COMPUTING RELATIONS IN DIVISOR CLASS GROUPS OF ALGEBRAIC CURVES OVER FINITE FIELDS

F. HESS

Abstract. We develop an efficient provably subexponential index calculus style algorithm for computing relation lattices of divisor classes in the divisor class groups of arbitrary algebraic curves over finite fields of large genus. We discuss applications such as computing discrete logarithms, S-units and the structure of S-class groups for S a finite set of places. We also describe an implementation and provide some tables with examples.

This is a draft version.

1. Introduction

Let F/k be the function field of a geometrically irreducible algebraic curve over the finite field k. The divisor class group $\mathcal{C}l(F/k)$ of F/k is defined to be the quotient group $\mathcal{D}(F/k)/\mathcal{P}(F/k)$ of the group of all divisors by the group of all principal divisors, and the divisor class group of degree zero divisors $\mathcal{C}l^0(F/k)$ is defined to be the subgroup of $\mathcal{C}l(F/k)$ generated by degree zero divisors. This latter group is a finite abelian group.

In this paper we are interested in devising an efficient general algorithm for solving the following problem for function fields of large genus: Given any divisors $D_1, \ldots, D_r$ of degree zero, find a basis for the relation lattice

$$\Lambda = \Big\{ (v_i)_i \in \mathbb{Z}^r \;\Big|\; \sum_{i=1}^{r} v_i [D_i] = 0 \Big\},$$

where $[D_i]$ denotes the class of D_i in $\mathcal{C}l(F/k)$.

Such an algorithm has a number of important applications. For example, if there is a $\lambda \in \mathbb{Z}$ such that $[D_2] = \lambda [D_1]$ we can use the algorithm to compute this λ, thereby solving an instance of the discrete logarithm problem. This is in principle of relevance to cryptography since there are cryptosystems based on the hardness of computing such discrete logarithms. If the classes of the D_i form a generating system of $\mathcal{C}l^0(F/k)$ then it is clear that $\mathcal{C}l^0(F/k) \cong \mathbb{Z}^r/\Lambda$. We can thus also use the algorithm to compute the structure of $\mathcal{C}l^0(F/k)$ in elementary divisor form, by doing so for $\mathbb{Z}^r/\Lambda$. There are further applications such as the computation of S-units and ideal class groups of rings of algebraic functions whose poles are restricted to a finite set S of places, and eventually the computation of class fields, which in turn has applications to the construction of algebraic curves with many rational points and coding theory.

Computing such relation lattices has attracted much interest, and the employed algorithms are so called index calculus methods. The case of hyperelliptic function fields has been considered in [1, 2, 6, 17, 22], and the case of arbitrary function fields in [5, 11]. A general framework for index calculus methods is given in [7]. This paper improves on a number of issues. First of all, we show that relation lattices in arbitrary function fields of large genus can be computed in provably subexponential

Key words and phrases. Algebraic curves over finite fields, global function fields, divisor class group, Jacobian, index calculus, discrete logarithm.

running time of exponent $1/2$. This was previously known only under the restriction that the class number $h(F/k)$, that is the cardinality of $\mathcal{C}l^0(F/k)$, is asymptotically approximately equal to q^g. We also present a way of generating sparse relations between the given divisor classes and elements of a factor basis, generalising the technique of the pure discrete logarithm strategy of [7]. This allows a much faster generation of the relations and also allows the use of sparse matrix techniques, which in turn leads to a considerable speed up in the overall algorithm and makes the running time not only rigorous but also efficient. Finally, we have implemented a variant of the algorithm in the computer algebra systems Kash [13] and Magma [3, 4] and provide some tables with examples.

Our main result is as follows. Let g denote the genus of F/k and q the number of elements in k. We consider a sequence of global function fields such that g tends to infinity, and view all other parameters such as q depending on g. The constants in the O- and o-notation used will be absolute positive constants in $\mathbb{R}$. We use the standard complexity function

$$L_x(a,b) = \exp\bigl(b \log(x)^a \log(\log(x))^{1-a}\bigr)$$

and let the constant $c_F = 1$ if $h(F/k)/q^g = L_{q^g}(1/2, o(1))$, and $c_F > 8$ otherwise. We require that $\log(q) = o(g \log(g))$ and assume that F/k is represented as the field of fractions of an absolutely irreducible plane affine curve of degree $O(g)$, that the number r of the D_i is polynomial in g and q and that the D_i are of the form $D_i = (D_i)_0 - (D_i)_\infty$ where $(D_i)_0$ and $(D_i)_\infty$ are effective divisors of degree $O(g)$. These assumptions do not pose a serious restriction, as explained in section 2.

Theorem 1. *The relation lattice Λ can be computed in expected running time*

$$L_{q^{2g}}\left(1/2, (2c_F + o(1))^{1/2}\right).$$

On heuristic grounds we expect that the term $2g$ in Theorem 1 can be replaced by g, for example for families of function fields which have a rational subfield $k(x)$ such that $[F : k(x)] = O(1)$ and consequently $c_F = 1$. Theorem 1 then improves on the discussion of [7] for solving discrete logarithms of hyperelliptic curves with respect to the required upper bound for $\log(q)$ and the constant b in the exponent when $\log(q)$ is large.

2. Preliminaries

Throughout the paper we will consider a global function field F/k over the exact constant field k of q elements and of genus g. We refer to [23] for background information on function fields, places, divisors and divisor classes. In the paper we will assume that algorithms for computing with elements, places and divisors of F/k and for computing Riemann-Roch spaces of divisors are available. A description of such algorithms can be found in [12], and implementations have been carried out by the author in Kash [13] and Magma [3, 4].

Any algebraic function field of one variable can be represented as the field of fractions of the coordinate ring of an absolutely irreducible plane affine curve of degree $O(g)$, and we assume throughout the paper that we are given such a representation of F/k. The running time of the above algorithms is then expected polynomial time in g, $\log(q)$ and the bit size of the involved objects. For the existence of the plane affine curve of degree $O(g)$ see Theorem 56 in the appendix. Furthermore, any divisor class $[D]$ of degree zero can be represented by a divisor D whose zero and pole divisors D_0 and D_∞ have degree $O(g)$ respectively, by the theorem of Riemann-Roch. This shows that the assumptions on F/k and D_i for Theorem 1 do not imply a loss of generality.

The paper is organised as follows. In section 3 we describe the basic algorithm to find relations and analyse its running time. This leads to a number of questions regarding smoothness probabilities, the generation of the divisor class group and the approximation of the class number, which are dealt with in the subsequent sections 4 to 6. In section 7 we combine the results of the previous sections and derive our main result. In the sections 8 to 10 we discuss immediate applications and our implementation, and give some tables with examples. In section 11 we gather some technical and supplementary material used in the previous sections, including a Brauer-Siegel type result with improved error estimation.

3. Relation search

Let $D_1, \dots, D_r$ be divisors of degree zero. A vector $v = (v_1, \dots, v_r) \in \mathbb{Z}^r$ is said to be a relation of the D_i in the divisor class group if $\sum_{i=1}^r v_i[D_i] = 0$. These relations form a sublattice Λ of $\mathbb{Z}^r$ such that $\mathbb{Z}^r/\Lambda$ is isomorphic to the subgroup of $\mathcal{C}l^0(F/k)$ generated by the $[D_i]$. Our fundamental goal is to compute a basis for Λ.

A possible deterministic strategy in the case $r = 1$ would be to check $v_1[D_1] = 0$ for all $1 \le v_1 \le (q^{1/2}+1)^{2g}$ since $h(F/k) \le (q^{1/2}+1)^{2g}$, or to use the more efficient baby-step-giant step method. The case $r > 1$ can also be handled deterministically. However, these methods yield running times exponential in g because $h(F/k)$ grows exponentially in g, for example by (54) or $h(F/k) \ge (q^{1/2}-1)^{2g}$. We remark that the upper and lower bounds on $h(F/k)$ given here are immediate consequences of the theorem of Hasse-Weil. We now describe a probabilistic method which yields a running time subexponential in g.

Let A be a divisor of degree $n \ge g$ and $S = \{P_1, \dots, P_s\}$ a set of places of F/k. We say that an effective divisor is S-smooth if its support is contained in S. The basic strategy is as follows. We pick for each $\nu = 1, 2, \dots$ in turn a random linear combination $B_\nu = \sum_{i=1}^r v_{i,\nu} D_i$, and choose a random effective divisor $E_\nu \in [B_\nu + A]$. By the theorem of Riemann-Roch, $\dim(B_\nu + A) > 0$ so that $[B_\nu + A]$ indeed contains effective divisors. If E_ν is S-smooth we compute $w_{j,\nu}$ such that $E_\nu = \sum_{j=1}^s w_{j,\nu} P_j$ and record the vectors $v_\nu = (v_{1,\nu}, \dots, v_{r,\nu})$ and $w_\nu = (w_{1,\nu}, \dots, w_{s,\nu})$. These steps are repeated until newly obtained vectors w_ν are highly likely to be linearly dependent of already computed vectors w_μ. Let $\sum_{\mu=1}^\nu \lambda_\mu w_\mu = 0$ be such a linear dependence over $\mathbb{Z}$. Then $\sum_{\mu=1}^\nu \lambda_\mu E_\mu = 0$ and $\sum_{\mu=1}^\nu \lambda_\mu = 0$ since the E_ν all have the same degree n. From this we get

$$\begin{aligned}\sum_{i=1}^r \left(\sum_{\mu=1}^\nu \lambda_\mu v_{i,\mu} \right) [D_i] &= \sum_{\mu=1}^\nu \lambda_\mu \left(\sum_{i=1}^r v_{i,\mu} [D_i] \right) = \sum_{\mu=1}^\nu \lambda_\mu [B_\mu] \\ &= \sum_{\mu=1}^\nu \lambda_\mu [E_\mu] - \sum_{\mu=1}^\nu \lambda_\mu [A] = \left(-\sum_{\mu=1}^\nu \lambda_\mu \right) [A] \\ &= 0.\end{aligned}$$

We have thus obtained a relation between the $[D_i]$ as desired.

We now give a more detailed description of the relation search. It varies slightly from the description above in order to enable the discussion of the running time and success probability in Lemma 3. All random choices are uniformly distributed.

Algorithm 2. *(Relation search)*

Input: *A divisor D_0 of degree one, the $D_1, \dots, D_r$ of degree zero and $t_1, t_2 \ge 1$.*
Output: *A submodule U of the relation lattice Λ.*

1. *Choose $v_1, \dots, v_r \in \mathbb{Z}$ with $0 \le v_i \le (q^{1/2}+1)^{4g}$ at random.*
2. *Compute a random effective divisor $E \in [nD_0 + \sum_{i=1}^r v_i D_i]$.*

3. *If E is S-smooth compute $E = \sum_{j=1}^{s} w_j P_j$. We obtain the vectors $(v_1, \ldots, v_r, w_1, \ldots, w_s)$.*
4. *Steps 1 to 3 are executed t_1 times.*
5. *Choose $v_1, \ldots, v_r \in \mathbb{Z}$ with $0 \leq v_i \leq (q^{1/2}+1)^{4g}$ at random.*
6. *Compute a random effective divisor $E \in [-P_\nu + (n + \deg(P_\nu))D_0 + \sum_{i=1}^{r} v_i D_i]$ for $P_\nu \in S$.*
7. *If E is S-smooth compute $E = \sum_{j=1}^{s} w_j P_j$. We obtain the vectors $(v_1, \ldots, v_r, w_1, \ldots, w_{\nu-1}, w_\nu + 1, w_{\nu+1}, \ldots, w_s)$.*
8. *Steps 5 to 7 are executed $\lceil t_2/s \rceil$ times for every ν with $1 \leq \nu \leq s$.*
9. *Let Λ_S be the module generated by all the obtained vectors. Define U to be the submodule of $\mathbb{Z}^r$ consisting of the first r coordinates of all vectors of Λ_S whose last s coordinates are zero. Return U.*

Again, all E have the same degree n so that it is clear that $U \subseteq \Lambda$, but it could be strictly smaller. We let $p_{S,n}$ denote the probability that a randomly chosen effective divisor of a randomly chosen divisor class of degree n is S-smooth, and we denote a lower bound for $p_{S,n}$ by $\widetilde{p}_{S,n}$.

Lemma 3. *For $g \to \infty$ assume $s \to \infty$, $\widetilde{p}_{S,n} \to 0$. Also assume that the $[D_i]$ form a generating system of $\mathcal{C}l^0(F/k)$ with $r = o((q^{1/2}+1)^{2g})$ and $r \geq 1$. If*

$$t_1 \geq 2\lfloor 4\widetilde{p}_{S,n}^{-1} \rfloor \big(8s \log_2(n+1) \log_2(s \log_2(n+1)) + 2r + (s+r) \log_2(r(q^{1/2}+1)^{8g} + (n+1)^2)\big)$$

and

$$t_2 \geq 2\lfloor 4\widetilde{p}_{S,n}^{-1} \rfloor \big(4s \log_2(s \log_2(n+1))\big)$$

then Algorithm 2 returns U such that $U = \Lambda$ with probability tending to 1.

Proof. We apply Lemma 57, (58) with $l = (q^{1/2}+1)^{4g}$, $s = r$, $\Lambda = \Lambda$ and $v + \Lambda$ corresponding to a prescribed divisor class. Then, and since the $[D_i]$ form a generating system of $\mathcal{C}l^0(F/k)$, the probability that a divisor class chosen in steps 2 and 6 is equal to the prescribed divisor class is $(1+o(1))/h(F/k)$. In other words, the divisor classes from steps 2 and 6 are essentially randomly and uniformly sampled from $\mathcal{C}l^0(F/k)$. Hence the probability that E will be S-smooth is $(1+o(1))p_{S,n}$ and certainly $\geq p_{S,n}/2$ for g large enough.

From Corollary 66 we obtain $8s \log_2(n+1) \log_2(s \log_2(n+1)))$ and $4s \log_2(s \log_2(n+1))$ vectors in steps 2 and 6 respectively after $4\lfloor 2\widetilde{p}_{S,n}^{-1} \rfloor \leq 2\lfloor 4\widetilde{p}_{S,n}^{-1} \rfloor$ times as many random choices of E with probability tending to 1, since $\widetilde{p}_{S,n} \leq p_{S,n}$ and $\widetilde{p}_{S,n} \to 0$, $s \to \infty$. Let us denote by Λ_S the lattice generated by these vectors, and let Λ_2 be the module obtained from Λ_S by omitting the first r coordinates. The generating system of Λ_2 obtained from the generating system of Λ_S by omitting the first r coordinates is of the form of the generating vectors in Lemma 64, since the w_i in step 4 and step 7 are chosen from a finite set of vectors of norm n with a certain probability distribution. According to Lemma 64 we thus have $\Lambda_2 = \mathbb{Z}^s$ with probability tending to 1.

Now Λ_S has large rank but not yet full rank and we will need to consider more relations by which to enlarge Λ_S. We investigate to what extend additional vectors found in step 3 yield random not yet known relations between the D_i in the divisor class group.

Let $v = (v_i)_i$ and $E \in [nD_0 + \sum_{i=1}^{r} v_i D_i]$ be an arbitrary S-smooth divisor as in step 2, of degree n. Write $E = \sum_{j=1}^{s} w_j P_j$ for $w = (w_j)_j$. Because $\Lambda_2 = \mathbb{Z}^s$ there exists at least one $v' \in \mathbb{Z}^r$ such that $(v', w) \in \Lambda_S$. For each such v' we have that $v - v' \in \Lambda$, by the discussion before Algorithm 2. We let ι denote a function which takes every v and E as above to some choice of v', which only depends on

E. Then $\iota(u+v,E)=\iota(v,E)$ for every $u\in\Lambda$ and $v-\iota(v,E)\in\Lambda$. Furthermore, the vectors chosen in step 1 are of the form $u+v$ with $u\in\Lambda$ and $v\in V$ where V is a fixed system of representatives of $\mathbb{Z}^r/\Lambda$ with coefficients in $[0,h(F/k)]$. In summary, considering the vector obtained in step 3 together with Λ_S we hence implicitly obtain relations between the $[D_i]$ of the form $u+v-\iota(u+v,E)=u-a$ with $a=\iota(v,E)-v$, in the case of smoothness. Note that if there are several possible v' for given v and E, we actually also have several relations $u-a$, but for the discussion below it is sufficient to consider only one relation $u-a$ per given v and E, as determined by ι.

