the operations and let $T_2 \in Gl_d(k[x])$ be the transformation. Let $-\lambda_i$ be the maximum degree in the ith column of the reduced matrix $M T_2$.*
(3) *Let $(a_1, \dots, a_d) = (a'_1, \dots, a'_d) T_2$.*
(4) *Return $(a_1, \dots, a_d; \lambda_1, \dots, \lambda_d)$.*

Proposition B.6. *Algorithm 5 computes $(a_1, \dots, a_d; \lambda_1, \dots, \lambda_d)$ such that $\{x^j a_i : 1 \le i \le d, 0 \le j \le \lambda_i\}$ is a k-basis of the k-vector space $A \cap B$ in polynomial time.*

Proof. The matrix M computed in Step 1 of the algorithm is exactly the matrix from Lemma B.3 that leads to the special basis for A: $(a_1, \dots, a_d) = (a'_1, \dots, a'_d) T_2$. By Lemma B.4 and its proof, $\{x^j a_i : 1 \le i \le d, 0 \le j \le \lambda_i\}$ where $-\lambda_j$ is the maximum column degree in the j-th column of MT_2 is a k-basis for the intersection $A \cap B$. By Lemma B.5, the a_i's and the λ_i's can be computed in poly time. □

Algorithm 6. Riemann-Roch Space
Input: Function field K, $k[x]$-basis $\omega_1, \ldots, \omega_d$ of $\mathcal{O}$, a divisor $D = (A, \sum_{i=1}^{n+1} t_i \mathfrak{p}_i)$ where A is a fractional ideal of $\mathcal{O}$ given in a $k[x]$-basis.
Output: $a_1, \ldots, a_d \in K$, $\lambda_1, \ldots, \lambda_d \in \mathbb{Z}$ such that $\{x^j a_i : 1 \leq i \leq d, 0 \leq j \leq \lambda_i\}$ is a basis of the Riemann-Roch space $L(D)$.

```
(1) Compute a $k[x]$-basis of $A^{-1}$.
(2) Compute an $\mathfrak{o}_\infty$-basis of $B := \Pi_{i=1}^{n+1} (\mathfrak{p}_i \cap \mathcal{O}_\infty)^{t_i} \subseteq \mathcal{O}_\infty$.
(3) Compute an $\mathfrak{o}_\infty$-basis of $B^{-1}$.
(4) Use Algorithm 5 to compute $A^{-1} \cap B^{-1}$.
(5) Return the $(a_1, \ldots, a_d; \lambda_1, \ldots, \lambda_d)$ computed by Algorithm 3.
```

Theorem B.7. *The above algorithm computes the Riemann-Roch space $L(D)$ in time polynomial in d, $\log q$, $\mathrm{ht}(D)$.*

Proof. Computing the inverse of a fractional ideal A of $\mathcal{O}$ can be done in time polynomial in $\log q, d$ and $\mathrm{ht}(\mathrm{div}(A))$ [Die08, Proposition 2.69(b)]. The ideals $\mathfrak{p}_i \cap \mathcal{O}_\infty$ in Step 2 are the prime ideals of $\mathcal{O}_\infty$ corresponding to the places in S. These can be computed in polynomial time with an algorithm similar to the one given for number fields in [EH10]. Hence we can compute an $\mathfrak{o}_\infty$-basis for the ideal B in Step 2 in polynomial time. The inverse of an ideal B in this ring can be computed efficiently as well. Finally, by Proposition B.6 above, a basis for the k-vector space $A^{-1} \cap B^{-1}$ can be computed in polynomial time. □

References

[ADH94] Leonard M. Adleman, Jonathan DeMarrais, and Ming-Deh Huang. A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields. In *Algorithmic number theory (Ithaca, NY, 1994)*, volume 877 of *Lecture Notes in Comput. Sci.*, pages 28–40. Springer, Berlin, 1994.

[BW89] Johannes A. Buchmann and Hugh C. Williams. A key exchange system based on real quadratic fields (extended abstract). In G. Brassard, editor, *Advances in Cryptology—CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 335–343. Springer-Verlag, 1990, 20–24 August 1989.

[CFA$^+$06] Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen, and Frederik Vercauteren, editors. *Handbook of elliptic and hyperelliptic curve cryptography.* Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton, FL, 2006.

[Chi89] A. L. Chistov. The complexity of the construction of the ring of integers of a global field. *Dokl. Akad. Nauk SSSR*, 306(5):1063–1067, 1989.

[CM01] Kevin K. H. Cheung and Michele Mosca. Decomposing finite abelian groups. *Quantum Info. Comput.*, 1:26–32, October 2001.

[Die08] Claus Diem. On arithmetic and the discrete logarithm problem in class groups of curves. Habilitationsschrift, Universität Leipzig. Available at http://www.math.uni-leipzig.de/~diem/preprints/habil.pdf, 2008.

