COUNTING CURVES AND THEIR PROJECTIONS

Joachim von zur Gathen
Department of Computer Science, University of Toronto
Toronto, Ontario M5S 1A4, Canada
gathen@cs.toronto.edu

Marek Karpinski
Department of Computer Science, University of Bonn
5300 Bonn 1, Germany
marek@theory.cs.bonn.edu

Igor Shparlinski
School of MPCE, Macquarie University
Sydney, NSW 2109, Australia
igor@macadam.mpce.mq.edu.au

June 5, 1994

Abstract. Some deterministic and probabilistic methods are presented for counting and estimating the number of points on curves over finite fields, and on their projections. The classical question of estimating the size of the image of a univariate polynomial is a special case. For curves given by sparse polynomials, the counting problem is $\#\mathcal{P}$-complete via probabilistic parsimonious Turing reductions.

1. Introduction

One of the most celebrated results in algebraic geometry is Weil's theorem on the number of points on algebraic curves over a finite field. In this paper, we address some computational problems related to this question.

Our main results are:

- A "computational Weil estimate" for projections of curves and images of polynomials, in Section 3.
- $\#\mathcal{P}$-completeness of the exact counting problem for sparse curves, in Section 4.

We consider a finite field $\mathbb{F}_q$ with q elements, an algebraic closure K of $\mathbb{F}_q$, a polynomial $f \in \mathbb{F}_q[x, y]$, the plane curve $\mathcal{C} = \{f = 0\} = \{(a, b) \in K^2 : f(a, b) = 0\}$ of degree d and genus $g \leq (d-1)(d-2)/2$ defined over $\mathbb{F}_q$, and the number $N_n(\mathcal{C}) = \#(\mathcal{C} \cap \mathbb{F}_{q^n}^2)$ of rational points on $\mathcal{C}$ over $\mathbb{F}_{q^n}$.

The only general information about the numbers $N_n(\mathcal{C})$ is Weil's fundamental result that for an absolutely irreducible smooth projective curve $\mathcal{C}$ over $\mathbb{F}_q$ there are algebraic integers $\vartheta_1, \ldots, \vartheta_g \in \mathbb{C}$—the *Frobenius roots*—with absolute values $|\vartheta_i| = q^{1/2}$ for $1 \leq i \leq g$ such that

$$N_n(\mathcal{C}) = q^n + 1 - \sum_{1 \leq i \leq g} (\vartheta_i^n + \bar{\vartheta}_i^n); \tag{1.1}$$

in particular, this implies the *Weil estimate*

$$|N_n(\mathcal{C}) - q^n - 1| \leq 2gq^{n/2}. \tag{1.2}$$

Recall that $f \in \mathbb{F}_q[x, y]$ is *absolutely irreducible* if it is irreducible in $K[x, y]$. $\mathcal{C} = \{f = 0\}$ is *smooth* if there are no solutions to $f = \partial f/\partial x = \partial f/\partial y = 0$, and a *projective* curve includes points at infinity. Any text on algebraic geometry, such as Shafarevich (1974) or Hartshorne (1977), explains these notions, and

Appendix C in the latter text gives a highly readable exposition of Weil's results and their far-reaching generalizations.

Some small improvements on the Weil estimate—exploiting the algebraic nature of the Frobenius roots—are in Serre (1983a, 1983b), and Shparlinski (1992a), Chapter 5. Vladuts & Drinfeld (1983) show $N_1(\mathcal{C}) \leq g(q^{1/2}-1)+o(g)$ for curves of large genus over a fixed field; this is twice better than the Weil estimate. Fried & Jarden (1986, Theorem 4.9) and Bach (1993) show that some variants of (1.2) hold for absolutely irreducible projective curves even if they are singular. Bach shows that (1.2) holds for the arithmetic genus g, which also satisfies $g \leq (d-1)(d-2)/2$.

The first question addressed in this paper is the *curve size problem*: Can we find an approximation to $N_n(\mathcal{C})$ that is better than the Weil estimate? The ultimate goal would be a deterministic algorithm that calculates $N_n(\mathcal{C})$ exactly and runs in polynomial time $(dn \log q)^{O(1)}$; it is not clear at all whether this goal can be attained. "Time" will usually mean arithmetic operations in $\mathbb{F}_q$; the number of bit operations is at most a factor $O\tilde{}(\log q)$ larger, where the "soft-O" notation $s = O\tilde{}(t)$ means that $s = t \cdot (\log t + 2)^{O(1)}$.

Pila (1990) presented an algorithm for calculating $N_n(\mathcal{C})$ that generalized Schoof's (1985) method, which applies to elliptic curves. The computing time of Pila's algorithm is $(n \log q)^{\Delta(d)}$, where $\Delta(d)$ is a doubly exponential function of d. Since the first version of the present paper appeared, Huang & Ierardi (1993) have given a probabilistic algorithm with running time as Pila's, but $\Delta(d) = d^{O(1)}$.

For the related problem of counting the number of zeros of a multivariate sparse polynomial over a "small" finite field some probabilistic approximation algorithms are known; see Grigoryev & Karpinski (1991) and the references there.

Section 2 introduces a "strip counting" method. It is based on the principle that the behaviour of a curve over a wide enough "strip" is the same as over the whole field, and uses Bombieri's (1966) bound on exponential sums along a curve. We use it to count the number of absolutely irreducible components defined over $\mathbb{F}_q$, and to get a size estimate from Weil's estimate.

The second question of this paper is the *projection size problem*: determine the number of points in the image of the projection $\pi : \mathcal{C} \to \mathbb{F}_q$, and in fact the number r_i of points in $\mathbb{F}_q$ with exactly i preimages under π. Section 3 presents a "computational Weil estimate" of the form

$$|r_i - \lambda_i q| = O(q^{1/2}), \tag{1.3}$$

with $\lambda_i \in \mathbb{Q}$, and a "strip counting" method for computing λ_i. This is "computational" in the sense that the classical Weil estimate (1.2)—for a different problem—sets $\lambda_i = 1$, without any computation. Both the computing time and the constant implied in (1.3) depend exponentially on d, and q has to be prime for "strip counting".

We apply this result to the important problem of counting the number of points with a fixed number of preimages under a univariate polynomial or rational function. Apparently no "Weil estimate" has been previously known for this problem, however, a special case of our resulting formula, namely for the total image size of a polynomial, is essentially in Birch & Swinnerton-Dyer (1959).

In Section 4, we show that curve counting is $\#\mathcal{P}$-complete via probabilistic Turing reductions if the defining polynomial $f \in \mathbb{F}_q[x,y]$ is given in sparse representation. This is based on efficient methods, due to McCurley and Alford, Granville & Pomerance, for finding primes in certain arithmetic progressions. The basic tool is a reduction from certain gcd problems for sparse polynomials over $\mathbb{Z}$ for which Plaisted (1977) proved $\mathcal{NP}$-hardness and Quick (1986) $\#\mathcal{P}$-hardness. In those papers, membership in $\mathcal{NP}$ or $\#\mathcal{P}$ was left as an open question. This is answered affirmatively for the curve counting problem, and as a consequence we also solve this open question for the special variant of the gcd problem that we consider.

In Section 5, we reduce the general curve counting problem to that of absolutely irreducible curves—that is the case to which the Weil estimate applies. In Section 6, we show how to compute $N_n(\mathcal{C})$ quickly in the case of "small" d and q and "large" n; in all other sections we restrict to the case $n = 1$. Section 8 gives an (ε, δ)-approximation scheme. This provides estimates also for "large" d, where the Weil estimate gives no information. In Section 10, we introduce a deterministic method to estimate the image size of special polynomial mappings $\mathbb{F}_{q^n}^m \to \mathbb{F}_q$; this method is particularly useful when n is large.

Throughout the paper, we use $M(n)$ to denote an upper bound on the cost of multiplication, so that polynomials in $R[x]$ of degree at most n can be multiplied with $O(M(n))$ operations in R, for any ring R. We may use $M(n) = n^2$ for "classical arithmetic", and $M(n) = n \log n \operatorname{loglog} n$ for "fast arithmetic". If $R \subseteq R[y]/(f) = S$ is an extension of degree m, then one can multiply in $S[x]$ with $O(M(mn))$ operations in R (see von zur Gathen & Shoup 1992, Lemma 2.2). All logarithms in this paper are natural.

2. Estimating the size of a curve over a prime field

We propose a deterministic "strip counting" algorithm to estimate the size of curves over a prime field $\mathbb{F}_q$. It relies on the general principle that the "behaviour" of a curve or an algebraic variety over a wide enough "strip" is the same as over the whole field; Shparlinski (1992b) gives another example of using this principle.

We will use the following notation: K is an algebraic closure of $\mathbb{F}_q$, $\mathcal{X} \subseteq K^{m+1}$ is a curve of degree d, $\mathcal{C} = \mathcal{X} \cap \mathbb{F}_q^{m+1}$ are the rational points of $\mathcal{X}$ over $\mathbb{F}_q$. Often we concentrate on plane curves $\mathcal{X} = \{(a,b) \in K^2 : f(a,b) = 0\}$ given by $f \in \mathbb{F}_q[x,y]$ of total degree d.

Since we are interested in counting the size of curves, we assume that they are *reduced*, i.e., without multiple components; for a plane curve $\mathcal{C} = \{f = 0\}$, this means that f is squarefree.

For an absolutely irreducible affine curve $\mathcal{C} \subseteq \mathbb{F}_q^m$, (1.2) implies that

$$|\#\mathcal{C} - q| \le d^2 q^{1/2}, \tag{2.1}$$

since

$$|\#\mathcal{C} - q| \le 2\frac{(d-1)(d-2)}{2} q^{1/2} + d + 1 \le d^2 q^{1/2}.$$

The last inequality holds for $d \ge 2$, but (2.1) is obviously true also for $d = 1$.

For a set $S \subseteq \mathbb{F}_q^m$ and $A \subseteq \mathbb{F}_q$, we write

$$S(A) = S \cap (A \times \mathbb{F}_q^{m-1}) \tag{2.2}$$

for the set of points in S over A. The crucial ingredient for our "strip-counting" is the following consequence of Bombieri's (1966) bound on exponential sums along a curve.