Define $Q(l)=\mathbb{Z}^r\cap[0,l]^r$ and $h=(q^{1/2}+1)^{2g}\geq h(F/k)$. Let $W=\Lambda\cap Q(h^2-h)$ and U be a proper sublattice of Λ. For $u\in W$ and $v\in V$ we have $u+v\in Q(h^2)$. We are now interested to estimate the probability that a randomly chosen vector from $Q(h^2)$, as done in step 1, will be of the form $u+v$ for some $u\in W$ and $v\in V$, that then in step 2 an S-smooth $E\in[nD_0+\sum_{i=1}^r(u_i+v_i)D_i]$ is chosen where $u=(u_i)_i$, and that the resulting implicit relation $u-a$ as above lies in $\Lambda\backslash U$. We first note that $(u,v)\mapsto u+v$ is injective because of the definition of V. We have $|V|=h(F/k)$ by definition and $|W|=(1+o(1))|Q(h^2-h)|/h(F/k)=(1+o(1))|Q(h^2)|/h(F/k)$ because of $r/h\to 0$ and Lemma 57, (58). Hence $|W+V|=|W|\cdot|V|=(1+o(1))|Q(h^2)|$, and a randomly chosen vector in $Q(h^2)$ in step 1 will lie in $W+V$ with probability tending to 1. Furthermore, if the vector lies in $W+V$, the decomposition into u and v is unique because of the injectivity, so $u\in W$ and $v\in V$ are uniformly and independently distributed in this case. This implies two things. Firstly, the probability, that E is S-smooth in step 2 under the condition that the randomly chosen vector from $Q(h^2)$ is in $W+V$, is then $(1+o(1))p_{S,n}$, since E is chosen from the divisor class corresponding to v, which is itself uniform and random. Secondly, u and a are independent, since $a=\iota(v,E)-v$ depends only on v and E, which are independent of u. Now $u-a\in\Lambda\backslash U$ precisely if the class $u+U$ is not equal to $a+U$. Let z run over a set of representatives of Λ/U, which may be infinite. For the probabilities we get $\Pr(u-z\in U)=0$ for almost all z and $\Pr(u-z\in U)\leq 1/2$ for the finite rest, using Lemma 57, (59) with $v=0$ (there are only finitely many possible classes $u+U$ because W is finite, and $u\in W$ is chosen uniformly). From this and the independence of u and a we see $\Pr(u-a\in U)=\sum_z\Pr(u-z\in U)\Pr(a-z\in U)\leq 1/2$. We observe that if U does not have finite index in Λ then $\Pr(u-a\in\Lambda\cap\mathbb{Q}U)\leq 1/2$ by the same argument. In conclusion, in step 3 we implicitly obtain a relation $u-a\in\Lambda\backslash U$, or $u-a\in\Lambda\backslash\mathbb{Q}U$ for $(\Lambda:U)=\infty$, with probability at least $(1+o(1))p_{S,n}/2$ or $\geq p_{S,n}/4$ for g large enough, for any given proper sublattice U of Λ.

From $4\lfloor 4\widetilde{p}_{S,n}^{-1}\rfloor r$ random choices in $Q(h^2)$ in step 2 we get r linearly independent $u-a\in\Lambda$ with probability tending to 1, by the previous paragraph and from Corollary 66. Enlarging Λ_S by the corresponding vectors found in step 3 we obtain a finite index $(\mathbb{Z}^{r+s}:\Lambda_S)\leq(rh^4+(n+1)^2)^{(s+r)/2}$, by the Hadamard inequality and because these vectors have norm $\leq(rh^4+(n+1)^2)^{1/2}$. Since $\Lambda_2=\mathbb{Z}^s$ from before we now have that the sublattice U of Λ, consisting of vectors of Λ_S where the last s coordinates are zero, satisfies $(\Lambda:U)\leq(\mathbb{Z}^r:U)=(\mathbb{Z}^{r+s}:\Lambda_S)$. From $4\lfloor 4\widetilde{p}_{S,n}^{-1}\rfloor(s+r)/2\log_2(rh^4+(n+1)^2)$ random choices in $Q(h^2)$ in step 2 we get $(s+r)/2\log_2(rh^4+(n+1)^2)$ implicit relations $u-a\in\Lambda$ which when included in U decrease the index $(\Lambda:U)$ to 1, with probability tending to 1. Enlarging Λ_S by the corresponding vectors obtained in step 3 we thus have $(\mathbb{Z}^{r+s}:\Lambda_S)=h(F/k)$ and $\Lambda=U$ respectively. □

If we are given the divisors $D_1,\ldots,D_r$ and an additional divisor D of degree zero we may want to find all relations between the corresponding divisor classes. This can clearly be achieved by running Algorithm 2 with input $D_1,\ldots,D_r,D$. Let

Λ be the relation lattice of the $[D_i]$ and Λ_D the relation lattice of the $[D_i]$ and D. Since the $[D_i]$ generate $\mathcal{C}l^0(F/k)$ it follows that there are $\lambda_1, \ldots, \lambda_r \in \mathbb{Z}$ such that $[D] = \sum_i \lambda_i [D_i]$. Then Λ_D is generated by the vector $(\lambda_1, \ldots, \lambda_r, 1)$ and by the elements of Λ with a zero coordinate added.

If we want to compute relations between the $[D_i]$ and many divisors D it is more efficient to run Algorithm 2 once and then use the following algorithm to compute $\lambda_1, \ldots, \lambda_r$ for every D.

Algorithm 4. *(Additional relation search)*

Input: *Divisors $D_1, \ldots, D_r$ and D of degree zero, Λ_S as computed in Algorithm 2 such that the module Λ_2 obtained by omitting the first r coordinates is equal to $\mathbb{Z}^s$.*

Output: *Integers $\lambda_1, \ldots, \lambda_r$ such that $[D] = \sum_i \lambda_i [D_i]$.*

1. *Choose $v_1, \ldots, v_r \in \mathbb{Z}$ with $0 \leq v_i \leq (q^{1/2}+1)^{4g}$ at random.*
2. *Compute a random effective divisor $E \in [D + nD_0 + \sum_{i=1}^r v_i D_i]$.*
3. *If E is not S-smooth go to step 1.*
4. *Compute $E = \sum_{j=1}^s w_j P_j$. We obtain the vector $u = (v_1, \ldots, v_r, w_1, \ldots, w_s)$.*
5. *Compute a vector $\lambda = (\lambda_1, \ldots, \lambda_r, 0, \ldots, 0)$ such that $\lambda \in -u + \Lambda_S$.*
6. *Return $\lambda_1, \ldots, \lambda_r$.*

Lemma 5. *Algorithm 2 computes $\lambda_1, \ldots, \lambda_r$ under the assumptions of Lemma 3 in expected time $(1+o(1))/p_{S,n}$.*

Proof. As in the proof of Lemma 3 we see that the probability that a divisor class chosen in step 2 is equal to any prescribed divisor class is $(1+o(1))/h(F/k)$. Hence the probability that E will be S-smooth is $(1+o(1))p_{S,n}$, and after expected $(1+o(1))/p_{S,n}$ tries an S-smooth E is found with probability $\geq 1-(1-(1+o(1))p_{S,n})^{(1+o(1))/p_{S,n}} \geq 1-1/e \geq 1/2$.

For any vector $(\mu_1, \ldots, \mu_r, \rho_1, \ldots, \rho_s) \in \Lambda_S$ and $B = \sum_{i=1}^s \rho_i P_i$ it holds by definition of Λ_S that $[B] = \deg(B)[D_0] + \sum_{i=1}^r \mu_i [D_i]$. From $u + \lambda \in \Lambda_S$ it thus follows that $[E] = n[D_0] + \sum_{i=1}^r (v_i + \lambda_i)[D_i]$. Now $[E] = [D] + n[D_0] + \sum_{i=1}^r v_i [D_i]$ by definition of E and hence $[D] = \sum_{i=1}^r \lambda_i [D_i]$, as required. □

From section 2 we conclude that the steps 2, 3, 6 and 7 of Algorithm 2 and steps 2 and 4 of Algorithm 4 have an expected running time polynomial in g and $\log(q)$. In order to satisfy the assumptions of Lemma 3 and to estimate the overall running time we now need to answer the following main questions. What is a lower bound for $p_{S,n}$ depending on S and n? How can we ensure that the given $[D_i]$ form a generating system of $\mathcal{C}l^0(F/k)$? It would be advantageous if we could check for $U = \Lambda$ in Algorithm 2. This is equivalent to checking $(\mathbb{Z}^r : U) = (\mathbb{Z}^r : \Lambda) = h(F/k)$. Note that for the latter it is actually sufficient to know an approximation h such that $2^{-1/2}h(F/k) < h < 2^{1/2}h(F/k)$. For then, $2^{-1/2}h < h(F/k) < 2^{1/2}h$ and $(\mathbb{Z}^r : U) = h(F/k)$ is equivalent to $2^{-1/2}h < (\mathbb{Z}^r : U) < 2^{1/2}h$, since $(\mathbb{Z}^r : U)$ is correct up to integral multiples. So how can we efficiently compute such an approximation h of $h(F/k)$? These and other issues are discussed in the next sections.

4. Smoothness

Informally a smooth divisor is an effective divisor of possibly large degree which consists only of places of small degree. Index calculus methods require that the number of smooth divisors is large enough in relation to a reference value such as the number of all effective divisors. In this section we prove some lower bounds for this ratio. We only use simple combinatorical techniques and do not consider analytical techniques like the saddle point method.

Let E be a randomly chosen effective divisor from a randomly chosen divisor class of degree n. We want to compute, or estimate from below, the probability $p_{S,n}$ that E is S-smooth where $S = \mathcal{P}l^{\leq m}(F/k)$ is the set of places of F/k of degree less than or equal to m for some $m \in \mathbb{Z}$.

We say that an effective divisor is (n,m)-smooth for $n, m \in \mathbb{Z}$ if its degree is n and all places in its support are of degree less than or equal to m. We denote the number of (n,m)-smooth divisors by $\psi(n,m)$ and the number of effective divisors in the divisor class $[D]$ by $\#_0[D]$.

The probability $\Pr(E = D)$ that E equals any given effective divisor D of degree n satisfies

$$\begin{aligned}\Pr(E = D) &= \Pr\big(E = D \,|\, [E] = [D]\big)\Pr([E] = [D]) \\ &= \frac{1}{\#_0[D]h(F/k)}.\end{aligned}$$

Thus we obtain

$$\begin{aligned}p_{S,n} &= \Pr(E \ (n,m)\text{-smooth}) = \sum_{D \ (n,\,m)\text{-smooth}} \Pr(E = D) \\ &= \frac{1}{h(F/k)} \sum_{D \ (n,\,m)\text{-smooth}} \frac{1}{\#_0[D]}. \qquad (6)\end{aligned}$$

If $n \geq 2g-1$ we have $\#_0[D_1] = \#_0[D_2]$ for any two effective divisors D_1 and D_2 of degree n according to [23, p. 160]. Thus $\#_0[D]h(F/k) = \psi(n,n)$ and continuing (6) we get in this case

$$\begin{aligned}p_{S,n} &= \frac{1}{h(F/k)} \sum_{D \ (n,\,m)\text{-smooth}} \frac{1}{\#_0[D]} = \frac{1}{\psi(n,n)} \sum_{D \ (n,\,m)\text{-smooth}} 1 \\ &= \psi(n,m)/\psi(n,n).\end{aligned}$$

We have proven

Lemma 7. *Let $n \geq 2g-1$ and $S = \mathcal{P}l^{\leq m}(F/k)$. The probability $p_{S,n}$, that a randomly chosen effective divisor from a randomly chosen divisor class of degree n is S-smooth, satisfies*

$$p_{S,n} = \psi(n,m)/\psi(n,n).$$

While Lemma 7 is already sufficient for our complexity analysis it is in practice more efficient to work with smaller values of n. The smallest sensible value is $n = g$ since divisor classes of smaller degree may not contain effective divisors and $\psi(n,n)$ decreases very fast with n. For $n = g$ it is in general quite problematic to estimate the sum in (6) since there is no explicit formula for $\#_0[D]$. On average we would expect $\#_0[D] = 1$ (or very small) by the theorem of Riemann-Roch and hence about $p_{S,n} = \psi(n,m)/h(F/k)$. But since the (n,m)-smooth divisors D form a very small subset of all effective divisors of degree n when for example $m \approx n^{1/2}$, the case we are interested in, it could well be that $\#_0[D]$ is significantly larger than 1, although this appears unlikely.

One solution is to find upper bounds for $\#_0[D]$. Let $\mathcal{L}(D) = \{\, z \in F^\times \,|\, (z) + D \geq 0 \,\} \cup \{0\}$ denote the Riemann-Roch space of the divisor D, which is a finite dimensional k-vector space, and $\dim(D)$ its dimension. The theorem of Gieseker-Petri [10] states that the multiplication map $\mathcal{L}(D) \otimes_k \mathcal{L}(W - D) \to \mathcal{L}(W)$ for W a canonical divisor is injective if F/k is defined by a curve with general moduli. Since $\dim(W - D) = \dim(D) - \deg(D) - 1 + g = \dim(D) - 1$ and $\dim(W) = g$ we see $\dim(D)(\dim(D) - 1) \leq g$. Thus we obtain roughly $\#_0[D] \leq q^{\sqrt{g}}$ and this yields $p_{S,n} \geq \psi(n,m)/(h(F/k)q^{\sqrt{g}})$, which would in fact be sufficient for the later complexity analysis. However, the theorem of Gieseker-Petri requires general

moduli and thus we cannot a priori rule out that there are special families of global function fields for which it is not sufficiently true.

Another solution is to construct special but still sufficiently many (n,m)-smooth divisors D such that $\#_0[D] = 1$ (or very small) by definition. This is implicitly done in the case of hyperelliptic function fields in [8], using the notion of reduced ideals. Such ideals give rise to (reduced) divisors D which indeed satisfy $\#_0[D] = \dim(D) = 1$. It is unclear whether this technique also works for non-hyperelliptic function fields.

We briefly indicate why $\dim(D) = 1$ for the above divisors in the hyperelliptic case. Let $k[x,y]$ be the affine coordinate ring of a hyperelliptic curve of genus g and assume for simplicity that there is only one rational place at infinity, denoted by ∞. As usual, for the valuation at infinity we have $v_\infty(x) = -2$ and $v_\infty(y) = -(2g+1)$. An ideal I of $k[x,y]$ is reduced if it is generated by $a, y-b \in k[x,y]$ for $a, b \in k[x]$ with $\deg(b) < \deg(a) \leq g$ and if there is an ideal $\bar{I}$ of $k[x,y]$ generated by $a, y-\bar{b} \in k[x,y]$ with $\deg(\bar{b}) < \deg(a)$ such that $I\bar{I} = ak[x,y]$, the ideal generated by a. Since $k[x,y]$ is a Dedekind domain we can factorise $I = \prod_i \mathfrak{p}^{n_i}$ and associate places P_i to the prime ideals $\mathfrak{p}_i$. Then by definition $D = \sum_i n_i P_i$ is the effective divisor of degree $\leq g$ corresponding to I. The space $\mathcal{L}(D)$ consists of all $z \in I^{-1}$ with $v_\infty(z) \geq 0$. We have $I^{-1} = \bar{I}/a$ from above, hence I^{-1} is generated by 1 and $(y-\bar{b})/a$, and z is of the form $z = \lambda + \mu(y-\bar{b})/a$ for some $\lambda, \mu \in k[x]$. Now if $\lambda \neq 0$ then $v_\infty(\lambda)$ is even and ≤ 0 since $v_\infty(x) = -2$, and if $\mu \neq 0$ then $v_\infty(\mu(y-\bar{b})/a) = v_\infty(\mu/a)v_\infty(y-\bar{b})$ is odd and ≤ -1. Namely, $v_\infty(y) = -(2g+1)$ and $v_\infty(\bar{b}) \geq -2g$ because of $\deg(\bar{b}) \leq g$, hence $v_\infty(y-\bar{b}) = -(2g+1)$, and $v_\infty(\mu/a)$ is even and $\leq 2g$, so $v_\infty(\mu/a)v_\infty(y-\bar{b}) \leq 2g-(2g+1) \leq -1$. Then $v_\infty(z) = \min\{v_\infty(\lambda), v_\infty(\mu(y-\bar{b})/a)\}$. For $v_\infty(z) \geq 0$ we hence necessarily have $\lambda \in k$ and $\mu = 0$, that is $\dim(D) = 1$.

In view of Lemma 7 we continue to investigate the ratio $\psi(n,m)/\psi(n,n)$. The number of places of F/k of degree d is denoted by $\pi(d)$.

Theorem 8. *The number of (n,m)-smooth divisors satisfies*

$$\psi(n,m) \;\geq\; \psi(n,n)/\exp\Big(\big(2+O(1/m)\big)\Big(\frac{n}{m}\Big)\big(\log(n)+\log(g+1)+O(1)\big)\Big)$$

for $m \geq 4\log(14g+4)/\log(q) + 2$.

Proof. We adopt the notation $\psi_{=,\leq}(n,m) = \psi(n,m)$ and let $\psi_{=,<}(n,m)$, $\psi_{=,\geq}(n,m)$ and $\psi_{=,\geq,\leq}(n,m_0,m)$ denote the numbers of effective divisors of degree equal to n whose support consist of places of degree d such that $d < m$, $d \geq m$ and $m_0 \leq d \leq m$ respectively.

Let $m_0 = \lfloor m/2 \rfloor$. We have

$$\psi_{=,\leq}(n,n) \;=\; \sum_{i=0}^{n} \psi_{=,<}(n-i,m_0)\,\psi_{=,\geq}(i,m_0) \tag{9}$$

by writing any divisor as a sum of a divisor containing only places of degree $< m_0$ and a divisor containing only places of degree $\geq m_0$. Similarly,

$$\psi_{=,\leq}(n,m) \;=\; \sum_{i=0}^{n} \psi_{=,<}(n-i,m_0)\,\psi_{=,\geq,\leq}(i,m_0,m). \tag{10}$$

Clearly, $\psi_{=,\geq}(i,m_0) = \psi_{=,\geq,\leq}(i,m_0,m) = 0$ for $i < m_0$, such that i in (9) and (10) effectively runs from m_0 to n. For $i \geq m_0$ we observe the following two estimations,

$$\psi_{=,\geq}(i,m_0) \;\leq\; q^i\,\exp\big((2+O(1/m))(i/m)(\log(g+1)+O(1))\big) \tag{11}$$

and

$$\psi_{=,\geq,\leq}(i,m_0,m) \geq q^i/\exp\big((2+O(1/m))(i/m)(\log(i/m)+\log(m)+O(1))\big). \tag{12}$$

Before proving (11) and (12) we show how the estimation of $\psi(n,m)$ in Theorem 8 follows. Combining (9) and (10) we obtain

$$\begin{aligned}
\psi_{=,\leq}(n,n) &= \sum_{i=m_0}^{n} \psi_{=,<}(n-i,m_0)\,\psi_{=,\geq,\leq}(i,m_0,m)\left(\frac{\psi_{=,\geq}(i,m_0)}{\psi_{=,\geq,\leq}(i,m_0,m)}\right)\\
&\leq \sum_{i=m_0}^{n} \psi_{=,<}(n-i,m_0)\,\psi_{=,\geq,\leq}(i,m_0,m)\cdot\\
&\quad \exp\big((2+O(1/m))(i/m)(\log(i/m)+\log(m)+\log(g+1)+O(1))\big)\\
&\leq \psi_{=,\leq}(n,m)\cdot\\
&\quad \exp\big((2+O(1/m))(n/m)(\log(n/m)+\log(m)+\log(g+1)+O(1))\big),
\end{aligned}$$

as required.