[EH10] Kirsten Eisenträger and Sean Hallgren. Algorithms for ray class groups and Hilbert class fields. In *Symposium on Discrete Algorithms (SODA) 2010 Proceedings*, pages 471–483. Society for Industrial and Applied Mathematics (SIAM), 2010.

[Eis95] David Eisenbud. *Commutative algebra with a view toward algebraic geometry*, volume 150 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 1995.

[Fon09] Felix Fontein. *The infrastructure of a global field and baby step-giant step algorithms.* PhD thesis, Univ. Zürich, 2009.

[Fon11] Felix Fontein. The infrastructure of a global field of arbitrary unit rank. *Math. Comp.*, 80(276):2325–2357, 2011.

[FW] Felix Fontein and Pawel Wocjan. Quantum algorithm for computing the period lattice of an infrastructure. `arXiv:1111.1348 [quant-ph]`.

[Gop88] V. D. Goppa. *Geometry and codes*, volume 24 of *Mathematics and its Applications (Soviet Series)*. Kluwer Academic Publishers Group, Dordrecht, 1988. Translated from the Russian by N. G. Shartse.

[GS95] Arnaldo Garcia and Henning Stichtenoth. A tower of Artin-Schreier extensions of function fields attaining the Drinfeld-Vladut bound. *Invent. Math.*, 121(1):211–222, 1995.

[Gur03] Venkatesan Guruswami. Constructions of codes from number fields. *IEEE Trans. Inf. Th.*, 49(3):594–603, 2003.

[Gur09] Venkatesan Guruswami. Artin automorphisms, cyclotomic function fields, and folded list-decodable codes. In *STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 23–32, New York, NY, USA, 2009. ACM.

[Hal05] Sean Hallgren. Fast quantum algorithms for computing the unit group and class group of a number field. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 468–474, 2005.

[Hes02] F. Hess. Computing Riemann-Roch spaces in algebraic function fields and related topics. *J. Symbolic Comput.*, 33(4):425–445, 2002.

[HI94] Ming-Deh Huang and Doug Ierardi. Efficient algorithms for the Riemann-Roch problem and for addition in the Jacobian of a curve. *Journal of Symbolic Computation*, 18(6):519 – 539, 1994.

[Ked06] Kiran Kedlaya. Quantum computation of zeta functions of curves. *Computational Complexity*, 15(1):1–19, 2006.

[LL93] A.K. Lenstra and H.W. Lenstra, editors. *The Development of the Number Field Sieve*, volume 1544 of *Lecture Notes in Mathematics*. Springer–Verlag, 1993.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. *Advances in Cryptology –EUROCRYPT 2010*, pages 1–23, 2010.

[MST99] Volker Müller, Andreas Stein, and Christoph Thiel. Computing discrete logarithms in real quadratic congruence function fields of large genus. *Math. Comp.*, 68(226):807–822, 1999.

[Pau98] Sachar Paulus. Lattice basis reduction in function fields. *Algorithmic Number Theory*, pages 567–575, 1998.

[PR07] Chris Peikert and Alon Rosen. Lattices that admit logarithmic worst-case to average-case connection factors. In *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 478–487, New York, NY, USA, 2007. ACM Press.

[Ros02] Michael Rosen. *Number theory in function fields*, volume 210 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2002.

[Sch92] René Schoof. Algebraic curves over $\mathbf{F}_2$ with many rational points. *J. Number Theory*, 41(1):6–14, 1992.

[Sch96] R. Scheidler. Compact representation in real quadratic congruence function fields. In *Algorithmic number theory (Talence, 1996)*, volume 1122 of *Lecture Notes in Comput. Sci.*, pages 323–336. Springer, Berlin, 1996.

[Sch00] R. Scheidler. Decision problems in quadratic function fields of high genus. *J. Complexity*, 16(2):411–423, 2000.

[Sma98] Nigel P. Smart. Reduced ideals in function fields. Technical Report HPL-98-201, HP Laboratories Bristol, 1998.

[Sti08] Henning Stichtenoth. *Algebraic Function Fields and Codes.* Springer Publishing Company, Incorporated, 2008.

[SV05] Arthur Schmidt and Ulrich Vollmer. Polynomial time quantum algorithm for the computation of the unit group of a number field. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 475–480, 2005.

[SW] Pradeep Sarvepalli and Pawel Wocjan. Quantum algorithms for one-dimensional infrastructures. `arXiv:1106.6347 [quant-ph]`.

[Thi95] Christoph Thiel. *On the complexity of some problems in algorithmic algebraic number theory.* PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1995.

[Vol94] Emil Volcheck. Computing in the Jacobian of a plane algebraic curve. In *ANTS-I: Proceedings of the First International Symposium on Algorithmic Number Theory*, pages 221–233, London, UK, 1994. Springer-Verlag.

Department of Mathematics, Penn State University, University Park, PA 16802, USA.
E-mail address: `eisentra@math.psu.edu`

Dept. Computer Science & Engineering, Penn State University, University Park, PA 16802, USA.
E-mail address: `hallgren@cse.psu.edu`