LEMMA 2.1. *Let p be a prime, and $\mathcal{C} \subseteq \mathbb{F}_p^m$ a curve of degree $d < p$ over $\mathbb{F}_p$ none of whose absolutely irreducible components defined over $\mathbb{F}_p$ is contained in a hyperplane $\{a\} \times \mathbb{F}_p^{m-1}$ with $a \in \mathbb{F}_p$. Furthermore, let $0 < h \le p$ and $A = \{0, \ldots, h-1\} \subseteq \mathbb{F}_p$. Then the number $\#\mathcal{C}(A)$ of points on $\mathcal{C}$ over A satisfies*

$$|\#\mathcal{C}(A) - h/p \cdot \#\mathcal{C}| \le ((d^2 - d)p^{1/2} + d^2) \log p,$$

and if $d \le p^{1/2}$, then

$$|\#\mathcal{C}(A) - h/p \cdot \#\mathcal{C}| \le d^2 p^{1/2} \log p.$$

PROOF. We have

$$\#\mathcal{C}(A) = \frac{1}{p} \sum_{(a_1,\ldots,a_m)\in\mathcal{C}} \sum_{0\le u<p} \sum_{0\le v<h} \exp(2\pi i u(a_1 - v)/p).$$

Rearranging the sum and separating the term $h\#\mathcal{C}/p$ corresponding to $u = 0$, we get

$$\#\mathcal{C}(A) - h\#\mathcal{C}/p = \frac{1}{p} \sum_{1\le u<p} \sum_{(a_1,\ldots,a_m)\in\mathcal{C}} \exp(2\pi i u a_1/p) \sum_{0\le v<h} \exp(-2\pi i u v/p).$$

The bound of Bombieri (1966), Theorem 6, on exponential sums along a curve implies for $1 \le u < p$ that

$$\Big| \sum_{(a_1,\ldots,a_m)\in\mathcal{C}} \exp(2\pi i u a_1/p) \Big| \le (d^2 - d)p^{1/2} + d^2.$$

Using this bound and the following well-known inequality

$$\sum_{1\le u<p} \Big| \sum_{0\le v<h} \exp(2\pi i u v/p) \Big| < p \log p$$

(see e.g., Vinogradov 1981, Exercise 11.c for chapter 3, p. 52), we get

$$|\#\mathcal{C}(A) - h/p \cdot \#\mathcal{C}| < ((d^2 - d)p^{1/2} + d^2) \log p. \quad \square$$

We say that a curve in $\mathbb{F}_q^{m+1}$ is *without vertical components* (over $\mathbb{F}_q$) if and only if it has no absolutely irreducible components defined over $\mathbb{F}_q$ in a hyperplane $\{a\} \times \mathbb{F}_q^m$ for any $a \in \mathbb{F}_q$, and say *without vertical lines* if $m = 1$. This condition is necessary for the estimate of Lemma 2.1 to be valid. In general, it is not clear to us how to check for it. However, for a plane curve $\mathcal{C}$ given by $f \in \mathbb{F}_q[x, y]$, we can compute the content $\mathrm{cont}_x(f) \in \mathbb{F}_q[x]$, and then $\mathcal{C}' = \{f/c = 0\}$ has no vertical lines. The vertical lines in $\mathcal{C}$ are precisely the lines $\{a\} \times \mathbb{F}_q$, where $a \in \mathbb{F}_q$ is a root of c. Given the number of absolutely irreducible components or (an estimate for) the number of points on $\mathcal{C}'$, it is easy to compute the corresponding quantity for $\mathcal{C}$. In the sequel, we will often assume that $\mathcal{C}$ is without vertical lines.

LEMMA 2.2. *Let $\mathcal{C} \subseteq \mathbb{F}_q^{m+1}$ be a reduced curve of degree d without vertical components, $\mathcal{C}_1, \ldots, \mathcal{C}_\sigma$ the absolutely irreducible components of $\mathcal{C}$ that are defined over $\mathbb{F}_q$, and $A \subseteq \mathbb{F}_q$. Then*

$$|\#\mathcal{C}(A) - \sum_{1\le i\le\sigma} \#\mathcal{C}_i(A)| \le d^2/2.$$

PROOF. We decompose $\mathcal{C}$ into its irreducible components over $\mathbb{F}_q$:

$$\mathcal{C} = \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_\sigma \cup \mathcal{C}_{\sigma+1} \cup \cdots \cup \mathcal{C}_\tau,$$

where $\mathcal{C}_{\sigma+1}, \ldots, \mathcal{C}_\tau$ are not absolutely irreducible, and let $d_i = \deg \mathcal{C}_i$ for $i \le \tau$, so that $d = \sum_{1 \le i \le \tau} d_i$. Then, using the estimates from Lemma 5.2 below, we have

$$\begin{aligned} \sum_{1\le i\le\sigma} \#\mathcal{C}_i(A) - \#\mathcal{C}(A) &\le \sum_{1\le i<j\le\sigma} \#(\mathcal{C}_i(A) \cap \mathcal{C}_j(A)) \\ &\le \sum_{1\le i<j\le\sigma} d_i d_j \le d^2/2, \end{aligned}$$

$$\begin{aligned} \#\mathcal{C}(A) - \sum_{1\le i\le\sigma} \#\mathcal{C}_i(A) &\le \sum_{\sigma<i\le\tau} \#\mathcal{C}_i(A) \\ &\le \sum_{\sigma<i\le\tau} d_i^2/4 \le d^2/4. \qquad \square \end{aligned}$$

THEOREM 2.3. *Let p be a prime, $\mathcal{C} \subseteq \mathbb{F}_p^{m+1}$ a curve of degree $d \le p^{1/2}$ with exactly σ absolutely irreducible components that are defined over $\mathbb{F}_p$ and without vertical lines, $0 < h \le p$, and $A = \{0, \ldots, h-1\} \subseteq \mathbb{F}_p$. Then*

(i) $|\#\mathcal{C}(A) - h\#\mathcal{C}/p| \le d^2 + d^2 p^{1/2} \log p < 2.03\, d^2 p^{1/2} \log p$,

(ii) $|\#\mathcal{C}(A) - \sigma h| \le d^2(1/2 + p^{1/2} + p^{1/2} \log p) < 3\, d^2 p^{1/2} \log p$.

PROOF. (i) Let $\mathcal{C}_1, \ldots, \mathcal{C}_\sigma \subseteq \mathbb{F}_p^{m+1}$ be the absolutely irreducible components of $\mathcal{C}$ that are defined of $\mathbb{F}_p$, and $d_i = \deg \mathcal{C}_i$ for $i \le \sigma$, so that $\sum_{1\le i\le\sigma} d_i \le d$. Using Lemmas 2.1 and 2.2, and (2.1), we have

$$\begin{aligned} &|\#\mathcal{C}(A) - h\#\mathcal{C}/p| \\ &\le |\#\mathcal{C}(A) - \sum_{1\le i\le\sigma} \#\mathcal{C}_i(A)| + \sum_{1\le i\le\sigma} |\#\mathcal{C}_i(A) - h/p \cdot \#\mathcal{C}_i| \\ &\qquad + h/p \cdot |\sum_{1\le i\le\sigma} \#\mathcal{C}_i - \#\mathcal{C}| \\ &\le d^2/2 + \sum_{1\le i\le\sigma} d_i^2 p^{1/2} \log p + h/p \cdot d^2/2 \\ &\le d^2 + d^2 p^{1/2} \log p < 2.03\, d^2 p^{1/2} \log p, \end{aligned}$$

since $1 < 1.03\,p^{1/2}\log p$ for $p \geq 2$. For (ii), we have from Lemmas 2.2 and 2.1, applied to the absolutely irreducible curves $\mathcal{C}_1, \ldots, \mathcal{C}_\sigma$, that

$$\begin{aligned}
|\#\mathcal{C}(A) - \sigma h| &\leq |\#\mathcal{C}(A) - \sum_{1\leq i\leq\sigma} \#\mathcal{C}_i(A)| \\
&+ \sum_{1\leq i\leq\sigma} |\#\mathcal{C}_i(A) - h/p\cdot\#\mathcal{C}_i| + h/p\cdot\sum_{1\leq i\leq\sigma}|\#\mathcal{C}_i - p| \\
&\leq d^2/2 + p^{1/2}\log p \sum_{1\leq i\leq\sigma} d_i^2 + h/p\cdot d^2p^{1/2} \\
&\leq d^2(1/2 + p^{1/2} + p^{1/2}\log p) \;<\; 3\,d^2p^{1/2}\log p. \quad \square
\end{aligned}$$

We will repeatedly use a "brute force" method for computing $\#\mathcal{C}(A)$, where $\mathcal{C} = \{f = 0\} \subseteq \mathbb{F}_{q^n}^2$, namely, by calculating

$$\deg\gcd(f(a,y), y^{q^n} - y) = \#\mathcal{C}(\{a\}) \tag{2.3}$$

for all $a \in A$, and summing up. To estimate the time for this, we let $MM(d)$ denote the cost of $d \times d$-matrix multiplication, so that $MM(d) = O(d^{2.376})$ (Coppersmith & Winograd 1990).

Lemma 2.4. *Let $d, m, n \geq 1$, $f \in \mathbb{F}_q[x,y]$ have degree d, $\mathcal{C} = \{f = 0\} \subseteq \mathbb{F}_{q^n}^2$, and $A \subseteq \mathbb{F}_q^m \subseteq \mathbb{F}_q^n$. Then we can compute $\#\mathcal{C}(A)$ with*

$$O\Big(\#A\cdot M(m)(1+\log m)\cdot(M(d)\log(dq) + d^{1/2}\log n[M(d) + MM(d^{1/2})])\Big)$$

$$\text{or } O\tilde{\ }(\#A\cdot m\cdot(d\log q + d^{1.7}\log n))$$

operations in $\mathbb{F}_q$, and when $n = m = 1$, with only

$$O(\#A\cdot M(d)\log(dq))$$

operations.