For the proof of (11) we note that F/k has a rational subfield $k(x)$ of index $[F:k(x)]=O(g)$. Above every place of $k(x)$ there are hence at most $O(g)$ places of F/k. Let us say that an effective divisor is (i,m_0)-rough if its degree is i and its support consists only of places of degree $\geq m_0$. If D is an (i,m_0)-rough divisor then it consists of at most $\lfloor i/m_0\rfloor$ places of F/k and its norm $\mathrm{N}_{F/k(x)}(D)$ has degree i. Together this means that there are at most $O(g)^{\lfloor i/m_0\rfloor}$ many (i,m_0)-rough divisors E of F/k such that $\mathrm{N}_{F/k(x)}(E)=\mathrm{N}_{F/k(x)}(D)$. In other words, the norm $\mathrm{N}_{F/k(x)}$ is an at most $O(g)^{\lfloor i/m_0\rfloor}$-to-1 map of (i,m_0)-rough divisors to divisors of degree i. Since the number of divisors of degree i of $k(x)$ is $(q^{i+1}-1)/(q-1)$, we obtain observing $(q^{i+1}-1)/(q-1)\leq 2q^i$ and $\lfloor i/m_0\rfloor=(2+O(1/m))(i/m)$ that

$$\begin{aligned}
\psi_{=,\geq}(i,m_0) &\leq O(g)^{\lfloor i/m_0\rfloor}(q^{i+1}-1)/(q-1)\\
&\leq q^i\exp\big((2+O(1/m))(i/m)(\log(g+1)+O(1))\big),
\end{aligned}$$

which proves (11). The two observations are seen as follows. Firstly,

$$\frac{q^{i+1}-1}{q^i(q-1)}\leq\frac{q^{i+1}}{q^i(q-1)}\leq\frac{q}{q-1}\leq 2.$$

Secondly, for any $a\in\mathbb{R}^{\geq 2}$ we have

$$\frac{1}{\lfloor a\rfloor}\leq\frac{1}{a-1}=\frac{1}{1-1/a}\cdot\frac{1}{a}\leq(1+2/a)\cdot\frac{1}{a}.$$

Thus with $a=m/2$, $m\geq 4$ and $m_0=\lfloor a\rfloor$ we see that

$$\lfloor i/m_0\rfloor\leq i/\lfloor a\rfloor\leq 2(1+4/m)/(i/m).$$

For the proof of (12) write $i=\lfloor i/m_0\rfloor m_0+r$ with $0\leq r<m_0$. Every divisor of degree i with support between m_0 and m can be written as a sum of $\lfloor i/m_0\rfloor$ places of degree m_0 if $r=0$, or in the form D_1+D_2 where D_1 consists of $\lfloor i/m_0\rfloor-1$ places of degree m_0 and D_2 consists of one place of degree m_0+r if $r>0$. The number of places of degree m_0 satisfies $|\pi(m_0)-q^{m_0}/m_0|\leq(7g+2)q^{m_0/2}/m_0$, see [23, p. 178–180]. Since $m_0\geq m/2-1\geq 2\log_q(14g+4)$ from the assumptions we have $1-(7g+2)q^{-m_0/2}\geq 1/2$ and $\pi(m_0)\geq q^{m_0}/m_0(1-(7g+2)q^{-m_0/2})\geq q^{m_0}/(2m_0)$. Using this in the following estimations we obtain for $r=0$

$$\begin{aligned}
\psi_{=,\geq,\leq}(i,m_0,m) &\geq \pi(m_0)^{\lfloor i/m_0\rfloor}/\lfloor i/m_0\rfloor!\\
&\geq (q^{m_0}/m_0(1-(7g+2)q^{-m_0/2}))^{\lfloor i/m_0\rfloor}/\lfloor i/m_0\rfloor!\\
&\geq q^i/\exp\big(\lfloor i/m_0\rfloor(\log(\lfloor i/m_0\rfloor)+\log(m_0)+O(1))\big)\\
&\geq q^i/\exp\big((2+O(1/m))(i/m)(\log(i/m)+\log(m)+O(1))\big)
\end{aligned}$$

and for $r > 0$

$$\begin{aligned}\psi_{=,\geq,\leq}(i,m_0,m) &\geq \pi(m_0)^{\lfloor i/m_0\rfloor-1}/(\lfloor i/m_0\rfloor-1)!\cdot\pi(m_0+r)\\ &\geq (q^{m_0}/m_0(1-(7g+2)q^{-m_0/2}))^{\lfloor i/m_0\rfloor-1}/(\lfloor i/m_0\rfloor-1)!\cdot\\ &\quad q^{m_0+r}/(m_0+r)(1-(7g+2)q^{-(m_0+r)/2})\\ &\geq q^i/\exp\big(\lfloor i/m_0\rfloor(\log(\lfloor i/m_0\rfloor)+\log(m_0)+O(1))\big)\\ &\geq q^i/\exp\big((2+O(1/m))(i/m)(\log(i/m)+\log(m)+O(1))\big).\end{aligned}$$

This proves (12). □

Lemma 7 and Theorem 8 yield a lower bound for $p_{S,n}$ which is sufficient for our general complexity analysis, but this bound is not as good as standard smoothness bounds obtained for example in the case of polynomial rings. The following theorem allows us to obtain this standard smoothness probability in our situation under the additional assumption that h is not much bigger than q^g, in a sense made precise in Theorem 29.

Theorem 13. *Let $0 < \alpha < 1$ and $\gamma = 3/(1-\alpha)$. There is $\delta \in \mathbb{R}^{>0}$ with $\log(\delta) \geq 3/2\log(2)(1-\alpha)/\alpha$ such that the number of (n,m)-smooth divisors satisfies*

$$\psi(n,m) \geq q^n/\exp\big(u\,(\log(u)+\log(\log(u))+\gamma)\big),$$

where $u = n/m$ and $n, m \in \mathbb{Z}^{\geq\delta}$ with $3\log_q(14g+4) \leq m \leq n^\alpha$ and $u \geq 2\log(g+1)$.

Proof. The proof is based on the method of [18] for the case $\mathbb{Z}$, compare [21]. First we observe

$$u \geq m^{(1-\alpha)/\alpha} \tag{14}$$

because of the assumptions. Thus $u > 1$ holds for $m > 1$ so that we can define $b := (1 - 1/\log(u))m$. The following estimations require a certain minimal size of the parameters n, m, u, b which we guarantee by a suitable minimal choice of δ only depending on α. First let $\delta \geq 1$, we will enlarge it several times. By (14) and $m \geq \delta$ as assumed the inequality $u \geq \delta$ holds. We enlarge δ such that $b \geq (1 - 1/\log(\delta))\delta \geq 1$ is fulfilled, which is the case for $\delta \geq 4$.

Until the end of the proof we use D for divisors being the sum of $\lfloor u\rfloor$ not necessarily distinct places of a degree $> b$ and $\leq m$ and we let $d := \deg(D)$. Thus $b\lfloor u\rfloor < d \leq m\lfloor u\rfloor \leq n$ holds and we obtain the lower estimate

$$\psi(n,m) \geq \sum_D \psi(n-d,\lfloor b\rfloor), \tag{15}$$

because the supports of the D and of the divisors counted by $\psi(n-d,\lfloor b\rfloor)$ are disjoint. We now estimate the $\psi(n-d,\lfloor b\rfloor)$ and then the sum from below. The following inequalities will be used,

$$\begin{aligned}(n-d)/b &\leq \big\lfloor (n-d)/\lfloor b\rfloor\big\rfloor + 1\\ &\leq 2(n-d)/b + 1 \qquad \leq \quad 2(n/b - \lfloor u\rfloor) + 1\\ &\leq 2\big((1-1/\log(u))^{-1}u - u + 1\big) + 1\\ &\leq 2u/\log(u) + 3\\ &\leq 3u/\log(u),\end{aligned} \tag{16}$$

valid for all divisors D. For the second estimation we use $b\lfloor u\rfloor < d$ as above and for the last estimation we enlarge δ such that $2\delta/\log(\delta) + 3 \leq 3\delta/\log(\delta)$, which is the case for $\delta \geq 5$. Also, if $\log(\delta) \geq 6$ then $1 - 1/\log(u) \geq 5/6$ and $5/6m - 1 \geq 2/3m$, so

$$\lfloor b\rfloor \geq (1-1/\log(u))m - 1 \geq 5/6m - 1 \geq 2/3m.$$

Thus we get from the assumptions

$$\begin{aligned} \lfloor b \rfloor &\geq 2/3m \geq 2/3 \cdot 3 \log_q(14g+4) \\ &\geq 2 \log_q(14g+4), \end{aligned}$$

hence

$$(7g+2)q^{-\lfloor b \rfloor/2} \leq 1/2. \tag{17}$$

The number of places of degree r satisfies $|\pi(r) - q^r/r| \leq (7g+2)q^{r/2}/r$, see [23, p. 178–180]. Write $n-d = l\lfloor b \rfloor + r$ with $0 \leq r < \lfloor b \rfloor$ and $l = \lfloor (n-d)/\lfloor b \rfloor \rfloor$. By (17) we have $1 - (7g+2)q^{-\lfloor b \rfloor/2} \geq 1/2$, hence $\pi(\lfloor b \rfloor) \geq q^{\lfloor b \rfloor}/\lfloor b \rfloor (1 - (7g+2)q^{-\lfloor b \rfloor/2}) \geq q^{\lfloor b \rfloor}/(2\lfloor b \rfloor)$. If $r \geq 2\log_q(14g+4)$ then similarly $\pi(r) \geq q^r/r \cdot (1-(7g+2)q^{-r/2}) \geq q^r/(2r)$ and

$$\begin{aligned} \psi(n-d, \lfloor b \rfloor) &\geq \pi(\lfloor b \rfloor)^l / l! \cdot \pi(r) \\ &\geq q^{n-d} / \exp\big(l(\log(l) + \log(2\lfloor b \rfloor)) + \log(2r)\big) \\ &\geq q^{n-d} / \exp\big((l+1)(\log(l) + \log(2\lfloor b \rfloor))\big). \end{aligned} \tag{18}$$

If on the other hand $r < 2\log_q(14g+4)$ then $q^{-r} \geq (14g+4)^{-2}$ and

$$\begin{aligned} \psi(n-d, \lfloor b \rfloor) &\geq \pi(\lfloor b \rfloor)^l / l! \cdot q^r \cdot q^{-r} \\ &\geq q^{n-d} / \exp\big(l(\log(l) + \log(2\lfloor b \rfloor)) + 2\log(14g+4)\big) \\ &\geq q^{n-d} / \exp\big(l(\log(l) + \log(2\lfloor b \rfloor)) + 2\log(g+1) + 6\big) \end{aligned} \tag{19}$$

using $2\log(14g+4) \leq 2\log(g+1) + 2\log(14)$ and $2\log(14) \leq 6$. Combining (18) and (19) we obtain

$$\begin{aligned} \psi(n-d, \lfloor b \rfloor) &\geq q^{n-d} / \exp\big((l+1)(\log(l) + \log(2\lfloor b \rfloor)) + 2\log(g+1) + 6\big) \\ &\geq q^{n-d} / \exp\left((l+1)(\log(n-d) + \log(2)) + 2\log(g+1) + 6\right) \end{aligned} \tag{20}$$

for all $0 \leq r \leq \lfloor b \rfloor$.

We have

$$l + 1 \leq 3u/\log(u) \tag{21}$$

by (16),

$$\log(n-d) = \log((n-d)/b) + \log(b) \leq \log(u) + \log(m) \tag{22}$$

by (16) if $\log(\delta) \geq 3$ and because of $b \leq m$ (this in turn because of the definition of b),

$$\log(m) \leq \alpha/(1-\alpha)\log(u) \tag{23}$$

because of (14) and

$$2\log(g+1) \leq u \tag{24}$$

due to the assumptions. Using (21), (22), (23) and (24) in (20) yields

$$\begin{aligned} \psi(n-d, \lfloor b \rfloor) &\geq q^{n-d} / \exp\big((l+1)(\log(n-d) + \log(2)) + 2\log(g+1) + 6\big) \\ &\geq q^{n-d} / \exp\big((3u/\log(u))(\log(u) + \log(m) + \log(2)) + u + 6\big) \\ &\geq q^{n-d} / \exp\big(u((3-2\alpha)/(1-\alpha) + 3\log(2)/\log(u)) + 6\big) \\ &\geq q^{n-d} / \exp(cu), \end{aligned} \tag{25}$$

with $c = 3/(1-\alpha)$ and sufficiently large δ such that $2\alpha/(1-\alpha)u \geq 3\log(2)u/\log(u) + 6$. This inequality holds for all divisors D (also for $n-d=0$).

Now we turn to the estimation of the sum in (15). Because of (25), (15) becomes

$$\psi(n,m) \geq q^n \cdot \exp(-cu) \cdot \sum_D q^{-d}. \tag{26}$$

We estimate the sum over all D via

$$\sum_D q^{-d} \geq \Big(\sum_{b<\deg(P)\leq m} q^{-\deg(P)} \Big)^{\lfloor u \rfloor} / \lfloor u \rfloor! \tag{27}$$

where the right hand sum runs over all places P with $b < \deg(P) \leq m$ (the $\lfloor u \rfloor$-power forms the sum of all values q^{-d} respecting the order wherefore we divide by $\lfloor u \rfloor!$).

Using (17) we see $q^{-j}\pi(j) \geq (1/m)(1-(7g+2)q^{-b/2}) \geq 1/(2m)$ for $b < j \leq m$. Observing $m - b = m/\log(u)$ by definition this then yields:

$$\begin{aligned} \sum_{b<\deg(P)\leq m} q^{-\deg(P)} &= \sum_{b<j\leq m} q^{-j}\pi(j) \\ &\geq (m-b)/(2m) \\ &\geq (2\log(u))^{-1}. \end{aligned} \tag{28}$$

Because of $\lfloor u \rfloor! \leq \exp\big(\lfloor u \rfloor(\log(\lfloor u \rfloor) - \log(2))\big)$ for δ large enough using Stirlings's formula in the form $n! = \exp(n(\log(n) - 1 + o(1)))$ it follows from (28), (27) and (26) that

$$\begin{aligned} \psi(n,m) &\geq q^n \cdot \exp(-cu) \cdot (2\log(u))^{-\lfloor u \rfloor} / \lfloor u \rfloor! \\ &\geq q^n / \exp\big(u\,(\log(u) + \log(\log(u)) + \gamma)\big) \end{aligned}$$

for $\gamma = -\log(2) + \log(2) + c = 3/(1-\alpha)$ and the sufficiently large δ, depending only on α. □

We remark that Theorem 8 and Theorem 13 are clearly also true for $\psi_{\leq,\leq}(n,m) = \sum_{j=0}^{n} \psi(j,m) \geq \psi(n,m)$ instead of $\psi(n,m)$. Combining the previous smoothness results we obtain

Theorem 29. *Let $0 < \alpha < 1$. For $g \to \infty$ assume $n \geq 2g-1$ and $4\log(14g+4)/\log(q) + 2 \leq m \leq n^\alpha$. The smoothness probability is then bounded from below by*

$$p_{S,n} \geq 1/\exp\left(\left(\frac{4 + O(1/m)}{1-\alpha} \right) \left(\frac{n}{m} \right) \log\left(\frac{n}{m} \right) \right).$$

If $h(F/k)/q^g \leq \exp(\sigma(n/m)\log(n/m))$ for some (not necessarily constant) $\sigma > 0$ and $m \geq \delta$ for the constant δ from Theorem 13 then

$$p_{S,n} \geq 1/\exp\left(\big(\sigma + 1 + o(1)\big) \left(\frac{n}{m} \right) \log\left(\frac{n}{m} \right) \right).$$

Proof. The first inequality follows immediately from Lemma 7 and Theorem 8, using $\log(g+1) \leq \log(n)$ for $g \geq 2$, $\log(n) \leq 1/(1-\alpha)\log(n/m)$ and that n/m tends to infinity because $m \leq n^\alpha$ by the assumptions.

For the second inequality we observe using [23, p. 160]

$$\begin{aligned} \psi(n,n) &= h(F/k)(q^{n+1-g}-1)/(q-1) \\ &\leq h(F/k)\,2q^{n+1-g}/q \\ &\leq q^n \exp(\log(2) + \sigma(n/m)\log(n/m)) \\ &\leq q^n \exp((\sigma + o(1))(n/m)\log(n/m)), \end{aligned} \tag{30}$$

because n/m tends to infinity. The inequality then follows from Lemma 7, Theorem 13 and (30). Theorem 13 can be applied since $n \geq m \geq \delta$, $3\log_q(14g+4) \leq 4\log(14g+4)/\log(q) + 2 \leq m$ and $n/m \geq 2\log(g+1)$ for sufficiently large g since $n \geq 2g-1$. □

From a function field version of the theorem of Brauer-Siegel with explicit error term it follows for example that $|\log(h(F/k)/q^g)| = O([F : k(x)]\log(g\log(q)))$ for a transcendent element $x \in F$, see Theorem 67 or [14, Theorem 3].

Theorem 29 is useful for the theoretical complexity analysis of Algorithm 2 as g tends to infinity. For practical applications it is however much better to compute the ratio $\psi(n,m)/\psi(n,n)$ more precisely instead of using the asymptotic formulae. This can be achieved by the next theorem.

Theorem 31. *Let* $N(n,m) = \sum_{0<d\le m, d|n} d\pi(d)$. *Then*

$$\psi(n,m) = \frac{1}{n}\sum_{j=0}^{n-1} N(n-j,m)\psi(j,m).$$

Proof. We compute with formal series. For the logarithm of the generating function of $\psi(n,m)$ we obtain

$$\begin{aligned}\log\left(\sum_{j=0}^{\infty}\psi(j,m)t^j\right) &= \log\left(\prod_{d=1}^{m}(1-t^d)^{-\pi(d)}\right)\\ &= \sum_{d=1}^{m}\pi(d)\sum_{j=1}^{\infty}t^{dj}/j \;=\; \sum_{j=1}^{\infty}\sum_{d=1}^{m}d\pi(d)t^{dj}/(dj)\\ &= \sum_{j=1}^{\infty}N(j,m)t^j/j.\end{aligned}$$

By the Newton relations (differentiate both sides, multiply by $\sum_{j=0}^{\infty}\psi(j,m)t^j$ and equate the coefficients of t^{n-1}) the assertion follows. □

If $\psi_{\le,\le}(n,m) = \sum_{j=0}^{n}\psi(j,m)$ then we obtain $\psi_{\le,\le}(n,m) = (1/n)\sum_{j=0}^{n-1}(1 + N(n-j,m))\psi_{\le,\le}(j,m)$ as in the above proof where we start with the generating function $\sum_{j=0}^{\infty}\psi_{\le,\le}(j,m)t^j = (1-t)^{-1}\sum_{j=0}^{\infty}\psi(j,m)t^j$.

A recursive algorithm for computing $\psi(n,m)$ is immediate from the formula, observing $\psi(0,m) = 1$. Assuming the $N(j,m)$ are known we successively compute and store $\psi(j,m)$ for $0 \le j \le n$ in $O(n^2)$ operations. The determination of the $N(j,m)$ is more problematic if m is large with respect to g, since knowledge of the values $\pi(d)$ for $1 \le d \le m$ is required. This problem can be overcome by recent point counting techniques such as [15, 24]. If it is sufficient to compute an approximate value of $N(n,m)$ we can compute $\pi(d)$ for some small values of d and then use the approximation $\pi(d) \approx q^d/d$ since $|\pi(d) - q^d/d| < (7g+2)q^{d/2}/d$ by [23, p. 178–180]. A convenient circumstance is that the deviations of $\pi(d)$ from q^d/d carry less weight in the total value of $\psi(n,m)$ as d increases, which is directly seen from the recursion formula.

We can apply this recursion to $\psi(n,m)$ and $\psi(n,n)$. If the class number is known and $n \ge 2g-1$ then $\psi(n,n)$ can also be computed via $\psi(n,n) = h(F/k)(q^{n+1-g} - 1)/(q-1)$, see [23, p. 160].

5. Generation of the divisor class group

In this section we describe how to compute a divisor D_0 of degree one and a small number of divisors $D_1, \ldots, D_r$ of degree zero such that we know without further information that $\mathcal{C}l(F/k)$ is generated by the classes $[D_i]$. We denote by $N_r(F/k)$ the number of places of degree one in the constant field extension of F/k of degree r.

Proposition 32. *Assume $g > 0$ and let $d = \lceil 2\log_q(2g) \rceil$. There is a divisor D_0 of degree one whose support consists of places of degree less than or equal to $d+1$.*

Proof. The theorem of Hasse-Weil states that $|N_r(F/k) - (q^r+1)| \leq 2gq^{r/2}$. For $r \geq d$ we have $q^r + 1 > q^r \geq 2gq^{r/2}$ and hence $N_r(F/k) > 0$. This implies that there is a place P of F/k of degree $\deg(P)|r$.

We apply the above reasoning with $r = d$ and $r = d+1$ and obtain the existence of two places P_1 and P_2 of F/k such that $\deg(P_1)|d$ and $\deg(P_2)|d+1$. The degrees $\deg(P_1)$ and $\deg(P_2)$ are coprime because d and $d+1$ are coprime. Let $l_1, l_2 \in \mathbb{Z}$ such that $l_1 \deg(P_1) + l_2 \deg(P_2) = 1$ by the extended Euclidean algorithm. Then $D_0 = l_1 P_1 + l_2 P_2$ is a divisor of degree one. □

In the case $g = 0$ there clearly exist places of degree one yielding divisors of degree one. From Proposition 32 the following algorithm is immediate.

Algorithm 33. *(Divisor of degree one)*
Input: *The function field F/k of genus $g > 0$.*
Output: *A divisor D_0 of degree zero.*

1. *Set $d = \lceil 2\log_q(2g) \rceil$.*
2. *Compute successively all places of degree $< d$. If there are places P_i of coprime degrees write $\sum_i l_i \deg(P_i) = 1$ by the extended Euclidean algorithm, and return $D_0 = \sum_i l_i P_i$.*
3. *Compute a place P_1 of degree dividing d. If $d = 1$ return $D_0 = P_1$. Otherwise compute a place P_2 of degree dividing $d+1$. Write $l_1 \deg(P_1) + l_2 \deg(P_2) = 1$ by the extended Euclidean algorithm, and return $D_0 = l_1 P_1 + l_2 P_2$.*

Algorithm 33 essentially only involves factoring techniques over finite fields. By our assumption on the representation of F/k, it has an expected running time polynomial in g and $\log(q)$.

The computation of a small generating system $[D_1], \ldots, [D_r]$ of $\mathcal{C}l^0(F/k)$ is achieved by the following theorem.

Theorem 34. *Let D_0 be a divisor of degree one of F/k and $d \in \mathbb{Z}^{\geq 1}$ such that $N_d(F/k) > (g-1)2q^{d/2}$. Denote by $P_1, \ldots P_r$ all places of degree dividing d and set $D_i = P_i - \deg(P_i)D_0$. Then $\deg(D_i) = 0$ and the classes $[D_i]$ generate $\mathcal{C}l^0(F/k)$.*

Proof. Let $\chi : \mathcal{C}l(F/k) \to \mathbb{C}^\times$ be a character of finite order of $\mathcal{C}l(F/k)$ and define the character sum $N_d(\chi) = \sum_{\deg(P)|d} \deg(P) \cdot \chi([P])^{d/\deg(P)}$. If $\chi = 1$ is the principal character then $N_d(\chi) = N_d(F/k)$. Let $\rho_\chi = 1$ if χ is trivial on $\mathcal{C}l^0(F/k)$, and $\rho_\chi = 0$ otherwise. By the theorem of Hasse-Weil for character sums we know that $|N_d(\chi)| \leq (g-1)2q^{d/2}$ for $\rho_\chi = 0$.

Let χ be a character with $\rho_\chi = 0$. Since character values are roots of unity the inequality $N_d(F/k) \geq |N_d(\chi)|$ holds for every $d \geq 1$. Hence if d satisfies $N_d(F/k) > (g-1)2q^{d/2}$ then $N_d(F/k) > |N_d(\chi)|$, and there exists a place P of degree dividing d such that $\chi([P]) \neq 1$.

Any character of χ of $\mathcal{C}l^0(F/k)$ can be extended to a character of $\mathcal{C}l(F/k)$ by defining $\chi([D_0]) = 1$. If χ is non trivial on $\mathcal{C}l^0(F/k)$ then $\rho_\chi = 0$ and there is a place of degree dividing d such that $\chi([P]) \neq 1$. Using $D = P - \deg(P)D_0$ we have found a divisor of degree zero such that $\chi([D]) \neq 1$.

We have shown that if $\chi([D]) = 1$ for all the above divisors $D = P - \deg(P)D_0$ then $\chi = 1$ on $\mathcal{C}l^0(F/k)$. Using duality we see that such divisors D generate all of $\mathcal{C}l^0(F/k)$. □

The algorithm to compute a generating system $[D_i]$ is now straightforward.

Algorithm 35. *(Generators of $\mathcal{C}l^0(F/k)$)*

Input: *The function field F/k.*
Output: *Compute divisors $D_1, \ldots, D_r$ whose classes generate $\mathcal{C}l^0(F/k)$.*

1. *Compute a divisor D_0 of degree one, using Algorithm 33.*
2. *Using increasing values of d find the smallest $d \geq 1$ such that $N_d(F/k) = \sum_{\deg(P)|d} \deg(P) > (g-1)2q^{d/2}$.*
3. *Compute all places $P_1, \ldots P_r$ of degree dividing d.*
4. *Return the $D_i := P_i - \deg(P_i)D_0$ for $1 \leq i \leq r$.*

The generating system computed by Algorithm 35 consists of a number of places which is polynomial in g and q. By our assumption on the representation of F/k, Algorithm 35 has an expected running time polynomial in g and q.

Clearly, the whole divisor class group $\mathcal{C}l(F/k)$ is generated by D_0 and $D_1, \ldots, D_r$. We remark that a similar, but in some situation much weaker bound for the generation of the class group has been obtained in [17, 22].

6. Approximation of the class number

The approximation of the class number $h(F/k)$ can be achieved using the following theorem.

Theorem 36. *Let $d \in \mathbb{Z}^{\geq 0}$. The class number $h(F/k)$ satisfies*

$$\left| \log\big(h(F/k) \,/\, q^g\,\big) - \sum_{r=1}^{d} \frac{q^{-r}}{r}\big(N_r(F/k) - q^r - 1\big) \right| \;\leq\; \frac{2g}{q^{1/2}-1} \cdot \frac{q^{-d/2}}{d+1}.$$

Proof. Let x be a separating element of F/k. The zeta-functions of F/k and $k(x)/k$ are equal to $\exp\big(\sum_{r=1}^{d} q^{-r}/rN_r(F/k)\big)$ and $\exp\big(\sum_{r=1}^{d} q^{-r}/r(q^r+1)\big)$ respectively since $N_r(k(x)/k) = q^r + 1$. Their quotient is the L-polynomial of F/k, thus

$$L(t) = \exp\Big(\sum_{r=1}^{d} \frac{q^{-r}}{r}\big(N_r(F/k) - q^r - 1\big)\Big).$$

Since the series in the exp-argument converges absolutely for $t = q^{-1}$ and since the value of the L-polynomial is equal to $h(F/k)/q^g$ for $t = q^{-1}$ its limit coincides with $\log\big(h(F/k)/q^g\big)$. The error term results from the following calculation:

$$\begin{aligned}\left| \sum_{r=d+1}^{\infty} \frac{q^{-r}}{r}\big(N_r(F/k) - q^r - 1\big) \right| \;&\leq\; \frac{2g}{d+1} \sum_{r=d+1}^{\infty} \frac{d+1}{r} q^{-r/2} \\ &\leq\; \frac{2gq^{-\frac{d+1}{2}}}{d+1} \sum_{r=0}^{\infty} (q^{-1/2})^r \leq \frac{2g}{q^{1/2}-1} \cdot \frac{q^{-d/2}}{d+1}.\end{aligned}$$

□

In order to approximate $h(F/k)$ within $2^{-1/2}h(F/k)$ and $2^{1/2}h(F/k)$ as in the end of section 3 it suffices to compute the sum with $d = \max\{\lceil 2\log_q(6g/(q^{1/2}-1))\rceil, 0\}$ since for this d the error term in Theorem 36 is less than $\log(2)/2$, using $2/\log(2) < 3$. In particular, if $d = 0$ then q^g is already an appropriate approximation of $h(F/k)$. By our assumption on the representation of F/k, the required expected time to approximate $h(F/k)$ is polynomial in g and $\log(q)$.

7. Complexity

We can now combine the previous results in order to meet the input requirements of Algorithm 2, to check its output and to estimate its running time.

From Lemma 3 and Theorem 29 we obtain the following theorem yielding the essential complexity of Algorithm 2 and Algorithm 4. As in section 1 we let $c_F = 1$ if $h(F/k)/q^g = L_{q^g}(1/2, o(1))$, and $c_F > 8$ otherwise.

Theorem 37. *Algorithm 2 computes the relation lattice Λ of a generating system $[D_1], \ldots, [D_r]$ of $\mathcal{C}l^0(F/k)$ using a factor basis S of size*

$$s \;=\; L_{q^n}\left(1/2, \left(\frac{c_F + o(1)}{2}\right)^{1/2}\right)$$

and

$$t_1 + t_2 \;=\; L_{q^n}\left(1/2, \left(2c_F + o(1)\right)^{1/2}\right)$$

iterations with probability tending to 1 as g tends to infinity, provided that $n \geq 2g-1$, $n = O(g)$, $r = O(s)$ and $\log(q) = o(n\log(n))$. Under the same assumptions, Algorithm 4 finishes after expected

$$L_{q^n}\left(1/2, \left(\frac{c_F + o(1)}{2}\right)^{1/2}\right)$$

iterations.

Proof. Before we start with the actual running time analysis we consider the expression

$$dm + c(n/m)\log(n/m) \tag{38}$$

where $n, m \in \mathbb{R}$, $c, d \geq 1/2$, $c = O(1)$ and $d = o(n\log(n))$. Then for $n \to \infty$, the m minimising (38) tend to infinity. Let $\rho = 1 + \log(d)/\log(n)$ and denote by β some function assuming positive values. For an arbitrary m of the form

$$m = \beta\left(\frac{n\log(n)}{d}\right)^{1/2} \tag{39}$$

we get

$$c(n/m)\log(n/m) = \left(\frac{c}{2\beta}\left(\rho - \frac{2\log(\beta)}{\log(n)} + o(1)\right)\right)\left(nd\log(n)\right)^{1/2}, \tag{40}$$

and the value of (38) becomes

$$dm + c(n/m)\log(n/m) = \left(\beta + \frac{c}{2\beta}\left(\rho - \frac{2\log(\beta)}{\log(n)} + o(1)\right)\right)\left(nd\log(n)\right)^{1/2}. \tag{41}$$

From $d \geq 1/2$ and $d = o(n\log(n))$ we see that ρ is asymptotically bounded from below and above by positive constants. Let β be such that (39) minimises (38). Then β minimises the big term in the middle of (41). It follows that β is as well asymptotically bounded from below and above by positive constants and hence $\log(\beta)/\log(n) = o(1)$. Computing the β minimising the middle term of (41) explicitly, which is now $\beta + c/(2\beta)(\rho + o(1))$, we obtain

$$\beta = \left(\frac{c\rho + o(1)}{2}\right)^{1/2}. \tag{42}$$

Using (42) and substituting $\log(n) = \rho^{-1}\log(nd)$ in (39) and in the last term of (40) and (41) yields

$$m = \left(\frac{c + o(1)}{2}\right)^{1/2}\left(\frac{n\log(nd)}{d}\right)^{1/2}, \tag{43}$$

$$c(n/m)\log(n/m) = \left(\frac{c+o(1)}{2}\right)^{1/2}\Big(nd\log(nd)\Big)^{1/2} \tag{44}$$

and

$$dm + c(n/m)\log(n/m) = \big(2c+o(1)\big)^{1/2}\Big(nd\log(nd)\Big)^{1/2}. \tag{45}$$

Let $1/2 < \alpha < 1$. We assume now $n \geq 2g-1$, $4\log_q(14g+4)+2 \leq m \leq n^\alpha$ and $m \to \infty$. Let S denote the set of places of degree $\leq m$. The cardinality of S is

$$s = \sum_{d=1}^{m} \frac{q^d}{d}(1+O(gq^{-d/2})) = q^{(1+o(1))m} \tag{46}$$

by the usual estimations of the number of places of degree $\leq m$, $m \geq 4\log_q(14g+4)+2$ and $m \to \infty$. Furthermore, the smoothness probability satisfies

$$p_{S,n} \geq 1/\exp\Big(\big(c+o(1)\big)\Big(\frac{n}{m}\Big)\log\Big(\frac{n}{m}\Big)\Big) \tag{47}$$

with $c = 4/(1-\alpha)$, and if $h(F/k)/q^g \leq \exp\big(o(1)(n/m)\log(n/m)\big)$ then (47) holds also true with $c=1$, according to Theorem 29.

We want to apply Lemma 3 and let $\widetilde{p}_{S,n}$ be the right hand side of (47). By Theorem 34 there are divisors $D_1,\ldots,D_r$ generating $\mathcal{C}l^0(F/k)$ such that $r = O(s)$ and $r \geq 1$. From the definitions it is clear that $s \to \infty$, $\widetilde{p}_{S,n} \to 0$ and $r = o((q^{1/2}+1)^{2g})$ if $g \to \infty$, so the assumptions of Lemma 3 are satisfied. The steps 1–3 and 5–7 of Algorithm 2 are iterated t_1+t_2 times with the t_1, t_2 defined in Lemma 3. Now t_1+t_2 is at least of the order of $\widetilde{p}_{S,n}^{-1}s$. We want to find m such that $\widetilde{p}_{S,n}^{-1}s$ is asymptotically minimal up to a term $1+o(1)$ in the exponent. From (46) and (47) we have

$$\widetilde{p}_{S,n}^{-1}s = \exp\big((1+o(1))(\log(q)m + c(n/m)\log(n/m))\big). \tag{48}$$

With $d = \log(q)$ the logarithm of this value is of the form considered in (38), and the conditions on c, d are satisfied. In order to minimise (48) we choose

$$m = \lceil (c/2 \cdot n\log(n\log(q))/\log(q))^{1/2} \rceil. \tag{49}$$

Since $\log(q) = o(n\log(n))$ is assumed in the theorem we have $m \to \infty$. It follows that m is of the form (43) and that it indeed minimises (48). Furthermore, $4\log_q(14g+4)+2 \leq m \leq n^\alpha$ for sufficiently large g by (39), consistent with our assumption above. If $c_F > 8$ then we can choose $\alpha > 1/2$ such that $c = 4/(1-\alpha) = c_F$. If $c_F = 1$ then

$$h(F/k)/q^g \leq L_{q^g}(1/2, o(1)) \leq \exp\big(o(1)(n/m)\log(n/m)\big)$$

where the first inequality holds by assumption and the second inequality follows from $n \geq 2g-1$ and (44). We can thus take $c = c_F$ also in this case.