Proof. For each $a \in A$, we can compute $y^{q^n} \bmod f(a,y)$ with

$$O(M(d)\log q + d^{1/2}\log n[M(d) + MM(d^{1/2})])$$

operations in $\mathbb{F}_q^m$ (von zur Gathen & Shoup 1992, Lemma 5.3), and the gcd in (2.3) with $O(M(d)\log d)$ operations. $\square$

Actually, this method even computes the set $\mathcal{C}(A)$.

In particular, the "brute force" method of calculating $N_n(\mathcal{C})$, with $A = \mathbb{F}_q^n$, can be executed with $O\tilde{\ }(dq^n)$ operations in $\mathbb{F}_q$.

COROLLARY 2.5. *Let p be a prime, $\mathcal{C} \subseteq \mathbb{F}_p^2$ have degree d, $\epsilon > 0$, and set $r = p^{1/2}/(2.03 \log p)$.*

(i) We can compute an approximation γ to $\#\mathcal{C}$ satisfying

$$|\#\mathcal{C} - \gamma| \leq \epsilon p$$

with $O\tilde{\ }(\epsilon^{-1} d^3 p^{1/2})$ operations in $\mathbb{F}_p$ if $d^2 \leq \epsilon r$, and $O\tilde{\ }(dp)$ operations otherwise.

(ii) If $d^2 \leq r/3$, then we can compute the number of absolutely irreducible components of $\mathcal{C}$ defined over $\mathbb{F}_p$ with $O\tilde{\ }(d^3 p^{1/2})$ operations in $\mathbb{F}_p$.

PROOF. We may assume that $\mathcal{C}$ has no vertical lines. For (i), we set $h = 2.03\, \epsilon^{-1} d^2 p^{1/2} \log p$. If $d^2 > \epsilon r$, so that $h > p$, we apply Lemma 2.4 with $p = q$, $m = n = 1$ and $A = \mathbb{F}_p$ to compute $\#\mathcal{C}$ exactly, with $O(pM(d) \log(dp))$ operations in $\mathbb{F}_p$. Otherwise, we set $A = \{0, \ldots, h-1\} \subseteq \mathbb{F}_p$, compute $\#\mathcal{C}(A)$ by Lemma 2.4 with $O(hM(d) \log(dp))$ operations in $\mathbb{F}_p$, and return $\gamma = p\#\mathcal{C}(A)/h$. The claim follows from Theorem 2.3 (i).

(ii) Let σ be the number of these components, let $\epsilon = 1/3$, and h as in (i). Since $h \leq p$, we have from Theorem 2.3 (ii)

$$|\#\mathcal{C}(A)/h - \sigma| \leq 1/3,$$

so that it is easy to determine the integer σ. □

In Theorem 5.3 below we give a different (probabilistic) algorithm which is faster than the one of Corollary 2.5 if d is small and p is large.

3. Estimating the size of a projection

In this section, we present an algorithm to estimate the size of the projection of a plane curve, and the size of the image of a univariate polynomial.

Let $\mathcal{C} = \{f = 0\} \subseteq \mathbb{F}_q^2$ be a curve without vertical lines, given by $f \in \mathbb{F}_q[x, y]$ of degree d, and let

$$\pi : \mathcal{C} \to \mathbb{F}_q \tag{3.1}$$

be the first projection, with $\pi((a,b)) = a$. Furthermore, let $A \subseteq \mathbb{F}_q$, $i \in \mathbb{N}$, and define

$$R_i(A) = \{a \in A : \#\pi^{-1}(\{a\}) = i\}, \quad r_i(A) = \#R_i(A). \tag{3.2}$$

We are interested in the two cases where $A = \mathbb{F}_q$, or where q is a prime and $A = \{0, \ldots, h-1\}$ for some h near $q^{1/2}$. Since A is fixed for most of this section, we usually write R_i and r_i.

For our task, it is sufficient to determine the more informative $r_1, \ldots, r_d$, since the size of the projection satisfies

$$\#\pi(\mathcal{C}) = \sum_{1 \le i \le d} r_i. \tag{3.3}$$

The idea is to reduce the determination of r_i to the size of certain associated curves $S_1, \ldots, S_d$. Then an estimate of $\#S_k$ à la Weil, with an error term $O(q^{1/2})$, will lead to an estimate of the r_i's with error $O(q^{1/2})$; we call this a "computational Weil estimate". The constants implied in the Big-Oh depend exponentially on d.

We may assume that the defining polynomial f is squarefree, since repetitions of factors do not change $\mathcal{C}$, and we can easily compute the squarefree part of an arbitrary f. Furthermore, we may assume that $\mathcal{C}$ contains no vertical lines $\{x = a\}$; any such lines are easily removed by dividing f by the gcd in $\mathbb{F}_q[x]$ of all coefficients of powers of y. Thus $R_i = \emptyset$ for $i > d$.

For $1 \le k \le d$, let

$$\begin{aligned} S_k = \{(a_0, a_1, \ldots, a_k) \in \mathbb{F}_q^{k+1} : f(a_0, a_1) = \cdots = f(a_0, a_k) = 0, \\ a_i \ne a_j \text{ for } 1 \le i < j \le k\}, \\ s_k = \#S_k. \end{aligned}$$

Under the surjective map $S_k \to \bigcup_{k \le i \le n} R_i$ with $(a_0, \ldots, a_k) \mapsto a_0$, each $a \in R_i$ has exactly $i_{(k)}$ images, where $i_{(k)} = i \cdot (i-1) \cdot \cdots \cdot (i-k+1)$ is the Pochhammer symbol. Thus

$$s_k = \sum_{k \le i \le d} i_{(k)} r_i \quad \text{for } 1 \le k \le d. \tag{3.4}$$

We now show how to determine the r_i's explicitly from the s_k's, using (3.4), then how to obtain approximations for the r_i's from approximations to the s_k's, and finally how to obtain approximations to the s_k's.

LEMMA 3.1. *(3.4) is equivalent to*

$$r_i = \frac{1}{i!} \sum_{i \le k \le d} \frac{(-1)^{i+k} s_k}{(k-i)!} \quad \text{for } 1 \le i \le d.$$

PROOF. We consider (3.4) as a system of linear equations in $\mathbb{Q}(s_1, \ldots, s_d)$, with indeterminates $s_1, \ldots, s_d$. Since it is triangular with nonzero entries $k_{(k)} = k!$ on the diagonal, it has a unique solution. Thus it is sufficient to show that the quantities stated satisfy (3.4), i.e., that s_k equals

$$\sum_{k \le i \le d} i_{(k)} \cdot \frac{1}{i!} \sum_{i \le j \le d} \frac{(-1)^{i+j} s_j}{(j-i)!} = \sum_{k \le j \le d} s_j \sum_{k \le i \le j} \frac{(-1)^{i+j} i!}{(i-k)! i! (j-i)!}.$$

Consider some j with $k \le j \le d$. The coefficient of s_j in the last expression equals

$$\begin{aligned} \sum_{k \le i \le j} \frac{(-1)^{i+j}}{(i-k)!(j-i)!} &= \sum_{0 \le l \le j-k} \frac{(-1)^{l+k+j}}{l!(j-k-l)!} \\ &= \frac{(-1)^{k+j}}{(j-k)!} \sum_{0 \le l \le j-k} (-1)^l \binom{j-k}{l} = \delta_{jk}. \qquad \square \end{aligned}$$

COROLLARY 3.2. *In the above notation, we have*

$$\#\pi(\mathcal{C}) = \sum_{1 \le k \le d} \frac{(-1)^{k+1} s_k}{k!}.$$

PROOF. By (3.3), we have

$$\begin{aligned} \#\pi(\mathcal{C}) &= \sum_{1 \le i \le d} \sum_{i \le k \le d} \frac{(-1)^{i+k} s_k}{i!(k-i)!} \\ &= \sum_{1 \le k \le d} \sum_{1 \le i \le k} (-1)^i \binom{k}{i} \cdot \frac{(-1)^k s_k}{k!} = \sum_{1 \le k \le d} \frac{(-1)^{k+1} s_k}{k!}. \qquad \square \end{aligned}$$

Suppose that we have an approximation σ_k to s_k for $1 \le k \le d$, so that

$$\epsilon_k = s_k - \sigma_k$$

is small in absolute value. If we set

$$\rho_i = \frac{1}{i!} \sum_{i \le k \le d} \frac{(-1)^{i+k} \sigma_k}{(k-i)!} \tag{3.5}$$

for $1 \le i \le d$, then ρ_i is an approximation to r_i, since

$$r_i - \rho_i = \frac{1}{i!} \sum_{i \le k \le d} \frac{(-1)^{i+k} \epsilon_k}{(k-i)!}. \tag{3.6}$$

If $q = p$ is a prime and not too large relative to d, we can apply our "strip counting" method from Section 2 to the present problem.

ALGORITHM 3.3. *Projection size estimation.*
Input: *A prime p, $f \in \mathbb{F}_p[x, y]$ of degree d and with $\mathrm{cont}_x f = 1$, and h with $1 \le h < p/2$.*
Output: *An estimate ρ_i of $r_i = r_i(\mathbb{F}_q)$, as in (3.2), for $1 \le i \le d$.*

1. *For each $a \in A = \{0, \ldots, h-1\}$, determine*

$$\#\pi^{-1}(\{a\}) = \deg \gcd(f(a, y), y^p - y).$$

2. *For $1 \le i \le d$, return*

$$\rho_i = \#\{a \in A : \#\pi^{-1}(\{a\}) = i\}.$$

THEOREM 3.4. *Let p, f, d, h be an input to the above algorithm and $d \ge 4$. The algorithm is deterministic and can be performed with $O(hM(d) \log pd)$ operations in $\mathbb{F}_p$, and the output satisfies for all $i \le d$*

$$|r_i - \rho_i| \le d^{2d} h^{-1} p^{3/2} \log p.$$

PROOF. Since one gcd calculation can be done with $O(M(d) \log p + M(d) \log d)$ operations, the whole algorithm takes

$$O(hM(d) \log(pd)) \quad \text{or} \quad O(hd \, \log d \, \mathrm{loglog}\, d \, \log pd)$$

operations in $\mathbb{F}_p$.

For $1 \le k \le d$, we consider the curve $\mathcal{S}_k \subseteq \mathbb{F}_p^{k+2}$ given by the $k+1$ polynomials

$$\begin{gathered} f(x_0, x_1), f(x_0, x_2), \ldots, f(x_0, x_k), \\ y \cdot \prod_{1 \le i < j \le k} (x_j - x_i) - 1 \in \mathbb{F}_q[x_0, \ldots, x_k, y], \end{gathered} \tag{3.7}$$

and the number

$$t_k = \#\mathcal{S}_k(A)$$

of points on $\mathcal{S}_k$ over A, so that

$$t_k = \sum_{k \le i \le d} i_{(k)} \rho_i.$$

The curve $\mathcal{S}_k$ is isomorphic to the curve given by the first k equations in (3.7) minus the diagonals, and thus the Zariski closure of the set in $\mathbb{F}_q^{k+2}$ corresponding to S_k, and its degree is at most d^k. We claim that no absolutely irreducible component $\mathcal{D}$ of $\mathcal{S}_k$ is vertical, i.e., contained in $\{a\} \times \mathbb{F}_q^{k+1}$ for some $a \in \mathbb{F}_q$. Otherwise there is a projection $\pi_i \colon \mathbb{F}_q^{m+2} \to \mathbb{F}_q^2$ with

$$\pi_i(a_0, a_1, \ldots, a_k, b) = (a_0, a_i)$$

for some $i \le k$ such that $\pi_i(\mathcal{D})$ is a curve; but then its closure is a vertical component of $\mathcal{C}$, contradicting our assumption.

Let $\sigma_k = pt_k/h$. By Theorem 2.3(i), we have

$$|s_k - \sigma_k| = p/h \cdot |h\#\mathcal{S}_k/p - \#\mathcal{S}_k(A)| \le p/h \cdot 2.03\, d^{2k} p^{1/2} \log p.$$

In step 2, ρ_i satisfies (3.5), and we find from (3.6)

$$|r_i - \rho_i| \le \frac{1}{i!} \sum_{i \le k \le d} \frac{2.03\, d^{2k} h^{-1} p^{3/2} \log p}{(k-i)!} \le d^{2d} h^{-1} p^{3/2} \log p. \qquad \square$$

For $d \le 3$, we find $|r_i - \rho_i| \le 3d^{2d} h^{-1} p^{3/2} \log p$. As an example, suppose that $d^{2d+1} \le p^{1/8}$, and set $h = p^{3/4}$. Then we obtain with $O\tilde{}(p^{3/4})$ operations in $\mathbb{F}_p$ an approximation of the r_i's with error $O\tilde{}(p^{7/8})$.

Of course, we can also estimate $\#\pi(\mathcal{C})$ via the resultant

$$r = \mathrm{res}_y(f(x, y), y^q - y) \in \mathbb{F}_q[x],$$

whose set of roots in $\mathbb{F}_q$ equals $\pi(\mathcal{C})$. Unfortunately this does not help much as the degree of r is too large.

We apply our technique to estimate the size of the image of a mapping $\mathbb{F}_q \to \mathbb{F}_q$ given by a univariate rational function.

COROLLARY 3.5. *Let $d \geq 4$, $f = g_1/g_2 \in \mathbb{F}_p(x)$ with $g_1, g_2 \in \mathbb{F}_p[x]$ relatively prime and of degree less than d, $0 < h < p$, and*

$$u_i = \#\{a \in \mathbb{F}_p : \#f^{-1}(\{a\}) = i\}$$

be the number of points in $\mathbb{F}_q$ with exactly i preimages under the associated partial mapping $f : \mathbb{F}_p \to \mathbb{F}_p$, and

$$V(f) = \sum_{1 \leq i \leq d} u_i$$

the number of images of f. We can compute with $O(hM(d) \log d)$ operations in $\mathbb{F}_p$ approximations v_i and ϕ such that for all $i \leq d$

$$|u_i - v_i|,\ d^{-1}|V(f) - \phi| \leq d^{2d} h^{-1} p^{3/2} \log p.$$

PROOF. We apply Theorem 3.4 to the curve $\mathcal{C} = \{g_1(y) = xg_2(y)\}$. □

A formula essentially equivalent to Corollary 3.2 in the particular case of $f = g_1 \in \mathbb{F}_q[x]$ is in Birch & Swinnerton-Dyer (1959).

The results of this and the previous section can be extended to counting complete intersections over finite fields, by using the results of Shparlinski & Skorobogatov (1990) on the distribution of points on such varieties instead of Lemma 2.1.

4. $\#\mathcal{P}$-completeness of sparse curve counting

In this section, we show that the counting problem for sparsely represented curves over finite fields is $\#\mathcal{P}$-complete, via probabilistic Turing reductions.

We rely on hardness results proved by Quick (1986), which in turn are based on Plaisted's (1977, 1984) work. A bonus of our approach is that we have upper bounds matching the previously known lower bounds, so that now some variants of Plaisted's and Quick's problems are shown to be complete for $\mathcal{NP}$ and $\#\mathcal{P}$, respectively.

Plaisted (1984), Theorem 5.1, associates to a given Boolean formula F in 3-conjunctive normal form polynomials $f_1, \ldots, f_n \in \mathbb{Z}[x]$ such that F is satisfiable if and only if $\gcd(f_1, \ldots, f_n) \neq 1$. This shows that the latter problem is $\mathcal{NP}$-hard. Quick gives a clever parsimonious variation of this reduction, i.e., such that the number of satisfying assignments of F can be efficiently computed from $\deg\gcd(f_1, \ldots, f_n)$ for several instances of such polynomials. This shows that the latter problem is $\#\mathcal{P}$-hard.

Here are the formal definitions—in the style of Garey & Johnson (1979)—of our counting problems. All input numbers are represented in binary, and for the elements of $\mathbb{Z}_p$, we use the representatives $\{0, 1, \ldots, p-1\}$. The sparse representation of a uni- or bivariate polynomial is a list of pairs (coefficient, exponent), and the dense representation lists all coefficients up to the degree. A finite field $\mathbb{F}_q$ with $q = p^m$ elements, where p is prime, is represented by (p, ψ), where $\psi \in \mathbb{Z}_p[x]$ is monic irreducible of degree m. An element u of $\mathbb{F}_q$ is represented by the usual coefficient vector $(u_0, \ldots, u_{m-1}) \in \mathbb{Z}_p^m$, with $u \equiv \sum_{0 \le i < m} u_i x^i \bmod \psi$.

Problem 4.1. CyclotomicGcd.
Instance: $n \in \mathbb{N}$, *pairwise distinct primes* $p_1, \ldots, p_n \le n^2$, $N = p_1 \cdots p_n$, *and the sparse representations of* $f_1, \ldots, f_n \in \mathbb{Z}[x]$, *where* f_i *divides* $x^N - 1$ *in* $\mathbb{Z}[x]$ *for* $1 \le i \le n$.
Output: $\deg\gcd(f_1, \ldots, f_n)$.

The bound $p_i \le n^2$ is rather arbitrary.

Problem 4.2. SpecialCyclotomicGcd.
Instance: $n, p_1, \ldots, p_n, N, f_1, \ldots, f_n$ *as in* CyclotomicGcd, *and a prime* q *such that* $q \equiv 1 \bmod N$.
Output: $\deg\gcd(f_1, \ldots, f_n)$.

Problem 4.3. CommonRootsMany.
Instance: A finite field $\mathbb{F}_q$, *and the sparse representations of* $f_1, \ldots, f_n \in \mathbb{F}_q[x]$.
Output: The number $\deg\gcd(f_1, \ldots, f_n, x^q - x)$ *of distinct common roots of* $f_1, \ldots, f_n$ *in* $\mathbb{F}_q$.

Problem 4.4. CommonRoots.
Instance: A finite field $\mathbb{F}_q$, *and the sparse representations of* $f, g \in \mathbb{F}_q[x]$.
Output: The number $\deg\gcd(f, g, x^q - x)$ *of distinct common roots of* f *and* g *in* $\mathbb{F}_q$.

Problem 4.5. SparseCurves.
Instance: A finite field $\mathbb{F}_q$, *and the sparse representation of a polynomial* $f \in \mathbb{F}_q[x, y]$.
Output: The number of points on the curve $\mathcal{C} = \{(a, b) \in \mathbb{F}_q^2 : f(a, b) = 0\}$.

Furthermore, we let #3Sat be the $\#\mathcal{P}$-complete problem of counting the number of satisfying truth assignments for a propositional formula in 3-CNF form. The following result can be obtained from Quick's (1986) work and is the basis for our completeness proofs.

Fact 4.6. #3Sat $\leq$ CyclotomicGcd, *via a deterministic parsimonious Turing reduction, and* CyclotomicGcd *is* $\#\mathcal{P}$*-hard.*

The proof that SparseCurves is $\#\mathcal{P}$-complete proceeds in three stages: first we exhibit probabilistic polynomial-time parsimonious Turing reductions

$$\begin{aligned} &\text{SpecialCyclotomicGcd} \leq \text{CommonRootsMany} \\ &\leq \text{CommonRoots} \leq \text{SparseCurves}, \end{aligned} \tag{4.1}$$

and then show how to find certain prime numbers efficiently, so that Quick's reduction actually implies that #3Sat$\leq$ SpecialCyclotomicGcd. Finally, it is easy to see that SparseCurves $\in \#\mathcal{P}$.