Equations (43) and (46) now yield the formula for s as in the theorem. Since $n = O(g)$ and $r = O(s)$ by assumption it follows that in fact $t_1+t_2 = \widetilde{p}_{S,n}^{-1}s^{1+o(1)}$, and so (48) and (45) yield the formula for t_1+t_2.

Lemma 3 applies and we conclude that Algorithm 2 computes the relation lattice Λ as claimed by the theorem. The statement about Algorithm 4 follows from (44) and Lemma 5. □

As discussed in section 4 we would actually expect Theorem 29 and hence Theorem 37 to hold for $n = g$ in many cases, for example when there are rational subfields $k(x)$ such that $[F : k(x)] = O(1)$, since then $|\log(h(F/k)/q^g)| = O(\log(g\log(q)))$.

A substantial part of the overall running time of Algorithm 2 is due to the linear algebra computations involving Λ_S and U. Let M_S be the $s^{1+o(1)} \times (r+s)$ integer matrix whose rows consists of the generators of Λ_S computed in Algorithm 2, and

let M_2 be the matrix obtained from M_S by omitting the first r columns. The rows of M_2 generate Λ_2. As in [8] we can compute a basis of U by means of a Hermite normal form computation applied to M_S, in time $s^{5+o(1)}$. The resulting running time would be as in Theorem 37 but with worse constants in front of c_F. Now every row of M_2 has at most n non zero entries. Since s is much bigger than n this means that M_2 is a sparse matrix so that sparse matrix techniques become applicable. Note that previous techniques to compute the group structure of $\mathcal{C}l^0(F/k)$ all return very dense matrices, which leads to considerably longer running times.

Lemma 50. *Assume that $g, q = s^{o(1)}$ and that r is polynomial in g and q. There is an algorithm which computes a basis of the relation lattice Λ from M_S with success probability at least $1/2$, given the factorisation of $h(F/k)$ and provided that $\Lambda_2 = \mathbb{Z}^s$. Otherwise it outputs failure or a basis of a submodule of Λ. The running time is $s^{2+o(1)}$.*

Proof. There is an algorithm which computes a randomly and uniformly distributed element of the row kernel of the $s^{1+o(1)} \times s$ matrix M_2 modulo $h(F/k)$ with success probability at least $1/2$, given the factorisation of $h(F/k)$ and provided that $\Lambda_2 = \mathbb{Z}^s$. Otherwise it outputs failure or a possibly not uniformly distributed element of the row kernel of M_2. The existence of this algorithm follows from Theorem 3 of [7] and the discussion thereafter, the running time is $O(s(gs^{1+o(1)}+s^{1+o(1)}) \log(s^{1+o(1)}h(F/k))^2 \log_2(h(F/k))^2) = s^{2+o(1)}$ using the notation $n = s^{1+o(1)}$, $d = s$, $q = O(h(F/k))$, $r = s$ and $\omega = O(gs^{1+o(1)})$. More precisely, the algorithm of Theorem 3 of [7] has to be applied to the $O(\log_2(h(F/k)))$ factors of $h(F/k)$ in turn, and for each of these factors it is in fact applied $O(\log_2(\log_2(h(F/k))))$ times to achieve an overall success probability of at least $1/2$.

If x is a randomly and uniformly distributed element of the row kernel of M_2 then omitting the last s columns from xM_S yields a randomly and uniformly distributed relation modulo $h(F/k)$, that is a randomly and uniformly distributed element of $\Lambda/h(F/k)\Lambda$. To see this, view M_S in lower triangular row Hermite normal form. The last s coordinates of x are then congruent to zero mod $h(F/k)$ while the first coordinates without the last s are uniform and random modulo $h(F/k)$ by assumption, and lead to such a linear combination of a basis of Λ.

The cardinality of $\Lambda/h(F/k)\Lambda$ is $h(F/k)^r$ and every proper subgroup has index at least 2. Let $e = \lceil \log_2(h(F/k)^r) \rceil$ and $l = \lceil \log_2(2e) \rceil$. Then sampling le randomly and uniformly distributed elements from $\Lambda/h(F/k)\Lambda$ yields a generating system of $\Lambda/h(F/k)\Lambda$ with probability at least $1/2$, since $(1 - 1/2^l)^e \geq 1 - e/2^l \geq 1/2$. Supplementing representatives of the generating system of $\Lambda/h(F/k)\Lambda$ by the unit vectors of $\mathbb{Z}^r$ times $h(F/k)$ leads to a generating system of Λ. A final Hermite normal form computation then yields a basis of Λ.

The expected running time to compute Λ from M_S given the factorisation of $h(F/k)$ is thus $s^{2+o(1)}$ because r and le are expressions which are polynomial in g and q, and these in turn are in $s^{o(1)}$. □

If we use the technique of Lemma 50 in step 9 of Algorithm 2 together with fast point counting algorithms and rigorous factoring we get a running time of the same asymptotic form as in Theorem 37. We supplement Algorithm 2 as follows.

Algorithm 51. *(Relation search)*

Input: *A divisor D_0 of degree one and divisors $D_1, \ldots, D_r$ of degree zero whose classes generate $\mathcal{C}l^0(F/k)$. Divisors $E_1, \ldots, E_t$ of degree zero.*

Output: *The relation lattice Λ of the $[D_j]$ and integers $\lambda_{i,j} \in \mathbb{Z}$ such that $[E_i] = \sum_{j=1}^r \lambda_{i,j}[D_j]$.*

1. *Compute the zeta function of F/k using the algorithms of* [15, 24]. *Compute the factorisation of the class number.*
2. *Let $n = 2g - 1$. Find m such that $t_1 + t_2 + s^2$ is minimal and $s \geq \exp(\log(gq)^2)$ where t_1, t_2 are from Lemma 3 with r replaced by $r + t$, s is the number of all places of degree $\leq m$ and $\widetilde{p}_{S,n} = p_{S,n}$. Here s and $p_{S,n}$ are computed exactly using the values $N_d(F/k)$ given by the zeta function, Lemma 7 and Theorem 31.*
3. *Run Algorithm 2 on input of D_0 and $D_1, \ldots, D_r, E_1, \ldots, E_t$. The linear algebra in Algorithm 2 is done using the method of Lemma 50. If this fails rerun Algorithm 2. Algorithm 2 returns a basis of a submodule U_1 of the relation lattice Λ_1 of the $[D_j]$ and $[E_i]$.*
4. *A submodule U of the the relation lattice Λ of the $[D_j]$, the index $(\mathbb{Z}^r : U)$ and the $\lambda_{i,j}$ are obtained by computing the Hermite normal form of the basis of U_1. If some $[E_i]$ cannot be expressed in terms of the $[D_j]$ then go to step 3.*
5. *If $(\mathbb{Z}^r : U) \neq h(F/k)$ then go to step 3.*
6. *Output U and the $\lambda_{i,j}$.*

The divisors $D_0, D_1, \ldots, D_r$ can be computed by Algorithm 33 and Algorithm 35 in expected time polynomial in g and q, according to section 5. Clearly, r and the degrees of $(D_i)_0$ and $(D_i)_\infty$ are polynomial in g and q. Recall that we have assumed that F/k is represented by an absolutely irreducible plane affine curve of degree $O(g)$.

Theorem 52. *The expected running time of Algorithm 51 is*

$$L_{q^n}\left(1/2, (2c_F + o(1))^{1/2}\right),$$

under the assumptions of Theorem 37 and if the numbers r and t of the D_i and E_j and the degrees of $(D_i)_0$, $(D_i)_\infty$, $(E_j)_0$ and $(E_j)_\infty$ are polynomial in n and q.

Proof. In step 1 the class number and in fact the zeta function of F/k can be computed in time polynomial in g, $\log(q)$ and the characteristic, which is polynomial in g and q. This follows from [15], Corollary 3, since we have assumed that the function field is represented by an absolutely irreducible plane affine curve of degree $O(g)$.

The class number is then factored in provably expected time

$$L_{h(F/k)}\big(1/2, 1 + o(1)\big)$$

using the class group factoring method [16]. From $h(F/k) \leq (q^{1/2} + 1)^{2g}$, $q \geq 2$, $(q^{1/2} + 1)^2 \leq 3q$ and $(1 + \log(3)/\log(q))/2 < 1.3$ we obtain

$$\begin{aligned}\log(h(F/k)) &\leq g \log(3q) \\ &\leq (1/2 + o(1))(1 + \log(3)/\log(q)) \cdot n \log(q) \\ &\leq 1.3 \cdot n \log(q).\end{aligned}$$

Then

$$\begin{aligned}\log(h(F/k)) \log(\log(h(F/k))) &\leq 1.3\, n \log(q) \log(1.3 n \log(q)) \\ &\leq (1.3 + o(1))\, n \log(q) \log(n \log(q)),\end{aligned}$$

so factoring the class number is bounded by

$$L_{q^n}\big(1/2, 1.3 + o(1)\big). \tag{53}$$

The computation of s and $p_{S,n}$ in step 2 takes time polynomial in g and $\log(q)$, the latter according to the remarks after Theorem 31. By Theorem 37 an upper bound for $t_1 + t_2$, s and hence for $t_1 + t_2 + s^2$ is given in the form $L_{q^n}(1/2, O(1))$.

In step 3, creating the factor base S takes s times a number of operations in k which is polynomial in g and $\log(q)$, sorting it with respect to some suitable order no more than a further s^2 operations. All single steps of Algorithm 2 then require an amount of operations polynomial in g and $\log(q)$, except step 9, which requires $s^{2+o(1)}$ operations according to Lemma 50. Note that $s \geq \exp(\log(gq)^2)$ and $r+t$ is polynomial in g and q so Lemmas 3 and 50 are applicable. The first part of step 3 computes U_1 with $U_1 = \Lambda_1$ with probability at least $1/2 + o(1)$. Hence we expect only a constant number of repetitions of step 3 until $U_1 = \Lambda_1$.

In step 4 the Hermite normal form of a matrix is computed with entries of size $h(F/k)$ and of dimension $(r+t) \times (r+t)$ where $r+t$ is polynomial in g and q. This requires a time polynomial in g and q as well.

Polynomial expressions in g and q are in $L_{q^n}(1/2, o(1))$, because $n \geq 2g-1$ and $\log(q) = o(n \log(n))$. Using $c = c_F$ and $d = \log(q)$, the overall running time of Algorithm 51 is the sum of the running time for factoring and $t_1 + t_2 + s^{2+o(1)}$ disregarding cofactors polynomial in g or q. With (39) an upper bound for $t_1 + t_2 + s^{2+o(1)}$ is given by the sum of the exponential of (41) (with $\log(\beta)/\log(n) = o(1)$) and the square of (46). This bound is asymptotically minimised for β such that both summands become of equal size. Thus we require

$$L_{q^n}\left(1/2, \rho^{-1/2}\left(\beta + \frac{c_F(\rho + o(1))}{2\beta}\right)\right) = L_{q^n}\left(1/2, \rho^{-1/2} \cdot 2\beta\right)$$

using (39), (41) and (46), hence

$$\beta + \frac{c_F \rho}{2\beta} = 2\beta.$$

This is solved by $\beta = (c_F\rho/2)^{1/2}$ as in Theorem 37, yielding the same running time bounds. In view of (53) the contribution of factoring is then negligible as $1.3 < \sqrt{2}$. □

Theorem 52 says that the running time of Algorithm 2 is subexponential of exponent $1/2$ in $g\log(q)$, while the size of the representation of the group $\mathcal{C}l^0(F/k)$ via the defining plane affine curve of degree $O(g)$ is $O(g^2 \log(q))$. From [14, Theorem 2, 1)] and $q \geq 2$ we obtain

$$\begin{aligned} \log(h(F/k)) &\geq \log\Big(q^g \cdot \frac{(q-1)^2}{q(q+1)} \cdot \frac{1}{g+1}\Big) \\ &\geq g\log(q)\Big(1 - \frac{\log(6) + \log(g+1)}{g\log(q)}\Big) \\ &\geq (1+o(1))g\log(q). \end{aligned} \tag{54}$$

The running time bound of Theorem 52 for $n = 2g$ thus becomes

$$L_{h(F/k)}\left(1/2, 2c_F^{1/2} + o(1)\right),$$

and this is subexponential of exponent $1/2$ in the group size $h(F/k)$.

If we want to avoid computing the zeta function of F/k in Algorithm 51 we can proceed as follows. In step 3 we find the m which minimises $t_1 + t_2 + s^5$ using $\pi(m) \approx q^m/m$ and the bounds for $p_{S,n}$ of Theorem 29 (also see the proof of Theorem 52). For the linear algebra in step 3 we use the Hermite normal form and in step 5 we compute an approximation h of the class number as in section 6 and return Λ if and only if $2^{-1/2}h < (\mathbb{Z}^r : \Lambda) < 2^{1/2}h$. This modification of Algorithm 51 has an expected running time of $L_{q^n}(1/2, 5(c_F/8)^{1/2} + o(1))$.

In Algorithm 4 we can make similar use of the fast linear algebra technique of Lemma 50. Note that Algorithm 4 requires that $\Lambda_2 = \mathbb{Z}^s$, a condition which holds with probability tending to one but which appears difficult to check with less cost

than Algorithm 51. Thus during Algorithm 4 it may be necessary at some stage to run Algorithm 2 again. The running time of Algorithm 4 is otherwise dominated by $s^{2+o(1)}$ from the linear algebra part, and is thus asymptotically equal to that of Algorithm 51. After successfully running Algorithm 4 for all $P_i - \deg(P_i)D_0$ as an (expensive) precomputation its running time can be lowered to that given in Theorem 37, which is $s^{1+o(1)}$.

Finally, if we take $n = g$ on heuristic grounds then the running time for factoring $h(F/k)$ becomes the most expensive part in the running time analysis of Algorithm 51. It would then be better to use the number field sieve in the factoring step since its running time is heuristically much faster and could be ignored again. Alternatively, it might be possible to generalise Lemma 50 to the case where the factorisation of $h(F/k)$ is unknown.

8. Applications

The Algorithms 51 and 4 have a variety of important applications such as the computation of discrete logarithms, of S-units and of the structure of the divisor class group of F/k or of the ideal class groups of the Dedekind rings $\mathfrak{o}^S$ of algebraic functions whose poles are restricted to a finite set of places S.

Class representations and discrete logarithms. Using Algorithm 51 under the requirements of Theorem 52 we obtain the structure of the divisor class group as a finite abelian group in elementary divisor form

$$\mathcal{C}l^0(F/k) \cong_{\phi_1} \mathbb{Z}^r/\Lambda \cong_{\phi_2} \mathbb{Z}/c_1\mathbb{Z} \times \cdots \times \mathbb{Z}/c_m\mathbb{Z}$$

by computing the Smith normal form of a basis matrix of Λ, where $c_i \in \mathbb{Z}$, $c_i | c_j$ for $i \leq j$ and $1 \leq m \leq 2g$. Note that this can be done very efficiently since such a basis matrix has coefficients of size $h(F/k)$ and dimension $r \times r$ where r is polynomial in g and q. Images and preimages under the second isomorphism ϕ_2 can be computed without difficulty using the transformation matrices giving the Smith normal form. Preimages under the first isomorphism ϕ_1 are also easy because $D_1, \ldots, D_r$ are known. A more difficult question is to determine the image of an arbitrary divisor class $[D] \in \mathcal{C}l^0(F/k)$ under first isomorphism ϕ_1, that is finding a class representation of $[D]$ in terms of the $[D_i]$. For this we use Algorithm 4 or Algorithm 51.

From the map $\phi_2 \circ \phi_1$ and the divisor of degree zero D_0 we finally obtain the computable isomorphism

$$\phi : \mathcal{C}l(F/k) \longrightarrow \mathbb{Z}/c_1\mathbb{Z} \times \cdots \times \mathbb{Z}/c_m\mathbb{Z} \times \mathbb{Z}.$$

Using ϕ we can translate questions corresponding the divisor class group into the context of explicitly given, finitely generated abelian groups.

For two divisor classes $[D_1], [D_2]$ the discrete logarithm of $[D_1]$ with respect to $[D_2]$, if it exists, is a number $r \in \mathbb{Z}$ determined modulo the order of $[D_2]$, for which $[D_1] = r[D_2]$ holds. By determining the class representations $\phi([D_1])$ and $\phi([D_2])$ of $[D_1]$ and $[D_2]$ we can translate the problem of computing the discrete logarithm of $[D_1]$ with respect to $[D_2]$ into the analogous problem in $\mathbb{Z}/c_1\mathbb{Z} \times \cdots \times \mathbb{Z}/c_m\mathbb{Z} \times \mathbb{Z}$ where it can be solved instantly because of the known entities c_i. The expected running time to compute a discrete logarithm is hence as in Theorem 52.