For the first reduction, let $(n, p_1, \ldots, p_n, N, f_1, \ldots, f_n, q)$ be an instance of SpecialCyclotomicGcd. By assumption, all f_i and $g = \gcd(f_1, \ldots, f_n) \in \mathbb{Z}[x]$ are monic. For any $u \in \mathbb{Z}[x]$, we denote by $\overline{u} \in \mathbb{F}_q[x]$ the polynomial obtained by taking each coefficient modulo q. The reduction maps the given instance to $(\mathbb{F}_q, \overline{f_1}, \ldots, \overline{f_n})$. We have

$$x^N - 1 = \prod_{j|N} \Phi_j \in \mathbb{Z}[x],$$

where $\Phi_j \in \mathbb{Z}[x]$ is the jth cyclotomic polynomial. Thus

$$x^N - 1 = \prod_{j|N} \overline{\Phi}_j$$

in $\mathbb{F}_q[x]$, and since $x^N - 1 \in \mathbb{F}_q[x]$ is squarefree, it follows that the $\overline{\Phi}_j$'s are pairwise relatively prime. For $1 \leq i \leq n$, let $A_i \subseteq \mathbb{N}$ be such that $f_i = \prod_{j \in A_i} \Phi_j$. Then

$$\overline{f}_i = \prod_{j \in A_i} \overline{\Phi_j}, \quad g = \prod_{j \in A} \Phi_j,$$

where $A = A_1 \cap \cdots \cap A_n$. Since $x^N - 1$ splits into linear factors in $\mathbb{F}_q[x]$ (see Lidl & Niederreiter (1983), Theorem 2.47), we have

$$\overline{g} = \gcd(\overline{f_1}, \ldots, \overline{f_n}) = \gcd(\overline{f_1}, \ldots, \overline{f_n}, x^q - x) = \prod_{j \in A} \overline{\Phi_j},$$

and $\deg g = \deg \overline{g}$.

For the second reduction in (4.1), we need a lemma; see Theorem 6.2 and "Note Added in Proof" of Kaltofen (1988) for multivariate polynomials.

LEMMA 4.7. *Let F be a field, $f_1, \ldots, f_n \in F[x]$ of degree at most d, $h = \gcd(f_1, \ldots, f_n)$, $A \subseteq F$ finite, $a_3, \ldots, a_n \in A$ uniformly chosen random elements, and $g = f_2 + \sum_{3 \le i \le n} a_i f_i \in F[x]$. Then h divides* $\gcd(f_1, g)$, *and*

$$prob\{h = \gcd(f_1, g)\} \ge 1 - d/\#A.$$

PROOF. After dividing each f_i by h, we may assume that $h = 1$, and also that $f_1 \neq 0$. Let $A_3, \ldots, A_n$ be new indeterminates over $F(x)$, $R = F[A_3, \ldots, A_n]$, $G = f_2 + \sum_{3 \le i \le n} A_i f_i \in R[x]$, $r = \mathrm{res}_x(f_1, G) \in R$. Then r is a polynomial in $A_3, \ldots, A_n$ of degree at most d, and if $r(a_3, \ldots, a_n) \neq 0$, then $\gcd(f_1, g) = 1$.

Any factor u of $\gcd(f_1, G)$ in $R[x]$ is in fact in $F[x]$, and therefore u divides $f_1, \ldots, f_n$. It follows that $u \in F$, $\gcd(f_1, G) = 1$, and $r \neq 0$. Now Schwartz's (1980) Lemma implies the claim. □

For the second reduction, let $(\mathbb{F}_q, f_1, \ldots, f_n)$ be an instance of COMMONROOTSMANY, and d the maximal degree of $f_1, \ldots, f_n$. If $q \ge 2d$, we choose $a_3, \ldots, a_n \in \mathbb{F}_q$ at random and set

$$g = f_2 + \sum_{3 \le i \le n} a_i f_i \in \mathbb{F}_q[x].$$

By Lemma 4.7, the reduction to the instance $(\mathbb{F}_q, f_1, g)$ of COMMONROOTS works correctly. If $q < 2d$, we first construct a field extension of $\mathbb{F}_q$ of degree $\lceil \log_q d \rceil$ and make our random choices from this new field. This does not change the gcd to be computed. All of this can be done in random polynomial time.

The third reduction in (4.1) is given by

$$(\mathbb{F}_q, f, g) \mapsto (\mathbb{F}_q, h)$$

where $f, g \in \mathbb{F}_q[x]$ and $h = f - gy \in \mathbb{F}_q[x, y]$. We let $c = \#\mathcal{C}$ be the size of the curve $\mathcal{C} = \{(a, b) \in \mathbb{F}_q^2 : h(a, b) = 0\}$ described by h; this number c is returned by the oracle for SPARSECURVES. The reduction returns

$$d = \lceil c/q \rceil - 1$$

if $q \nmid c$ or $g(0) \neq 0$, and

$$d = c/q$$

if $q \mid c$ and $g(0) = 0$. Let

$$d^* = \deg \gcd(f, g, x^q - x)$$

be the output required by CommonRoots. We prove $d^* = d$; since $g(0)$ can easily be computed and $c \leq q^2$, this will show that we have a (deterministic) polynomial-time reduction.

Let $k = \deg \gcd(g, x^q - x)$ be the number of distinct roots of g in $\mathbb{F}_q$; then $0 \leq k \leq q$. For $(a, b) \in \mathbb{F}_q^2$, we have

$$\begin{aligned} (a, b) \in \mathcal{C} &\iff h(a, b) = 0 \\ &\iff (g(a) \neq 0 \text{ and } b = f(a)/g(a)) \\ &\qquad \text{or } f(a) = g(a) = 0. \end{aligned}$$

Thus

$$c = (q - k) + qd^*, \quad d^* = \frac{c + k}{q} - 1.$$

Since $k \leq q$, this equals $\lceil c/q \rceil - 1$ if $q \nmid c$. If $q \mid c$, it equals $c/q - 1$ if $k = 0$ (and then $g(0) \neq 0$), and c/q if $k \geq 1$ (and then $k = q$ and $g(0) = 0$). This shows $d^* = d$, and finishes the reductions in (4.1).

The following is Quick's (1986) reduction from #3Sat to Cyclotomic-Gcd. He uses certain polynomials $\mathrm{Poly}(F_j, Q_i) \in \mathbb{Z}[x]$ introduced by Plaisted.

Algorithm 4.8. *Quick's reduction.*
Input: *A Boolean formula $F = F_1 \wedge \cdots \wedge F_k$ on n variables in 3-CNF form.*
Output: *The number s of truth assignments that satisfy F.*

1. *Let $r_1, \ldots, r_n$ be the n smallest prime numbers satisfying $r_i/(\log r_i)^2 \geq 1500n$. For $1 \leq i \leq n$, let $P_i = (p_{i1}, \ldots, p_{in})$ be the vector of the n smallest prime numbers in the arithmetic progression $\{2 + r_i a : a \in \mathbb{N}\}$.*

2. *For $i = 1, \ldots, n$ do the following steps.*

   i. *For $j = 1, \ldots, k$ compute the sparse representation of*

   $$f_{ij} = \mathit{Poly}(F_j, P_i) \in \mathbb{Z}[x].$$

   [Then each f_{ij} divides $x^{N_i} - 1$, where $N_i = p_{i1} \cdots p_{in}$.]

ii. *Call* CyclotomicGcd *to compute* $s_i = \deg \gcd(f_{i1}, \ldots, f_{ik})$.

3. *Compute* $s \in \mathbb{N}$ *such that* $s < r_1 \cdots r_n$ *and* $s \equiv s_i \bmod r_1 \cdots r_n$ *for* $1 \leq i \leq n$.

The missing link in our reductions is to prove that #3Sat≤ Special-CyclotomicGcd, similar to Fact 4.6. For this, we have to be able to find efficiently for all $i, j \leq n$ the primes r_i, p_{ij} of step 1., and furthermore primes q_i such that

$$q_i \equiv 1 \bmod \prod_{1 \leq j \leq n} p_{ij}, \tag{4.2}$$

as in SpecialCyclotomicGcd. The generally valid estimates for the number of primes in arithmetic progressions are not good enough for this task. Fortunately, there are better estimates that are valid for all but a few "exceptional" moduli; see Davenport (1980), McCurley (1984). Our argument is based on the following special case of Theorem 2.1 from Alford *et al.* (1993), where we have set $A = 49/20$, $\epsilon = 1/2$, and $\delta = 2/245$ in that Theorem. We denote, as is usual, by $\pi(y; d, a)$ the number of primes up to y in the arithmetic progression $a + d\mathbb{N}$, so that $\pi(y) = \pi(y; 1, 1)$.

Fact 4.9. *(Alford, Granville & Pomerance 1993) There exist positive* x_0, $\eta \in \mathbb{R}$ *and* $t \in \mathbb{N}$ *such that for all* $x, y \in \mathbb{R}$ *with* $y \geq x \geq x_0$ *there exists* $E \subset \mathbb{N}$ *with* $\#E < t$ *and* $e \geq \log x$ *for all* $e \in E$, *and such that*

$$\pi(y; d, a) \geq \frac{\pi(y)}{2\phi(d)}$$

whenever $\gcd(a, d) = 1$, $1 \leq d \leq \min\{x^{2/5}, yx^{-3/5}\}$, *and* d *is not divisible by any* $e \in E$. *Furthermore, there exists* e_0 *such that* $e \geq x^\eta$ *for all* $e \in E$ *with* $e \neq e_0$.

We now consider the following algorithm, which takes as inputs n and $c \geq 1$. It also uses x_0, η, and t as above.

1. Set $B = 3cn \log^2 n$ and determine the first $n + 1$ primes $r_0, \ldots, r_n$ that are at least $B/2$.

2. For $i = 0, \ldots, n$, determine the first nt primes $p_{i,1}, \ldots, p_{i,nt}$ in the arithmetic progression $2 + r_i\mathbb{N}$, by consecutively testing its members for primality. If a member greater than $x = \max\{B^{1/\eta}, x_0\}$ has to be tested,

mark i as "exceptional" and abandon that i. [At most one number i will be so marked.] Renumber the r_i's so that $r_1, \ldots, r_n$ are not marked "exceptional".

3. Set $y = x^{5n/2}$ and $m = \lceil 5n \log x \cdot \log(2/\gamma) \rceil$. For $i = 1, \ldots, n$, do the following.

   i. For $1 \le k \le t$, set $d_{ik} = \prod_{(k-1)n < j \le kn} p_{ij}$.

   ii. For $1 \le k \le t$, choose m integers $t_{ik\ell}$ in $\{1, \ldots, \lfloor y/d_{ik} \rfloor\}$ uniformly at random, for $1 \le \ell \le m$.

   iii. For $1 \le k \le t$ and $\ell = 1, \ldots, m$, test $1 + t_{ik\ell} \cdot d_{ik}$ for primality. If some prime q_i is found, then include this prime in the output, else return "failure".

For the primality test in the last step, we may use any of the usual probabilistic tests with an additional input $\gamma > 0$: it takes time polynomial in the input size and $\log \gamma^{-1}$, returns the correct answer if the input is prime, and for a composite input, it returns the correct answer with probability at least $1 - \gamma/2tn^2$. In Fact 4.9, we may assume that $\eta \le 2/7$. We set

$$N = \max\{2 \cdot 10^{34}, 2/5\eta, 100t^2, c^{10}\}.$$

THEOREM 4.10. *The above algorithm uses $(n \log \gamma^{-1})^{O(1)}$ bit operations and, if $n \ge N$, it returns with probability at least $1 - \gamma$ values which have the following properties for $1 \le i \le n$:*

(i) a prime r_i such that $cn \log^2 r_i \le B/2 \le r_i \le B$ and $r_j \ne r_i$ for all $j \ne i$,

(ii) distinct primes $p_{i1}, \ldots, p_{in}$ such that $p_{ij} \le \max\{B^{1/\eta}, x_0\}$ and $p_{ij} \equiv 2 \bmod r_i$ for all j,

(iii) a prime q_i satisfying (4.2).

PROOF. The time bound is clear. We will show that (i) and (ii) always hold, and that (iii) is probably true.

For (i), we have by Rosser & Schoenfeld (1962, 3.8) that

$$\pi(B) - \pi(B/2) \ge \frac{3B}{10 \log(B/2)} = \frac{9n \log^2 n}{10} \cdot \frac{c}{\log c + \log(\frac{3}{2} n \log^2 n)}.$$

Since $c \geq 1$ and $n \geq 11$, for any $A \geq 1$ we have

$$\log(\frac{3}{2}n\log^2 n) \geq 1, \qquad \frac{c}{\log c + A} \geq \frac{1}{A},$$
$$\frac{9}{10}n(\log n)^{1/4} \geq n+1, \qquad (\log n)^{7/4} \geq \log(\frac{3}{2}n\log^2 n).$$

This implies that

$$\pi(B) - \pi(B/2) \geq \frac{9n\log^2 n}{10} \cdot \frac{1}{\log(\frac{3}{2}n\log^2 n)} \geq n+1,$$

so that $r_n \leq B$. The assumption that $n \geq N$ implies that

$$\begin{aligned} n^{1/10} &\geq c, \\ n^{\sqrt{3/2}-11/10} &\geq 3\log^2 n, \\ n^{\sqrt{3/2}-1} &\geq 3c\log^2 n, \\ \frac{3}{2}\log^2 n &\geq \log^2(3cn\log^2 n) = \log^2 B, \end{aligned}$$

and for all $i \leq n$

$$r_i \geq r_0 \geq B/2 = \frac{3}{2}cn\log^2 n \geq cn\log^2 B \geq cn\log^2 r_n \geq cn\log^2 r_i.$$

For (ii), we first note that $r_i \leq x^\eta$ for each $i \leq n$. The assumptions that $\eta \leq 2/7$ and $n \geq N$ imply that

$$\begin{aligned} (2-5\eta)/2\eta &\geq 1, \\ \eta B^{(2-5\eta)/2\eta} &\geq \eta n^{(2-5\eta)/2\eta} \geq \frac{2}{5}, \\ \eta B^{1/\eta} &\geq \frac{2}{5}B^{5/2}. \end{aligned}$$

Since $\pi(x) \geq x/\log x$, Fact 4.9 says that for all i with at most one exception (namely, when possibly $r_i = e_0$) we have

$$\begin{aligned} \pi(x; r_i, 2) &\geq \frac{\pi(x)}{2\phi(r_i)} \geq \frac{B^{1/\eta}}{2\log(B^{1/\eta})B} \\ &= \frac{\eta B^{1/\eta - 1}}{2\log B} \geq \frac{\frac{2}{5}B^{3/2}}{2\log B} = \frac{(3cn\log^2 n)^{3/2}}{5\log(cn\log^2 n)}. \end{aligned}$$

We have $c^{3/2} \geq \log c$, $n \geq \log^2 n$, and

$$3^{3/2} n^{1/2} \log^2 n > n^{1/2} \geq 10t,$$

so that

$$\begin{aligned} \frac{(3cn\log^2 n)^{3/2}}{5\log(cn\log^2 n)} &\geq \frac{3^{3/2}c^{3/2}n^{3/2}\log^3 n}{5(\log c + \log(n^2))} \\ \geq \frac{3^{3/2}c^{3/2}n^{3/2}\log^3 n}{5\log c \cdot \log(n^2)} &\geq tn \cdot \frac{3^{3/2}n^{1/2}\log^3 n}{10t\log n} > tn. \end{aligned}$$

This shows that for all but at most one i we will find tn primes $p_{ij} \leq x$, and that indeed at most one i will be marked "exceptional" in step 2.

For (iii), fix some i with $1 \leq i \leq n$. Since $d_{i1}, \ldots, d_{it}$ are pairwise relatively prime, each $e \in E$ divides at most one of them, and some d_{ij} is not a multiple of any $e \in E$; denote this value of j by j_0, and $d = d_{ij_0}$. We show that step 3.iii for $j = j_0$ is likely to return a prime satisfying (iii). We have

$$\pi(y; d, 1) \geq \frac{\pi(y)}{2\phi(d)} \geq \frac{y}{2\log y \cdot d}.$$

Therefore, for $q = 1 + td$ with $t \in \{1, \ldots, \lfloor y/d \rfloor\}$ random, we have

$$\operatorname{prob}(q \text{ prime}) \geq \frac{\pi(y; d, 1)}{\lfloor y/d \rfloor} \geq \frac{y}{2\log y \cdot d \cdot y/d} = \frac{1}{2\log y} = \frac{1}{5n\log x}.$$

Therefore, the probability that none of the random choices gives a prime number is at most

$$\left(1 - \frac{1}{5n\log x}\right)^m \leq \left(1 - \frac{1}{5n\log x}\right)^{5n\log x \cdot \log(2/\gamma)} \leq e^{-\log(2/\gamma)} = \frac{\gamma}{2}.$$

Furthermore, the probability that all primality tests answer correctly is at least

$$\left(1 - \frac{\gamma}{2tn^2}\right)^{tn^2} \geq 1 - \frac{\gamma}{2},$$

so that step 3.iii returns a prime number q_i with probability at least $(1 - \frac{\gamma}{2})^2 \geq 1 - \gamma$. $\square$

We can now prove the main result of this Section.

THEOREM 4.11. *The problems* SPECIALCYCLOTOMICGCD, COMMONROOTS, COMMONROOTSMANY, *and* SPARSECURVES *are* $\#\mathcal{P}$*-complete under probabilistic polynomial-time parsimonious Turing reductions.*

PROOF. Using Theorem 4.10, we can implement Quick's reduction so that it actually yields $\#3$SAT $\leq$ SPECIALCYCLOTOMICGCD. By the $\#\mathcal{P}$-completeness of $\#3$SAT and (4.1), it is sufficient to see that SPARSECURVES $\in \#\mathcal{P}$. We consider a non-deterministic Turing machine which on input $(\mathbb{F}_q, f)$ generates all pairs $(a, b) \in \mathbb{F}_q^2$ and checks for each whether $(a, b) \in \mathcal{C} = \{f = 0\}$. If so, it accepts, and otherwise it rejects. Thus the number of accepting computations is $\#\mathcal{C}$. Each check can be done in polynomial time. □

5. Reduction to absolutely irreducible curves

In this section, we show that the computation of the curve size can be reduced in random polynomial time to the same question about absolutely irreducible curves. This is the case to which Weil's Theorem applies.

So now we are given $f \in \mathbb{F}_q[x, y]$, and want to compute $\#\mathcal{C}$ for $\mathcal{C} = \{f = 0\} \subseteq \mathbb{F}_q^2$. We factor f as

$$f = f_1 \cdots f_\tau,$$

with $f_1, \ldots, f_\tau \in \mathbb{F}_q[x, y]$ irreducible.

There are several algorithms in the literature for this factorization: Chistov & Grigoryev (1982), Lenstra (1985), and von zur Gathen & Kaltofen (1985). The latter paper gives $O\tilde{\ }(n^7 \log^2 q(n^5 + \log q))$ bit operations. These algorithms are probabilistic of the Las Vegas type, i.e., they never return an incorrect answer, but they may fail, with controllably small probability. Shparlinski (1992a), Theorem 1.7, gives a deterministic method whose cost is $O(n^{3.7} \log q)$ for *almost all* input polynomials.

We will assume from now on, without loss of generality, that f is squarefree.

Next we determine for each f_i whether it is *absolutely irreducible*, i.e., irreducible over an algebraic closure of $\mathbb{F}_q$. Kaltofen's (1985) algorithm seems to take $O\tilde{\ }(n^8 + n^2 \log q)$ operations in $\mathbb{F}_q$.

We order the irreducible factors of f so that $f_1, \ldots, f_\sigma$ are absolutely irreducible and $f_{\sigma+1}, \ldots, f_\tau$ are not, for some $\sigma \le \tau$. For $1 \le i, j \le \tau$, we set

$$\begin{array}{rclrcl} U_i & = & \{f_i = 0\} \subseteq \mathbb{F}_q^2, & u_i & = & \#U_i, \\ V_{ij} & = & \{f_i = f_j = 0\} \subseteq \mathbb{F}_q^2, & v_i & = & \# \bigcup_{k \ne i} V_{ik}. \end{array}$$

Since $\mathcal{C} = \bigcup_{1 \le i \le \tau} U_i$, we have

$$\#\mathcal{C} = \sum_{1 \le i \le \tau} (u_i - v_i) + \# \bigcup_{1 \le i < j \le \tau} V_{ij}.$$

We now show show to compute all these quantities quickly, except for the u_i with $i \le \sigma$.

Lemma 5.1. *Let F be a field, and $f, g \in F[x, y]$ have total degree at most d. Then the resultant*

$$r = \mathrm{res}_y(f, g) \in F[x]$$

can be computed with $O(M(d^3) \log d)$ operations in F.