S-Units and S-class groups. Let S be an arbitrary finite set of places of F/k and let $\langle S \rangle$ be the subgroup of $\mathcal{D}(F/k)$ generated by S. We define $\mathcal{P}(S) = \langle S \rangle \cap \mathcal{P}(F/k)$, the S-unit group $U(S) = \{ z \in F^\times \mid (z) \in \langle S \rangle \}$ and the S-class group $\mathcal{C}l(S) = \mathcal{D}(F/k)/(\langle S \rangle + \mathcal{P}(F/k))$. Let $\mathfrak{o}^S$ be the ring of all elements of F which are integral outside S. Then $\mathfrak{o}^S$ is a Dedekind ring and for non-empty S we have $(\mathfrak{o}^S)^\times = U(S)$

and $\mathcal{C}l(\mathfrak{o}^S) \cong \mathcal{C}l(S)$ where $(\mathfrak{o}^S)^\times$ and $\mathcal{C}l(\mathfrak{o}^S)$ denote the unit group and ideal class group of $\mathfrak{o}^S$ respectively.

The question is how to compute generators and relations for $U(S)$ and $\mathcal{C}l(S)$. By means of the class representation map ϕ this can also be translated to the situation of finitely generated abelian groups. In the following we assume that the required algorithms for finitely generated abelian groups and homomorphisms between them are available.

The restriction of the residue class homomorphism $\phi_S : \langle S\rangle \longrightarrow \mathcal{C}l(F/k)$ has kernel $\ker \phi_S = \mathcal{P}(S)$ and cokernel $\operatorname{coker} \phi_S = \mathcal{C}l(F/k)/\phi_S\big(\langle S\rangle\big) \cong \mathcal{C}l(S)$. Furthermore, $\mathbb{Z}^S \cong \langle S\rangle$ and $\mathcal{C}l(F/k) \cong \mathbb{Z}/c_1\mathbb{Z} \times \cdots \times \mathbb{Z}/c_m\mathbb{Z} \times \mathbb{Z}$, so combining ϕ_S with these isomorphisms we get an explicitly given homomorphism $\phi'_S : \mathbb{Z}^S \to \mathbb{Z}/c_1\mathbb{Z} \times \cdots \times \mathbb{Z}/c_m\mathbb{Z} \times \mathbb{Z}$ of which we can compute the kernel $\ker \phi'_S \cong \mathcal{P}(S)$ and the cokernel $\operatorname{coker} \phi'_S \cong \mathcal{C}l(S)$. From a basis of the principal divisor group $\mathcal{P}(S)$ we can get fundamental S-units in an unevaluated product representation by means of the principal divisor test in [12], in time polynomial in g, $\log(q)$ and $|S|$. Using $U(S) \cong k^\times \times \mathcal{P}(S)$ we thus obtain a generating system for $U(S)$ in $F^\times$. The computation of $U(S)$ and $\mathcal{C}l(S)$ can hence be done in essentially one run of Algorithm 51 with $E_i = P_i - \deg(P_i)D_0$ where the P_i are the places in S, or one run of Algorithm 51 with no E_i and $|S|$ runs of Algorithm 4.

S-Class group exact sequence. The S-unit group and S-class group are linked by the exact sequence

$$1 \to U(S) \to F^\times \to \mathcal{D}(F/k)/\langle S\rangle \to \mathcal{C}l(S) \to 0.$$

We have $U(S) \cong \mathbb{Z}/(q-1)\mathbb{Z} \times \mathbb{Z}^{|S|-1}$, where $\mathbb{Z}^0 = \mathbb{Z}^{-1} = 0$, and $\mathcal{C}l(S) \cong \mathbb{Z}/c_1\mathbb{Z} \times \cdots \times \mathbb{Z}/c_n\mathbb{Z}$ for some $c_i \in \mathbb{Z}$ with $c_i | c_j$ for $i \le j$ and $1 \le n \le 2g+1$. Here $c_n = 0$ if and only if S is empty. Substituting this into the exact sequence yields

$$0 \to \mathbb{Z}/(q-1)\mathbb{Z} \times \mathbb{Z}^{|S|-1} \to F^\times \to \mathcal{D}(F/k)/\langle S\rangle \to \mathbb{Z}/c_1\mathbb{Z} \times \cdots \times \mathbb{Z}/c_n\mathbb{Z} \to 0.$$

We denote the three maps in the middle by f_1, f_2 and f_3.

Combining the results of the previous two sections we can essentially compute images and preimages under every f_i. We make a few brief comments how this is achieved. Images under f_3 are computed by smoothing divisors as in the previous sections, and preimages are computed using (precomputed) divisors corresponding to the cyclic factors of the codomain of f_3. Thus computing images under f_3 is expensive while computing preimages is efficient. Images under f_2 are computed by taking principal divisors, which is essentially factoring polynomials over finite fields. For preimages we proceed as follows. Let D be a representative of a class in $\mathcal{D}(F/k)/\langle S\rangle$. We map D into $\mathcal{C}l(F/k)$ and check whether it is in the image of $\langle S\rangle$ in $\mathcal{C}l(F/k)$. If no, $D + \langle S\rangle$ does not contain a principal divisor and there is no preimage under f_2. Otherwise using the representation of D in terms of S in $\mathcal{C}l(F/k)$ we find $D' \in D + \langle S\rangle$ which is principal. The preimage under f_2 can then be computed using the principal divisor test from [12]. If $|S| \le 1$ we can avoid mapping D to $\mathcal{C}l(F/k)$ and simply (try to) find D' such that $\deg(D') = 0$ and $D' \in D + \langle S\rangle$. Thus computing images under f_2 is efficient while computing preimages under f_2 is expensive unless $|S| \le 1$. Images under f_1 are computed using a basis (generating system) of the S-units as elements in $F^\times$. Computing preimages under f_1 requires the computation of the principal divisor of the given element, checking that the support of that divisor is a subset of S, some linear algebra and taking a discrete logarithm in $k = \mathbb{F}_q$. If $|S| \le 1$ we can directly check whether the given element is constant, and take the discrete logarithm. Thus computing images and preimages under f_1 is efficient since q is small in comparison to the subexponential running time of Theorem 52 and hence discrete logarithms in $k = \mathbb{F}_q$ are easy. We remark

that in all of the above it is best to allow a representation of elements of $F^\times$ in form of unevaluated products, since otherwise one has in the end easily to deal with polynomials in $k[x]$ of degree exponential in g.

9. Practice

In this section we consider some practical issues of the relation search. We have implemented simplified variants of the above relation search algorithms and the algorithms of section 8 in the computer algebra systems Kash [13] and Magma [3, 4] with the emphasis not on asymptotical speed but on heuristic practicality for small to medium sized genera and small constant field sizes. In particular the asymptotic running time of the algorithm given below is worse than that of Theorem 52. Section 10 contains some examples illustrating the behaviour of our implementation.

The basic observation is that steps 2 and 6 of Algorithm 2 and step 2 of Algorithm 4 are fairly expensive operations albeit polynomial in g and $\log(q)$. Also, taking $n \geq 2g-1$ makes the running time much worse than taking $n \approx g$ on heuristic grounds, as indicated in section 4. To improve on these points we essentially adopt the following strategy.

We fix a divisor A of degree $\deg(A) > 0$ which is small in comparison to the genus, and ensure that the factor basis S contains all the places in the supports of the D_i, the support of A and the support of the pole divisor $(x)_\infty$ of the fixed separating element x. We then take small random linear combinations D of the places P_i, and compute $r \in \mathbb{Z}$ minimal such that the class $[rA+D]$ contains at least one effective divisor E, by means of the divisor reduction in [12]. We also compute $c \in F^\times$ such that $E = (c) + rA + D$. In addition to E we now consider the other effective divisors in $[rA + D]$ as well. These are the form $(a) + E$ with $a \in \mathcal{L}(E)$ and $a \neq 0$. Observe that $(a) + E = (b) + E$ precisely if $a/b \in k$. Once a basis of $\mathcal{L}(E)$ is computed, the generation of the representatives a modulo multiplication by elements in $k^\times$ is fast, as it involves only additions and scalar multiplications. In order to check whether the divisor $(a) + E$ is S-smooth we first compute the (reduced) norm $\mathrm{N}_{F/k(x)}(a)$ and check that its numerator does not contain a prime factor corresponding to places outside S. If $(a)+E$ does not pass this test then it is not smooth since $(a) + E$ contains places above all prime factors of the numerator of $\mathrm{N}_{F/k(x)}(a)$. Otherwise we can simply compute the full factorisation of $(a) + E$. Most non-smooth $(a) + E$ will not pass the first test because most places of S have relative residue degree one over $k(x)$. Now $(a) + (c)$ is smooth if and only if $(a) + E$ is smooth, because $(a) + E = (a) + (c) + rA + D$ and $rA + D$ is smooth. From every smooth $(a) + E$ we thus obtain a relation between places of S in the form (ac).

Since r is chosen minimal we have that $\dim(E) = \dim(rA + D) \leq \deg(A)$ and $\deg(E) < g + \deg(A)$ by the theorem of Riemann-Roch, so there are at most $(q^{\deg(A)} - 1)/(q - 1)$ effective divisors in $[E] = [rA + D]$. We thus consider effective divisors of degree $\leq n = g + \deg(A) - 1$ rather than $n \geq 2g - 1$, and for every E computed from $rA + D$, we obtain at most $(q^{\deg(A)} - 1)/(q - 1)$ further effective divisors in an efficient way. As a variation we could also simply choose r such that $g \leq \deg(rA+D) < g+\deg(A)$. Then $[rA+D]$ contains at least one effective divisor. While we expect that the number of effective divisors in $[rA + D]$ is approximately $(q^{\deg(A)} - 1)/(q - 1)$, it could now also be larger. The computation of the minimal r and of a basis of $D + rA$ can be carried out very efficiently for $A = (x)_\infty$. We will use this divisor only if $[F : k(x)]$ is small in comparison to g, because otherwise the dimension of $D + rA$ may be too large since $\deg(A) = [F : k(x)]$. We remark that using $A = (x)_\infty$ means that we are looking for algebraic functions which have small degrees when expressed in x and y.

The factor basis is created as follows. Let $n = g + \deg(A) - 1$ and d be the maximum degree for which we have to compute $N_d(F/k)$ during Algorithm 33 and 35 and during the approximation of the class number by Theorem 36 with error interval $2^{-1/2}, 2^{1/2}$. For $1 \leq r \leq d$ we compute $\widetilde{\pi}_r = \pi(r)$ by Moebius inversion from the $N_r(F/k)$, see [23, p. 178]. For $r > d$ we define $\widetilde{\pi}(r) = q^r/r$. We then find m such that $1 \leq m \leq n^{0.7}$ and the value

$$g^2 \left[F : k(x)\right]^5 \left(\psi(n,n)/\psi(n,m)\right) \left(\sum_{r=1}^{m} \widetilde{\pi}_r\right) + \left(\sum_{r=1}^{m} \widetilde{\pi}_r\right)^3$$

becomes minimal but unequal to zero. This expression is meant to be a rough approximation to the overall running time if a factor basis up to degree m is used.

The computation of the ψ-values is done as described after Theorem 31, with the $\widetilde{\pi}(r)$ instead of the exact, but unknown $\pi(r)$. We then define $m_S = m$, m_g the maximum d in Algorithm 33 and 35, m_h the maximum d occurring during the approximation of the class number by Theorem 36 with error interval $2^{-1/2}, 2^{1/2}$, and $m_B = \max\{m_S, m_g\}$. The factor basis is defined as

$$S := \mathcal{P}l^{\leq m_B}(F/k) \cup \operatorname{supp}(D_0) \cup \operatorname{supp}\big((x)_\infty\big) \cup \operatorname{supp}(A).$$

We expect a smoothness probability of about $\widetilde{p}_{S,n} := \psi(n, m_B)/\psi(n,n)$.

For the relation search the following algorithm has proven useful in practice. Let $\mathbb{Z}_0^s$ denote the sublattice of $\mathbb{Z}_0^s$ whose vectors are orthogonal to $(1, \ldots, 1)$.

Algorithm 55. *(Relation search)*

Input: *The factor basis S, the divisor A, the approximation h of the class number and the expected smoothness probability $\widetilde{p}_{S,n}$.*

Output: *The relation lattice of the places of S in the divisor class group.*

1. *(Initialisation) Write $S := \{P_1, \ldots, P_s\}$ and put $W := 1$. For each $1 \leq i \leq s$ choose a random $P \in S$, define $B_i := -P_i + P + rA$ with a minimal $r \in \mathbb{Z}$ such that $\dim(B_i) > 0$, choose a non-zero $c_i \in \mathcal{L}(B_i)$ and compute $C_i = (c_i) + B_i$ and a basis of $\mathcal{L}(C_i)$. Define $RV := 0$ (number of attempts to find relations), $R := 0$ (number of found relations), $RV_0 := 0$, $R_0 := 0$, $\Delta RV := \max\{10\lceil \widetilde{p}_{S,n}^{-1} \rceil, 20\}$, $\Delta R := \max\{\lceil s/10 \rceil, 10\}$, $S_R := 1$ (increase of rank relative to increase of number of relations), $R_{RV} := 1$ (relations relative to attempts to find relations), $i := 1$.*
2. *(Loop over i) After step 9 go to step 2.*
3. *(Next divisor) If $\mathcal{L}(C_i)$ multiplicatively modulo $k^\times$ contains only already used elements, a new B_i and C_i are generated: Define $B_i := -P_i + ($ sum of W randomly chosen places of S with random exponents in $[1, W]\,) + rA$, with a minimal $r \in \mathbb{Z}$, so that $\dim(B_i) > 0$. Choose a non-zero $c_i \in \mathcal{L}(B_i)$ and compute $C_i = (c_i) + B_i$. Compute a basis of $\mathcal{L}(C_i)$.*
4. *(Next element) Let a be an element of $\mathcal{L}(C_i)$ not yet used and put $RV := RV + 1$.*
5. *(Relation found?) Test if $(a) + C_i$ is smooth. If yes, store (a, c_i) for later evaluation and put $R := R + 1$.*
6. *(Smoothness test) If $RV - RV_0 \geq \Delta RV$, the following steps are performed:*
   6.1. *Put $R_{RV} := R/RV$.*
   6.2. *(Enlarge the factor base?) If $R_{RV} \leq \widetilde{p}_{S,n}/100$, the factor base is extended by the places of the next degree (as already used), $\widetilde{p}_{S,n}$ is adapted and the new B_j and ΔRV are computed as in the initialisation. Otherwise set $\Delta RV := \max\{10\lceil 1/R_{RV} \rceil, 20\}$.*
   6.3. *Put $RV_0 := RV$.*
7. *(Relation test) If $R - R_0 \geq \Delta R$, the following steps are performed:*

7.1. *(Evaluate relations) Compute the vectors* $(v_{P_j}(a) + v_{P_j}(c_i))_{1\le j\le s}$ *for the stored tuples* (a, c_i)*, reusing information on* c_i *if it occurs for several* a*. Define* U *to be the sublattice generated those vectors and the previous vectors.*

7.2. *(Finite index) The following is done for finite index* $(\mathbb{Z}_0^s : U)$*: If* $2^{-1/2}h < (\mathbb{Z}_0^s : U) < 2^{1/2}h$*, the loop of step* 2 *is left. If the index and* W *have stayed unchanged with the last* $\max\{\lceil s/10\rceil, 10\}$ *relations, then* $W := \min\{W + 1, s, 10\}$ *and all* B_j, C_j *are recomputed as in step* 3*. Finally define* $\Delta R := \lceil 2R_{RV}s/[F : k(x)]\rceil$*.*

7.3. *(Non-finite index) If the index was not yet finite, the following is done: Let* Δr *be the increase of the rank with the last* ΔR *relations. Put* $S_R := (\Delta r/\Delta R + S_R)/2$*. If* $S_R < 0.2$ *is true and the rank and* W *have stayed unchanged with the last* $\max\{s/10, 10\}$ *relations, then let* $W := \min\{W + 1, s, 10\}$*, all* B_j *are recomputed as in step* 3 *and put* $\Delta R := \max\{\lceil s/10\rceil, 10\}\rceil$*. Otherwise let*

$$\Delta R := \lceil \min\{\, 3S_R\Delta R,\ 2R_{RV}s(s - \operatorname{rank}(U))/(S_R[F : k(x)])\,\}\rceil.$$

8. *(Next* i*) If a relation was found in step* 5*, put* $i := i + 1$*. If* $i > s$*, then put* $i := 1$*.*
9. *(End of the loop over* i*) Go to step* 2*.*
10. *(End) The relation search is finished. Output of the found relations and* U*. Terminate.*

Note that since the D_i are S-smooth, the relations between the D_i in the divisor class group can be computed from the output of Algorithm 55 using the Hermite normal form algorithm. In order to express divisor classes $[E_j]$ in terms of the $[D_i]$ we apply a variant of Algorithm 4, see also the remarks at the end of section 7.

Step 6 is based on the experience that the relation search occasionally cannot find enough relations for a very small factor basis or for very small q and g respectively, so that an extension of the factor basis becomes necessary. However, this occurs quite rarely. If one wants to perform the relation search for large g without extending the factor basis, step 6 can be omitted.

Step 7 realises the following strategy: Depending on ΔR, the relations are evaluated frequently in the beginning and less frequently later in the relation search, if the rank is increasing well enough. Thus we can avoid the frequent computation of the Smith normal form in step 7, which gets rather expensive for larger factor bases. Finally, the evaluation of the relations is done when a full rank can be expected by predicting the likely increase of the rank based on the previous growth rate. On the other hand, if the rank is not increasing well enough, the relations are evaluated more frequently and possibly W is increased. The number ΔR of the relations to be found until the next evaluation also includes the approximate ratio of the cost needed to find a relation and the cost needed to compute the Smith normal form: If the computation of the Smith normal form is rather expensive, a rather large number of relations are generated until the next evaluation, and vice versa.

If a factor basis is used which is not known to generate the class group the output of Algorithm 55 may be a proper sublattice of the relation lattice. Let us denote the output by U. It is possible to deterministically enlarge U to the full relation lattice by considering all overlattices $\mathbb{Z}v + U$ of U in $\mathbb{Z}_0^s$ for some $v \in \mathbb{Z}_0^s$ which are of prime index l dividing $(\mathbb{Z}_0^s : U)$, and by checking whether v corresponds to a principal divisor. Note that such v modulo U are elements of order l in $\mathbb{Z}_0^s/U$. This strategy is thus in general exponential in g, since the l-rank of $\mathbb{Z}_0^s/U$ may be bounded from below by $2g$ for every sublattice of the relation lattice and $\gcd(l, q) = 1$, for function

fields over sufficiently large constant fields for example. On the other hand, if $\mathbb{Z}_0^s/U$ is cyclic then this strategy is indeed quite efficient.

10. Examples

In the following we give tables containing some sample runs of our implementation. We use the following additional notation. The structure of the class group is given via the orders of the cyclic factors in elementary divisor form. For a divisor $A = \sum_{i=1}^{r} e_i P_i$ we write $A \sim \big(e_1, \deg(P_1); \ldots; e_r, \deg(P_r)\big)$. For the number RV of attempts to find a relation we provide the according to $\widetilde{p}_{S,n}$ expected ($=_?$) and the actual values. We denote the number of computations of Riemann-Roch spaces by RR. The running time T is with respect to a 600MHz SunBlade 1000 computer.

$q = 2, \quad g = 50$	1
$h(F/\mathbb{F}_2) = 1743271585380988, \quad \mathcal{C}l^0(F/\mathbb{F}_2) \cong 1743271585380988.$	$T = 40\,\text{min}$
$F := \mathbb{F}_2(x,\rho): \quad \rho^2 + (x+1)\rho + x^{101} + x + 1 = 0.$	
$A = (x)_\infty \sim (2,1).$	
$m_g = 14, \quad m_h = 12, \quad m_S = 12 \;\rightarrow\; m_B := 10, \quad \lvert S\rvert = 250, \quad RV =_? 83985.$	
$RR = 79940, \quad RV = 98342, \quad R = 1057 \;\rightarrow\; R/RV = 0.011, \quad \lvert S\rvert/R = 0.24.$	
$W = 5.$	

$q = 3, \quad g = 30$	2
$h(F/\mathbb{F}_3) = 205217259503652, \quad \mathcal{C}l^0(F/\mathbb{F}_3) \cong 2 \times 102608629751826.$	$T = 60\,\text{min}$
$F := \mathbb{F}_3(x,\rho): \quad \rho^2 + (x+1)\rho + x^{62} + x + 1 = 0.$	
$A = (x)_\infty \sim (1,2).$	
$m_g = 8, \quad m_h = 7, \quad m_S = 7 \;\rightarrow\; m_B := 6, \quad \lvert S\rvert = 230, \quad RV =_? 74270.$	
$RR = 80377, \quad RV = 130973, \quad R = 1127 \;\rightarrow\; R/RV = 0.0086, \quad \lvert S\rvert/R = 0.2.$	
$W = 5.$	

$q = 5, \quad g = 19$	3
$h(F/\mathbb{F}_5) = 16563730252090, \quad \mathcal{C}l^0(F/\mathbb{F}_5) \cong 16563730252090.$	$T = 28\,\text{min}$
$F := \mathbb{F}_5(x,\rho): \quad \rho^3 + x\rho + x^{21} + x + 1 = 0.$	
$A \sim (1,1).$	
$m_g = 5, \quad m_h = 4, \quad m_S = 5 \;\rightarrow\; m_B := 4, \quad \lvert S\rvert = 219, \quad RV =_? 63731.$	
$RR = 106132, \quad RV = 126754, \quad R = 909 \;\rightarrow\; R/RV = 0.0072, \quad \lvert S\rvert/R = 0.24.$	
$W = 4.$	

$q = 7, \quad g = 14$	4
$h(F/\mathbb{F}_7) = 1322299613348, \quad \mathcal{C}l^0(F/\mathbb{F}_7) \cong 2 \times 661149806674.$	$T = 7\,\text{min}$
$F := \mathbb{F}_7(x,\rho): \quad \rho^5 + (x+1)\rho + x^8 + x + 1 = 0.$	
$A = (x)_\infty \sim (5,1).$	
$m_g = 4, \quad m_h = 3, \quad m_S = 4 \;\rightarrow\; m_B := 3, \quad \lvert S\rvert = 148, \quad RV =_? 102364.$	
$RR = 14923, \quad RV = 90844, \quad R = 468 \;\rightarrow\; R/RV = 0.0052, \quad \lvert S\rvert/R = 0.32.$	
$W = 2.$	

$q = 13, \quad g = 10$	5
$h(F/\mathbb{F}_{13}) = 206665304791, \quad \mathcal{C}l^0(F/\mathbb{F}_{13}) \cong 206665304791.$	$T = 8\,\text{min}$
$F := \mathbb{F}_{13}(x,\rho): \quad \rho^5 + (x+1)\rho + x^6 + x + 1 = 0.$	
$A \sim (1,1).$	
$m_g = 3, \quad m_h = 2, \quad m_S = 2 \;\rightarrow\; m_B := 2, \quad \lvert S\rvert = 111, \quad RV =_? 10848.$	
$RR = 19116, \quad RV = 20802, \quad R = 395 \;\rightarrow\; R/RV = 0.019, \quad \lvert S\rvert/R = 0.28.$	
$W = 2.$	

$q = 17, \quad g = 10$	6
$h(F/\mathbb{F}_{17}) = 2231475497166, \quad \mathcal{C}l^0(F/\mathbb{F}_{17}) \cong 2231475497166.$	$T = 80\,\text{min}$
$F := \mathbb{F}_{17}(x, \rho): \quad \rho^5 + (x+1)\rho + x^6 + x + 1 = 0.$	
$A \sim (1, 1).$ $m_g = 2, \quad m_h = 2, \quad m_S = 2 \;\rightarrow\; m_B := 2, \quad \lvert S\rvert = 168, \quad RV =_? 33990.$ $RR = 49690, \quad RV = 52399, \quad R = 666 \;\rightarrow\; R/RV = 0.013, \quad \lvert S\rvert/R = 0.25.$ $W = 2.$	

$q = 23, \quad g = 10$	7
$h(F/\mathbb{F}_{23}) = 37953554676269, \quad \mathcal{C}l^0(F/\mathbb{F}_{23}) \cong 37953554676269.$	$T = 52\,\text{min}$
$F := \mathbb{F}_{23}(x, \rho): \quad \rho^5 + (x+1)\rho + x^6 + x + 1 = 0.$	
$A \sim (1, 1).$ $m_g = 2, \quad m_h = 2, \quad m_S = 2 \;\rightarrow\; m_B := 2, \quad \lvert S\rvert = 322, \quad RV =_? 81501.$ $RR = 148690, \quad RV = 154883, \quad R = 1113 \;\rightarrow\; R/RV = 0.0072, \quad \lvert S\rvert/R = 0.29.$ $W = 2.$	

$q = 25, \quad g = 10$	8
$h(F/\mathbb{F}_{25}) = 147510773172045, \; \mathcal{C}l^0(F/\mathbb{F}_{25}) \cong 3 \times 3 \times 3 \times 5463361969335.$	$T = 5\,\text{h}$
$F := \mathbb{F}_{25}(x, \rho): \quad \rho^5 + (x+1)\rho + x^6 + x + 1 = 0.$	
$A \sim (1, 1).$ $m_g = 2, \quad m_h = 2, \quad m_S = 2 \;\rightarrow\; m_B := 2, \quad \lvert S\rvert = 375, \quad RV =_? 47170.$ $RR = 93274, \quad RV = 1544153, \quad R = 1053 \;\rightarrow\; R/RV = 0.00068, \quad \lvert S\rvert/R = 0.36.$ $W = 5.$	

In the following we let S_0 be the set of poles of the separating element $x \in F$, specified by the defining equation. The entries $R(S_0)$, $h(S_0)$ and $\mathcal{C}l(\mathfrak{o}^{S_0})$ denote the regulator, the ideal class number and the structure of the ideal class group of the ring of algebraic functions $\mathfrak{o}^{S_0}$ whose poles are restricted to S_0. We now consider specially constructed function fields with many places of degree one. The first two examples have been communicated to us by R. Auer and G. Pirsic and feature the maximally possible number $N_1(F/k)$, the third example is taken from table [9]. We do not observe a particularly abnormal behaviour of Algorithm 55.

$q = 4, \quad g = 13, \quad N_1(F/k) = 33$	9
$h(F/\mathbb{F}_4) = 96486886125, \quad \mathcal{C}l^0(F/\mathbb{F}_4) \cong 3 \times 3 \times 21 \times 21 \times 21 \times 105 \times 105 \times 105.$	$T = 6\,\text{min}$
$R(S_0) = 1, \quad h(S_0) = 96486886125,$ $\mathcal{C}l(\mathfrak{o}^S) \cong 3 \times 3 \times 21 \times 21 \times 21 \times 105 \times 105 \times 105.$	
$F := \mathbb{F}_4(x, \rho): \quad \rho^8 + (x^8 + x^6 + x^5 + x^4 + x^3 + x^2 + 1)\rho^4 + (x^{10} + x^9 + x^3 + x^2)\rho^2 + (x^{10} + x^9 + x^8 + x^6 + x^5 + x^4)\rho + x^{22} + wx^{20} + wx^{18} + wx^{16} + x^{15} + w^2x^{12} + x^{11} + wx^{10} + x^9 + wx^8 + x^6 + x^5 + x^4 = 0,$ $w^2 + w + 1 = 0.$	
$A = (x)_\infty \sim (8, 1).$ $m_g = 5, \quad m_h = 4, \quad m_S = 1 \;\rightarrow\; m_B := 5, \quad \lvert S\rvert = 345, \quad RV =_? 1286.$ $RR = 397, \quad RV = 1051, \quad R = 644 \;\rightarrow\; R/RV = 0.61, \quad \lvert S\rvert/R = 0.54.$ $W = 1.$	

$q=2, \quad g=9, \quad N_1(F/k)=12$	10
$h(F/\mathbb{F}_2)=135200, \quad \mathcal{C}l^0(F/\mathbb{F}_2)\cong 260\times 520.$	$T=5\,\mathrm{min}$
$R(S_0)=135200, \quad h(S_0)=1, \quad \mathcal{C}l(\mathfrak{o}^S)\cong 1.$	
$F:=\mathbb{F}_2(x,\rho):$ $\rho^{12}+(x^2+x+1)\rho^{10}+(x^2+x+1)\rho^9+(x^4+x^2+1)\rho^8+(x^6+x^5+x^4+x^3+x^2+x)\rho^6+(x^6+x^5+x^3+x+1)\rho^5+(x^6+x^5+x^3+x^2)\rho^4+(x^6+x^5+x^3+x+1)\rho^3+(x^4+x^2+1)\rho^2+(x^4+x^2+1)\rho+x^2+x+1=0.$	
$A=(x)_\infty\sim(1,1;1,1;1,1;1,1;1,1;1,1;1,1;1,1;1,1;1,1;1,1;1,1;1,1;1,1;1,1;1,1).$	
$m_g=8, \quad m_h=8, \quad m_S=2 \;\rightarrow\; m_B:=8, \quad \lvert S\rvert=83, \quad RV=_?\,112.$	
$RR=249, \quad RV=252, \quad R=196 \;\rightarrow\; R/RV=0.78, \quad \lvert S\rvert/R=0.42.$	
$W=3.$	

$q=9, \quad g=7, \quad N_1(F/k)=36$	11
$h(F/\mathbb{F}_9)=86704128, \quad \mathcal{C}l^0(F/\mathbb{F}_9)\cong 16\times 48\times 336\times 336.$	$T=88\,\mathrm{s}$
$R(S_0)=48, \quad h(S_0)=1806336, \quad \mathcal{C}l(\mathfrak{o}^S)\cong 16\times 336\times 336.$	
$F:=\mathbb{F}_9(x,\rho):$ $\rho^4+(-x^5+x^4-x^3+x^2)\rho^2+x^8-x^6+x^5+x^4+x^3+x^2=0.$	
$A=(x)_\infty\sim(2,1;2,1).$	
$m_g=3, \quad m_h=2, \quad m_S=2 \;\rightarrow\; m_B:=3, \quad \lvert S\rvert=256, \quad RV=_?\,834.$	
$RR=314, \quad RV=792, \quad R=470 \;\rightarrow\; R/RV=0.59, \quad \lvert S\rvert/R=0.54.$	
$W=1.$	

11. Appendix

This section contains some technical and supplementary results which have been used in the previous sections.

Theorem 56. *Any global function field F/k can be represented as the field of fractions of an absolutely irreducible plane affine curve $f(x,y)=0$ over k of degree $O(g)$.*

Proof. The main point of the theorem is the degree statement. We may assume $g\geq 2$ since the cases $g=0$ and $g=1$ correspond to the affine line and elliptic curves respectively. Let P be a place of F/k and $a\geq 0$ an integer such that $2g\leq a\deg(P)=O(g)$, and let $b=a+1$. By the theorem of Riemann-Roch there exist elements $x,y\in F$ such that for the (surjective) valuations $v_P(x)=-a$ and $v_P(y)=-b$, and x,y are integral at all other places. Also, let $z\in F^\times$ be a generator of the residue class field of P over its prime field.

Let Q_a,Q_b be two further places and $m_a,m_b\geq 0$ integers such that P, Q_a and Q_b are pairwise distinct and m_aQ_a-P, m_bQ_b-P are non-special and of degree $O(g)$. The strong approximation theorem and its proof [23, p. 31] imply that any residue class $\bmod P$ can be represented by an element in $\mathcal{L}(m_aQ_a)$ (or $\mathcal{L}(m_bQ_b)$), and is hence of degree $O(g)$ as well. We choose elements $z_a\in\mathcal{L}(m_aQ_a)$ and $z_b\in\mathcal{L}(m_bQ_b)$ satisfying $z_a\equiv zy^a/x^b \bmod P$ and $z_b\equiv zy^a/x^b \bmod P$.

We claim $F=k(xz_a,yz_b)$. To prove this consider the place $P'=P\cap k(xz_a,yz_b)$. Since xz_a and yz_b have only P as a common pole it follows that P is the only place of F/k lying above P'. Also, $v_P(z_a)=v_P(z_b)=0$, so $v_P(xz_a)=-a$, $v_P(yz_b)=-b$, and a,b generate the value group $\mathbb{Z}$ of the valuation at P. It follows that P is unramified over P'. It is thus inert over P' and F can be generated over $k(xz_a,yz_b)$ by any element of F which generates the residue class field of P over that of P'. But $z\equiv(xz_a)^b/(yz_b)^a \bmod P$ is a generator, which proves the claim.

For the field indices we obtain $[F:k(xz_a)]=O(g)$ and $[F:k(yz_b)]=O(g)$ because the degrees of the pole divisors of x,y and z_a,z_b are $O(g)$. The algebraic relation $f(xz_a,yz_b)=0$ between xz_a and yz_b then yields the desired affine curve of degree $O(g)$.

One can also generate F by $x,y+\lambda z$ or $x+\lambda z,y$ with $\lambda\in k$, if there are enough elements in k. Also, at least one of a and b is coprime to the characteristic, so at

least one of x and y will be a separating element and $f(x,y)$ is separable in at least one variable. □

We remark that such a curve can in fact be computed from quite arbitrary representations of F/k, only that the running time may not be polynomial in g since the representation may involve singularities of arbitrary degree. The resulting affine curve $f(x,y)=0$ will generally not be non-singular as well.

Lemma 57. *For $l \to \infty$ let $s = o(l^{1/2})$, $s \geq 1$ and $Q(l) = [0,l]^s \cap \mathbb{Z}^s$. For every lattice Λ of rank s in $\mathbb{Z}^s$ with $d(\Lambda) := (\mathbb{Z}^s : \Lambda) \leq l^{1/2}$ and $v \in \mathbb{Z}^s$ we have*

$$|(v+\Lambda) \cap Q(l)| \;=\; \frac{1+o(1)}{d(\Lambda)} \cdot |Q(l)|. \tag{58}$$

For every proper sublattice U of such a lattice Λ we have

$$|(v+U) \cap Q(l)| \;\leq\; \frac{1+o(1)}{2} \cdot |\Lambda \cap Q(l)|. \tag{59}$$

If additionally U has the rank s and possesses a basis in $Q(l)$ then we have

$$(\Lambda : U) \;\leq\; (s^{1/2}l)^s/d(\Lambda) \;\leq\; (s^{1/2}l)^s. \tag{60}$$

Proof. Clearly $|(v+\Lambda)\cap Q(l)| = |\Lambda \cap (Q(l)-v)|$ so we prove (58) for the latter intersection. Consider a basis of Λ in Hermite normal form and denote the echelon entries by $d_i \in \mathbb{Z}^{\geq 1}$. Then $\big|d_i\mathbb{Z} \cap [z, l+z]\big| = \lfloor l/d_i \rfloor + \delta$, where $z \in \mathbb{Z}$ and $\delta \in \{0,1\}$ depending on z, and taking the echelonised shape of the basis into account yields $|\Lambda \cap (Q(l)-v)| = \prod_{i=1}^{s} \big(\lfloor l/d_i \rfloor + \delta_i\big)$ for some $\delta_i \in \{0,1\}$. From this equation we get observing $d(\Lambda) = \prod_{i=1}^{s} d_i$ and $d_i \leq d(\Lambda) \leq l^{1/2}$:

$$(l-l^{1/2})^s/\,d(\Lambda) \quad\leq\quad |\Lambda \cap (Q(l)-v)| \quad\leq\quad (l+l^{1/2})^s/\,d(\Lambda). \tag{61}$$

Now we have $|Q(l)| = (1+o(1))l^s$ and $(l \pm l^{1/2})^s/l^s = (1 \pm l^{-1/2})^s = 1+o(1)$ because of $s = o(l^{1/2})$ and since $(1+z/x)^x \to e^z$ for $x \to \infty$. From (61) we thus obtain (58).