Proof. It is well-known that r can be calculated by a fast continued-fraction algorithm in $F(x)[y]$ and using the subresultant description of r with $O(M(d) \log d)$ operations in $F(x)$; see, e.g., von zur Gathen (1991), steps 4, 5, 6 of the algorithm in Section 2. By the subresultant theory, the degree in x of each intermediate result is at most d^2, so that the total cost is $O(M(d)M(d^2) \log d)$ operations in F. In fact, using Lemma 2.2 from von zur Gathen & Shoup (1992), this can be done with $O(M(d^3) \log d)$ operations. $\square$

Lemma 5.2. *Let $f, g \in \mathbb{F}_q[x, y]$ have total degree m, n, respectively, and assume that $m, n \le d$.*

(i) If $\gcd(f, g) = 1$, then

$$V = \{(a, b) \in \mathbb{F}_q^2 : f(a, b) = g(a, b) = 0\}$$

can be computed probabilistically with $O((d^4 + d^2 \log q) \cdot \log^2 d \,\mathrm{loglog}\, d)$ operations in $\mathbb{F}_q$, and

$$\#V \le mn.$$

(ii) *If f is irreducible and not absolutely irreducible, then*

$$U = \{(a,b) \in \mathbb{F}_q^2 : f(a,b) = 0\}$$

can be computed probabilistically with $O\tilde{}(d^{11}\log^2 q\,(d^4 + \log q))$ operations in $\mathbb{F}_q$, and

$$\#U \leq d^2/4.$$

PROOF. (i) The classical method for finding V is to compute the resultant

$$r = \mathrm{res}_y(f,g) \in \mathbb{F}_q[x],$$

find all roots $a_1, \ldots, a_k \in \mathbb{F}_q$ of r, and then all roots b_{ij} in $\mathbb{F}_q$ of

$$h_i = \gcd(f(a_i,y), g(a_i,y), y^q - y) \in \mathbb{F}_q[y],$$

for $1 \leq i \leq k$. Then V is the set of all these (a_i, b_{ij}).

By Lemma 5.1, r can be computed with $O(M(d^3)\log d)$ operations in $\mathbb{F}_q$. Since $\deg r \leq d^2$, we can compute $a_1, \ldots, a_k$ (probabilistically) with $O((d^4 + d^2\log q)\cdot\log^2 d \operatorname{loglog} d)$ operations in $\mathbb{F}_q$ (von zur Gathen & Shoup 1992). Now fix some $i \leq k$. Then $y^q \bmod f(a_i,y)$ can be calculated with $O(M(d)\log(dq))$ operations in $\mathbb{F}_q$, and then h_i with $O\tilde{}(d)$ operations. All roots of h_i can be found with $O\tilde{}(d_i^2 + d_i\log q)$ operations in $\mathbb{F}_q$, where $d_i = \deg h_i$. Since $\sum_{1\leq i\leq k} d_i \leq mn \leq d^2$ by Bezout's Theorem, the total time is $O\tilde{}(d^4 + d^2\log q)$, and also $\#V \leq mn$.

(ii) The assumption implies that for some $e \leq d$, f has a nontrivial factorization $f_1 \cdots f_r$ with $f_1, \ldots, f_r \in \mathbb{F}_q^e[x,y]$ irreducible and $r \geq 2$. We first find an irreducible polynomial $h \in \mathbb{F}_q[x]$ of degree e, so that $= \mathbb{F}_q[x]/(h)$, and then factor f into two factors $f = h_1h_2$ over $\mathbb{F}_q^e$, with

$$O\tilde{}(d^7\log^2(q^e)(d^5 + \log q^e))$$

operations in $\mathbb{F}_q^e$, or

$$O\tilde{}(d^{11}\log^2 q(d^4 + \log q))$$

operations in $\mathbb{F}_q$. Then we apply (i) to h_1 and h_2, which yields

$$V = \{(a,b) \in^2 : h_1(a,b) = h_2(a,b) = 0\}.$$

Then $U = V \cap \mathbb{F}_q^2$, since the Galois group of $\mathbb{F}_q^e$ over $\mathbb{F}_q$ operates transitively on $\{f_1, \ldots, f_r\}$, and

$$\#U \leq \#V \leq \deg h_1 \cdot \deg h_2 \leq d^2/4,$$

by Bezout's Theorem. $\square$

THEOREM 5.3. *Let $f \in \mathbb{F}_q[x,y]$ have degree d, and $\mathcal{C} = \{f = 0\}$.*

(i) *We can compute probabilistically the number of irreducible (over $\mathbb{F}_q$) components and the number of absolutely irreducible components of $\mathcal{C}$ with $O\tilde{\ }(d^{11} \log^2 q\,(d^4 + \log q))$ operations in $\mathbb{F}_q$.*

(ii) *Suppose that we know $\#\{g = 0\}$ for each absolutely irreducible factor $g \in \mathbb{F}_q[x,y]$ of f. Then we can determine $\#\{f = 0\}$ probabilistically with the same number of operations.*

PROOF. (i) has been proven above. For (ii), we use Lemma 5.2 to calculate all U_i and u_i with $i > \sigma$, and all V_{ij} and v_i as above. □

6. The number of points in extensions

The following result shows how to compute $N_n(\mathcal{C})$ quickly for curves given by an equation of "small" degree over a "small" field.

THEOREM 6.1. *Let $\mathcal{C}$ be a smooth projective absolutely irreducible plane curve over $\mathbb{F}_q$ of degree d. Then $N_n(\mathcal{C})$ can be calculated with $O\tilde{\ }(q^{d^2})$ operations in $\mathbb{F}_q$, plus $O\tilde{\ }(n^2 \log q)$ bit operations.*

PROOF. Let $g \le d^2/2$ be the genus of $\mathcal{C}$. Each Frobenius root ϑ in (1.1) satisfies a polynomial equation

$$a = \sum_{0 \le j \le 2g} a_j x^j \in \mathbb{Q}[x] \quad \text{with } a(\vartheta) = 0.$$

Since $|\vartheta| = q^{1/2}$, we have

$$|a_j| \le \binom{2g}{j} q^{j/2}$$

for all j. The integers

$$u_k = N_k(\mathcal{C}) - q^k - 1 = \sum_{1 \le i \le g} (\vartheta_i^k + \overline{\vartheta}_i^k)$$

satisfy the linear recurrence relation

$$\sum_{0 \le j \le 2g} a_j u_{k+j} = 0$$

for all $k \ge 0$. We first calculate $v_1 = (u_1, \ldots, u_{2g})$ by the brute force method (including the points "at infinity") with $O\tilde{~}(\sum_{1 \le j \le 2g} dq^j)$ or $O\tilde{~}(dq^{2g})$ operations in $\mathbb{F}_q$. If $n \le 2g$, we are done. Otherwise, we calculate the recurrence coefficients $a_1, \ldots, a_{2g}$ with $O(M(g) \log g)$ and $u_{2g+1}, \ldots, u_n$ with $O(M(n))$ operations in $\mathbb{Q}$ (von zur Gathen & Shoup (1992), Lemma 10.1).

Since $u_k \le q^{2k}$ for all k, all rational numbers occurring in this computation have binary length $O(n \log q)$, and the last step can be done with

$$O(M(n) \log n \, M(n \log q))$$

bit operations. $\square$

This method is better than the "brute force" method whenever $d^2 \le n$. If $d^2 \le 2 \log_q n$, its running time is $O\tilde{~}(n^2 \log q)$.

Can we extract some nontrivial information about $N_n(\mathcal{C})$ from the first $k < 2g$ values $N_j(\mathcal{C})$ for $1 \le j \le k$?

Is it possible to generalize Theorem 6.1 to the case where d and q are small enough but the curve is given by an equation over $\mathbb{F}_{q^n}$ rather than over $\mathbb{F}_q$?

7. Probabilistic approximation of the size of a curve

Ma & von zur Gathen (1993) have previously given a probabilistic algorithm to estimate the size of the image of a rational function. We now give a similar algorithm for the more general problem of estimating the number $\#\mathcal{C}$ of points on a curve $\mathcal{C} = \{f = 0\}$, where $f \in \mathbb{F}_q[x, y]$ of degree d is given.

ALGORITHM 7.1. *Approximation Scheme.*
Input: $f \in \mathbb{F}_q[x, y]$ *and* $t \in \mathbb{N}$.
Output: *An estimate* E *of* $\#\{f = 0\}$.

(i) *Choose* t *random independent uniformly distributed elements* $a_1, \ldots, a_t \in \mathbb{F}_q$.

(ii) *For $j = 1, \ldots, t$, compute the number $m_j = \deg\gcd(f(x, a_j), x^q - x)$ of $u \in \mathbb{F}_q$ with $f(u, a_j) = 0$.*

(iii) *Return $E = q(m_1 + \ldots + m_t)/t$.*

THEOREM 7.2. *Let $\mathcal{C}$ be a plane curve given by $f \in \mathbb{F}_q[x, y]$ of degree d without vertical lines. Then Algorithm Approximation Scheme uses $O(tM(d)\log(dq))$ or $O\tilde{~}(td\log q)$ operations in $\mathbb{F}_q$, and for any $\delta > 0$ we have*

$$\Pr\left\{|\#\mathcal{C} - E| \leq \Big(2d(d+1)\log(2d/\delta)qt^{-1}\#\mathcal{C}\Big)^{1/2}\right\} \geq 1 - \delta.$$

PROOF. Using r_i and R_i from (3.2), we have

$$E = \frac{q}{t} \sum_{1 \leq i \leq d} i \sum_{a_j \in R_i} 1.$$

Then writing

$$c = (4\log(2d/\delta)qt^{-1})^{1/2} \quad \text{and } \epsilon_i = cr_i^{-1/2} \text{ for } 1 \leq i \leq d,$$

we get from the general result in Karp *et al.* (1989)

$$\Pr\Big\{|r_i - \frac{q}{t} \sum_{\substack{a_j \in R_i \\ 1 \leq j \leq t}} 1| \geq \epsilon_i r_i\Big\} \leq \delta/d \quad \text{for } 1 \leq i \leq d.$$

This inequality and $\#\mathcal{C} = \sum_{1 \leq i \leq d} ir_i$ imply that

$$\Pr\{|\#\mathcal{C} - E| \geq \sum_{1 \leq i \leq d} i\epsilon_i r_i\} \leq \delta.$$

Furthermore, we have

$$\begin{aligned} \sum_{1 \leq i \leq d} i\epsilon_i r_i &= c \sum_{1 \leq i \leq d} i\sqrt{r_i} \leq c\Big(\sum_{1 \leq i \leq d} i\Big)^{1/2}\Big(\sum_{1 \leq i \leq d} ir_i\Big)^{1/2} \\ &= c(d(d+1)/2)^{1/2}\#\mathcal{C}^{1/2}, \end{aligned}$$

which implies our claim. □

Using the trivial bound $\#\mathcal{C} \leq dq$, we can rewrite Theorem 8.2 in the following form.