If r is the rank of U there are $\nu_j, w_{\nu_j} \in \mathbb{Z}^{\geq 1}$ such that the echelon elements of U are given by $w_{\nu_j} d_{\nu_j}$. For $r < s$ we obtain that

$$\begin{aligned} |U \cap (Q(l)-v)| \;&\leq\; \prod_{j=1}^{r} \big(\lfloor l/(w_{\nu_j} d_{\nu_j}) \rfloor + 1\big) \\ &\leq\; (l+1)^{s-1} \\ &\leq\; (1+o(1)) \,/\, \big(l^{1/2} d(\Lambda)\big) \cdot |Q(l)|. \end{aligned} \tag{62}$$

Since there is at least one $w_{\nu_j} \geq 2$ we obtain for $r = s$ that

$$\begin{aligned} |U \cap (Q(l)-v)| \;&\leq\; \prod_{j=1}^{r} \big(\lfloor l/(w_{\nu_j} d_{\nu_j}) \rfloor + 1\big) \\ &\leq\; d(\Lambda)^{-1} \prod_{j=1}^{s} \big(l/w_{\nu_j} + d_{\nu_j}\big) \\ &\leq\; \big(l + 2l^{1/2}\big)^s \,/\, \big(2d(\Lambda)\big) \\ &\leq\; (1+o(1)) \,/\, \big(2d(\Lambda)\big) \cdot |Q(l)|. \end{aligned} \tag{63}$$

The bound (59) now follows from (58) for $v=0$, (62) and (63).

The last statement (60) about the index of U is the estimation of the determinant of a matrix generated by s independent vectors of U in $Q(l)$ by means of the Hadamard bound. □

The next lemma is a generalisation of [19, Lemma 4.1].

Lemma 64. *Assume $n \geq 1$. Let Q be a finite subset of $\mathbb{Z}^s$ containing vectors of norm $\leq n$ and let $b_1, \ldots, b_s$ be a basis of $\mathbb{Z}^s$. If $t = 4s\log_2(n+1)$ and $r = 2\log_2(s\log_2(n+1))$ then after choosing elements $v_1, \ldots, v_{tr}$, $w_1, \ldots, w_{2sr}$ from Q according to some random (not necessarily uniform) distribution we can generate $\mathbb{Z}^s$ by the elements $v_1, \ldots, v_{tr}$ and $b_j + w_{(j-1)2r+i}$ for $j = 1, \ldots, s$ and $i = 1, \ldots, 2r$ with probability tending to 1 as s tends to infinity.*

Proof. (i) Let $R = \mathbb{Q}$, $a = s$, $V_0 = \{0\}$ and $r \geq 1$ arbitrary. In the following we consider ar elements v_i and sr elements w_j chosen from Q according to the random distribution. For $\nu = 1, \ldots, a$ let V_ν denote the R-submodule of R^s spanned by $v_1, \ldots, v_{\nu r}$.

Let p_ν denote the probability, that an (according to the distribution) random $v \in Q$ lies in V_ν. We claim that $p_a \geq 1/2$ with probability $\Pr(p_a \geq 1/2) > (1-2^{-r})^a$. To prove this let X_ν denote the event that $p_\nu \geq 1/2$ or $V_\nu \subsetneq V_{\nu+1}$. Using the abbreviation $\Pr_\nu(x) = \Pr(x \,|\, X_0 \wedge \cdots \wedge X_{\nu-1})$ for conditional probability we obtain $\Pr_\nu(V_\nu = V_{\nu+1} \,|\, p_\nu < 1/2) < 2^{-r}$ by the definition of V_ν and p_ν, and then

$$\begin{aligned}
\Pr_\nu(X_\nu) &= \Pr_\nu(p_\nu \geq 1/2) + \Pr_\nu(V_\nu \subsetneq V_{\nu+1} \wedge p_\nu < 1/2) \\
&= \Pr_\nu(p_\nu \geq 1/2) + \Pr_\nu(V_\nu \subsetneq V_{\nu+1} \,|\, p_\nu < 1/2)\Pr_\nu(p_\nu < 1/2) \\
&\geq \Pr_\nu(V_\nu \subsetneq V_{\nu+1} \,|\, p_\nu < 1/2)\,(\Pr_\nu(p_\nu \geq 1/2) + \Pr_\nu(p_\nu < 1/2)) \\
&= \Pr_\nu(V_\nu \subsetneq V_{\nu+1} \,|\, p_\nu < 1/2) \\
&> 1 - 2^{-r}.
\end{aligned}$$

The event $p_a \geq 1/2$ contains the event $X_0 \wedge \cdots \wedge X_{a-1}$. Indeed, if $p_\nu \geq 1/2$ for some ν then $p_a \geq 1/2$ since $p_\mu \leq p_{\mu+1}$. Otherwise $p_\nu < 1/2$ for all ν and $X_0 \wedge \cdots \wedge X_{a-1}$ implies $V_0 \subsetneq \cdots \subsetneq V_a$. Because of $R = \mathbb{Q}$ this in turn implies $V_a = R^s$ and hence $p_a \geq 1/2$. Using also the definition of $\Pr_\nu$ we thus obtain

$$\begin{aligned}
\Pr(p_a \geq 1/2) \geq \Pr(X_0 \wedge \cdots \wedge X_{a-1}) &= \prod_{\nu=0}^{a-1} \Pr_\nu(X_\nu) \\
&> (1-2^{-r})^a.
\end{aligned}$$

Let W_j be the R-submodule generated by V_a and the elements $b_j + w_{(j-1)r+i}$ for $j = 1, \ldots, s$ and $i = 1, \ldots, r$. If $p_a \geq 1/2$ then with probability at least $1 - 2^{-r}$ we have at least one i such that $w_{(j-1)r+i} \in V_a$, hence $b_j \in W_j$. Thus, if $p_a \geq 1/2$ then $b_j \in W_j$ for all j with probability at least $(1-2^{-r})^s$.

Finally, with probability at least $(1-2^{-r})^{a+s}$ we have both $p_a \geq 1/2$ and all $b_j \in W_j$. In this case, $W_a = R^s$.

(ii) From (i) we obtain a generating system of $\mathbb{Q}^s$ by randomly choosing sr elements v_i and sr elements w_j from Q, with probability at least $(1-2^{-r})^{2s}$. Over $\mathbb{Z}$ this system will generate a $\mathbb{Z}$-submodule of $\mathbb{Z}^s$ of index less than or equal to $(n+1)^s$, according to the vector sizes and the Hadamard bound. We can now repeat the reasoning in (i) verbatim where we let V_0 be this submodule and $R = \mathbb{Z}$. The ascending chain of submodules $V_0 \subsetneq \cdots \subsetneq V_\nu$ must stop after $a \leq \log_2((n+1)^s)$ steps, as the index in $\mathbb{Z}^s$ is divided by at least 2 every step. We arrive at $R^s = \mathbb{Z}^s$ with probability at least $(1-2^{-r})^{s+s\log_2(n+1)}$. Step (ii) requires another $(s + s\log_2(n+1))r$ elements v_i and sr elements w_j from Q.

Combining (i) and (ii) we get a probability of at least $(1-2^{-r})^{3s+s\log_2(n+1)} \geq (1-2^{-r})^{4s\log_2(n+1)}$ to generate $\mathbb{Z}^s$ with all $(3s+s\log_2(n+1))r \leq tr$ elements v_i and $2rs$ elements w_j randomly chosen from Q. Because of $(1-2^{-r})^{4s\log_2(n+1)} \geq 1 - (4s\log_2(n+1))\cdot 2^{-r}$ we see that this probability tends to 1 for $r = 2\log_2(s\log_2(n+1))$ and $s \to \infty$. □

Proposition 65. *Let $p_0, p_1 \in \mathbb{R}^{>0}$ with $p_0 + p_1 = 1$ and let V denote an algorithm returning the result "true" with probability p_1.*

(i) The probability to get the result "true" at least once in $\lfloor p_1^{-1} \rfloor$ executions of V tends to $1 - 1/e$ with $p_1 \to 0$.

(ii) For $k_0 \in \mathbb{Z}^{\geq 2}$ with $p_0^{k_0-1} e k_0 < 1$ the probability to get the result "true" at least r times after $k_0 r$ executions of V tends to 1 with $r \to \infty$.

Proof. For part (i) we take into account that the probability to not get the value "true" is $(1-p_1)^{\lfloor p_1^{-1} \rfloor}$. For $p_1 \to 0$ this tends to $1/e$.

For the proof of part (ii) we put $w := k_0 r$ and let X_r denote the minimal number of executions of V providing exactly r times "true". We consider the probability $\Pr(X_r \leq w)$ to get the result "true" at least r times in w executions of V:

$$\Pr(X_r \leq w) \;=\; \sum_{j=r}^{w} \binom{w}{j} p_1^j \, p_0^{w-j}.$$

Using the Stirling formula in the form $\log(r!) = r(\log(r) - 1 + o(1))$ we get for the complementary probability

$$\begin{aligned} \Pr(X_r > w) &= \sum_{j=0}^{r-1} \binom{k_0 r}{j} (1-p_0)^j \, p_0^{k_0 r - j} \\ &\leq r \, p_0^{(k_0-1)r} \binom{k_0 r}{r} \leq r \, p_0^{(k_0-1)r} \, (k_0 r)^r / r! \\ &\leq \left(p_0^{k_0-1} e k_0\right)^r e^{o(r)} \end{aligned}$$

which tends to zero for $r \to \infty$ due to the choice of k_0. □

Corollary 66. *For $r \to \infty$ let $p_1 \to 0$ from above. Let V be an algorithm returning the result "true" with a probability of p_1. After $4\lfloor p_1^{-1} \rfloor r$ executions of V we get the result "true" at least r times with probability tending to 1.*

Proof. Consider the algorithm $V' :=$ "$\lfloor p_1^{-1} \rfloor$ executions of V". The probability p_1' to get "true" at least once on execution of V' tends to $1 - 1/e$ according to Proposition 65, (i). Using (ii) we see that $4r$ executions of V' yield "true" at least r times with probability tending to 1. Here $k_0 = 4$ is possible because with $p_0' = 1 - p_1'$ we have $p_0' \to 1/e$ and $e^{-k_0} k_0 < e^{-2}$. □

The following theorem is a function field version of the theorem of Brauer-Siegel with explicit error term. Note that the error term is much smaller than the upper bound $\log(h/q^g) \leq O((n-1)\log(g\log(q)))$ in [14, Theorem 3], as g tends to infinity.

Theorem 67. *Let F/k run through a sequence of global function fields with $g \to \infty$ and assume that F/k has a rational subfield $k(x)$ of index $n = n(g) := [F : k(x)]$. The class number of F/k then satisfies*

$$\left|\log\left(h(F/k) \,/\, q^g\right)\right| \;\leq\; \min\left\{O(g/q^{1/2}), (n-1)\left(\log(\lceil 2\log_q(g)\rceil) + O(1)\right)\right\}.$$

Proof. The number $N_r(F/k)$ of places of degree one of the constant field extension F_r/k_r of F/k of degree r can be bounded from above by $N_r(F/k) \leq n(q^r+1)$, because there are at most n places of F_r/k_r over each place of $k_r(x)/k_r$. Using this we obtain from Theorem 36 that

$$\begin{aligned} \left|\log\left(h(F/k) \,/\, q^g\right)\right| &\leq \left|\sum_{r=1}^{d} \frac{q^{-r}}{r}\left(N_r(F/k) - q^r - 1\right)\right| \;+\; \frac{2g}{q^{1/2}-1} \cdot \frac{q^{-d/2}}{d+1} \\ &\leq (n-1)\sum_{r=1}^{d} q^{-r}(q^r+1)/r \;+\; \frac{2g}{q^{1/2}-1} \cdot \frac{q^{-d/2}}{d+1} \qquad (68) \end{aligned}$$

for all $d \geq 0$. The first part of the bound of the theorem follows from taking $d = 0$. Let now $d = \lceil 2\log_q(g) \rceil$. Using $\sum_{r=1}^{d} 1/r \leq \log(d) + 1$ we obtain

$$\begin{aligned}\sum_{r=1}^{d} q^{-r}(q^r+1)/r &\leq \sum_{r=1}^{d} 1/r + \sum_{r=1}^{\infty} q^{-r} \\ &\leq \log(d) + 1 + \frac{1}{q(1-1/q)}.\end{aligned}$$

Putting this together yields the second part of the bound of the theorem. □

More generally one can show that $h(F/k)/q^g$ decreases at most polynomially in g, and that $h(F/k)/q^g$ increases exponentially in g if for example the number of places of degree one becomes asymptotically large. We refer to [14, 20].

12. Acknowledgements

I thank S. Contini and P. Gaudry for helpful discussions.

References

1. L. Adleman, J. DeMarrais, and M.-D. Huang. A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields. In L. Adleman et al., editors, *Proceedings of the First Symposium on Algorithmic Number Theory, ANTS-I*, LNCS 877, pages 28–40, Ithaca, New York, 1994. Springer-Verlag, Berlin-Heidelberg-New York.
2. M. Bauer. A subexponential algorithm for solving the discrete logarithm problem in the Jacobian of high genus hyperelliptic curves over arbitrary finite fields. Preprint, 2001.
3. W. Bosma, J. Cannon, and C. Playoust. The Magma algebra system I: The user language. *J. Symbolic Comp.*, 24, 3/4:235–265, 1997.
4. Comp. algebra group. Magma. `http://www.maths.usyd.edu.au:8000/u/magma/`, 2004.
5. J.-M. Couveignes. Algebraic groups and discrete logarithms. In *Public key cryptography and computational number theory*, pages 17–27, Warsaw (2000), 2001. de Gruyter, Berlin.
6. A. Enge. Computing discrete logarithms in high-genus hyperelliptic Jacobians in provably subexponential time. *Math. Comp.*, 71:729–742, 2002.
7. A. Enge and P. Gaudry. A general framework for subexponential discrete logarithm algorithms. *Acta Arith.*, 102(1):83–103, 2002.
8. A. Enge and A. Stein. Smooth ideals in hyperelliptic function fields. *Math. Comp.*, 71:1219–1230, 2002.
9. G. van der Geer and M. van der Vlugt. Tables of curves with many points. Regularly updated table under `http://www.wins.uva.nl/~geer`, 2004.
10. D. Gieseker. Stable curves and special divisors: Petri's conjecture. *Invent. Math.*, 66(2):251–275, 1982.
11. F. Hess. *Zur Divisorenklassengruppenberechnung in globalen Funktionenkörpern*. PhD Thesis, Technische Universität Berlin, 1999.
12. F. Hess. Computing Riemann-Roch spaces in algebraic function fields and related topics. *J. Symbolic Comp.*, 33(4):425–445, 2002.
13. Kant group. Kash. `http://www.math.tu-berlin.de/~kant`, 2004.
14. G. Lachaud and M. Martin-Deschamps. Nombre de points des jacobiennes sur un corps fini. *Acta Arith.*, 56(4):329–340, 1990.
15. A. Lauder and D. Wan. Counting points on varieties over finite fields of small characteristic. To appear in the MSRI proceedings of the workshop on algorithmic number theory Aug-Dec 2000, 2000.
16. H. W. Lenstra Jr. and C. Pomerance. A rigorous time bound for factoring integers. *J. Amer. Math. Soc.*, 5(3):483–516, 1992.
17. V. Müller, A. Stein, and C. Thiel. Computing discrete logarithms in real quadratic congruence function fields of large genus. *Math. Comp.*, 68:807–822, 1999.
18. C. Pomerance. Analysis and comparison of some integer factoring algorithms. In R. Tijdeman & H. Lenstra, editor, *Computational Methods in Number Theory*, pages 89–139. Mathematisch Centrum, Tract 154, Amsterdam, 1982.
19. C. Pomerance. Fast, rigorous factorization and discrete logarithm algorithms. In *Discrete algorithms and complexity (Kyoto, 1986)*, volume 15 of *Perspect. Comput.*, pages 119–143. Academic Press, Boston, MA, 1987.

20. H.-G. Quebbemann. Estimates of regulators and class numbers in function fields. *J. Reine angew. Math.*, 419:79–87, 1991.
21. M. Seysen. A probabilistic factorization algorithm with quadratic forms of negative discriminant. *Math. Comp.*, 48:757–780, 1987.
22. A. Stein. *Algorithmen in reell-quadratischen Kongruenzfunktionenkörpern*. PhD Thesis, Universität des Saarlandes, 1996.
23. H. Stichtenoth. *Algebraic Function Fields and Codes*. Springer-Verlag, Berlin-Heidelberg-New York, 1993.
24. F. Vercauteren. Private communication. 2003.

Computer Science Department, Woodland Road, University of Bristol, BS8 1UB, UK
E-mail address: florian@cs.bris.ac.uk