COROLLARY 7.3. *Let $\mathcal{C}$ be a plane curve given by an equation of degree d over $\mathbb{F}_q$ and without vertical lines. Then the algorithm runs in time $O\tilde{~}(td\log q)$, and for any $\delta > 0$ we have*

$$\Pr\{|\#\mathcal{C} - E| \le dq[2(d+1)\log(2d/\delta)t^{-1}]^{1/2}\} \ge 1-\delta.$$

For fixed $\delta > 0$, the error term is $O\tilde{~}(d^{3/2}qt^{-1/2})$.

We have three methods for deriving estimates for $\#\mathcal{C}$: the Weil estimate (1.2), which requires the number of absolutely irreducible components of $\mathcal{C}$, the brute force method of Lemma 2.4 (with $m = n = 1$, $A = \mathbb{F}_q$), and the probabilistic method of Corollary 8.3. The following are the parameters for these three algorithms, for any $\alpha > 0$:

	Weil	Lemma 2.4	probabilistic
error	$d^2q^{1/2}$	0	q^α
time	0	dq	$d^4q^{2-2\alpha}$

all in the $O\tilde{~}$ sense.

Our probabilistic method is competitive when d is at least a constant power of q. The novelty in the method is that we allow a trade-off between accuracy and computing effort. As a concrete example, let us consider the case when $\mathcal{C}$ is absolutely irreducible and d is close to $q^{1/4}$. Then we have

			probabilistic	
	Weil	Lemma 2.4	general	$\alpha = 15/16$
error	q	0	q^α	$q^{15/16}$
time	0	$q^{5/4}$	$q^{3-2\alpha}$	$q^{9/8}$

again in the $O\tilde{~}$ sense. Thus the last algorithm gives a better result for approximations with error between $q^{7/8}$ and q.

For general plane curves of large degree d, when the required factorization procedure may preclude the application of Weil's estimate in practice, our probabilistic estimate is better than the brute force method whenever one can tolerate an error q^α with $q^{2\alpha-1}$ sufficiently greater than d^3.

COROLLARY 7.4. *Let $\mathcal{C}$ be an absolutely irreducible plane curve given by an equation of degree d over $\mathbb{F}_q$ and without vertical lines. Then the algorithm runs in time $O\tilde{~}(td\log q)$, and for any $\delta > 0$ we have*

$$\Pr\{|\#\mathcal{C} - E| \le [2(d+1)\log(2d/\delta)q(q+d^2q^{1/2})t^{-1}]^{1/2}\} \ge 1-\delta.$$

Now we give a similar algorithm to count the size of the projection.

ALGORITHM 7.5. *Projection Approximation.*
Input: $f \in \mathbb{F}_q[x, y]$ *and* $t \in \mathbb{N}$.
Output: *An estimate* E *of* $\#\pi\{f = 0\}$.

(i) Choose t *random independent uniformly distributed elements* $a_1, \ldots, a_t \in \mathbb{F}_q$.

(ii) For $j = 1, \ldots, t$, *set* $m_j = \min\{1, \deg \gcd(f(x, a_j), x^q - x)\}$

(iii) Return $E = q(m_1 + \ldots + m_t)/t$.

THEOREM 7.6. *Let* $\mathcal{C}$ *be a plane curve given by* $f \in \mathbb{F}_q[x, y]$ *of degree* d *and without vertical lines. Then Algorithm Projection Approximation uses* $O(tM(d)\log(dq))$ *or* $O\tilde{}(td \log q)$ *operations in* $\mathbb{F}_q$, *and for any* $\delta > 0$ *we have*

$$\Pr\left\{ |\#\pi(\mathcal{C}) - E| \le 2\Big(\log(2/\delta) q t^{-1} \#\pi(\mathcal{C})\Big)^{1/2} \right\} \ge 1 - \delta.$$

PROOF. This follows from the general result in Karp *et al.* (1989). □

Our algorithms are in the spirit of the (ϵ, δ)-approximation schemes of Karp *et al.* (1989), Grigoryev & Karpinski (1991), Karpinski & Luby (1993).

In a subsequent paper (von zur Gathen & Shparlinski 1994), we just take $a \in \mathbb{F}_q$ at random, set $a_i = a + i$ for $1 \le i \le t$, and show that these dependent random variables yield a reasonable approximation of $\#\mathcal{C}$.

8. Probabilistic approximation of the size of a curve

Ma & von zur Gathen (1993) have previously given a probabilistic algorithm to estimate the size of the image of a rational function. We now give a similar algorithm for the more general problem of estimating the number $\#\mathcal{C}$ of points on a curve $\mathcal{C} = \{f = 0\}$, where $f \in \mathbb{F}_q[x, y]$ of degree d is given.

ALGORITHM 8.1. *Approximation Scheme.*
Input: $f \in \mathbb{F}_q[x,y]$ *and* $t \in \mathbb{N}$.
Output: *An estimate* E *of* $\#\{f=0\}$.

(i) *Choose* t *random independent uniformly distributed elements* $a_1,\ldots,a_t \in \mathbb{F}_q$.

(ii) *For* $j=1,\ldots,t$, *compute the number* $m_j = \deg\gcd(f(x,a_j), x^q - x)$ *of* $u \in \mathbb{F}_q$ *with* $f(u,a_j)=0$.

(iii) *Return* $E = q(m_1+\ldots+m_t)/t$.

THEOREM 8.2. *Let* $\mathcal{C}$ *be a plane curve given by* $f \in \mathbb{F}_q[x,y]$ *of degree* d. *Then Algorithm Approximation Scheme uses* $O(tM(d)\log(dq))$ *or* $O\tilde{}(td\log q)$ *operations in* $\mathbb{F}_q$, *and for any* $\delta > 0$ *we have*

$$\Pr\left\{|\#\mathcal{C} - E| \le \left(2d(d+1)\log(2d/\delta)qt^{-1}\#\mathcal{C}\right)^{1/2}\right\} \ge 1-\delta.$$

PROOF. Using r_i and R_i from (3.2), we have

$$E = \frac{q}{t}\sum_{1\le i\le d} i \sum_{a_j\in R_i} 1.$$

Then writing

$$c = (4\log(2d/\delta)qt^{-1})^{1/2} \quad \text{and } \epsilon_i = cr_i^{-1/2} \text{ for } 1 \le i \le d,$$

we get from the general result in Karp *et al.* (1989)

$$\Pr\Big\{|r_i - \frac{q}{t}\sum_{\substack{a_j\in R_i\\ 1\le j\le t}} 1| \ge \epsilon_i r_i\Big\} \le \delta/d \quad \text{for } 1\le i\le d.$$

This inequality and $\#\mathcal{C} = \sum_{1\le i\le d} ir_i$ imply that

$$\Pr\{|\#\mathcal{C} - E| \ge \sum_{1\le i\le d} i\epsilon_i r_i\} \le \delta.$$

Furthermore, we have

$$\begin{aligned}\sum_{1\le i\le d} i\epsilon_i r_i &= c\sum_{1\le i\le d} i\sqrt{r_i} \le c\Big(\sum_{1\le i\le d} i\Big)^{1/2}\Big(\sum_{1\le i\le d} ir_i\Big)^{1/2}\\ &= c(d(d+1)/2)^{1/2}\#\mathcal{C}^{1/2},\end{aligned}$$

which implies our claim. □

Using the trivial bound $\#\mathcal{C} \le dq$, we can rewrite Theorem 8.2 in the following form.

COROLLARY 8.3. *Let $\mathcal{C}$ be a plane curve given by an equation of degree d over $\mathbb{F}_q$. Then the algorithm runs in time $O\tilde{}(td \log q)$, and for any $\delta > 0$ we have*

$$\Pr\{|\#\mathcal{C} - E| \le dq[2(d+1)\log(2d/\delta)t^{-1}]^{1/2}\} \ge 1 - \delta.$$

For fixed $\delta > 0$, the error term is $O(d^{3/2}qt^{-1/2}\log^{1/2} d)$, and this is nontrivial—i.e., is $o(q)$—for $t/d^3 \log d \longrightarrow \infty$; for $t = d^4$, the algorithm takes polynomial time $O\tilde{}(d^5 \log q)$.

We have three methods for deriving estimates for $\#\mathcal{C}$: the Weil estimate (1.2), the brute force method of Lemma 2.4, and the probabilistic method of Corollary 8.3, with the following parameters, for any $\alpha > 0$:

	Weil	Lemma 2.4	probabilistic
error	$d^2q^{1/2}$	0	q^{α}
time	0	$d^{1.7}q$	$d^4q^{2-2\alpha}$

all in the $O\tilde{}$ sense. Recall that "1.7" stands for an upper bound on $(\omega+1)/2$, where ω is a matrix multiplication exponent.

Our probabilistic method is competitive when d is at least a constant power of q. The novelty in the method is that we allow a trade-off between accuracy and computing effort. As a concrete example, let us consider the case when d is close to $q^{1/4}$. Then we have

			probabilistic	
	Weil	Lemma 2.4	general	$\alpha = 7/8$
error	q	0	q^{α}	$q^{7/8}$
time	0	$q^{1.425}$	$q^{3-2\alpha}$	$q^{5/4}$

again in the $O\tilde{}$ sense.

Our algorithm is in the spirit of the (ϵ, δ)-approximation schemes of Karp *et al.* (1989), Grigoryev & Karpinski (1991), Karpinski & Luby (1993).

In a subsequent paper (von zur Gathen & Shparlinski 1994), we just take $a \in \mathbb{F}_q$ at random, set $a_i = a + i$ for $1 \le i \le t$, and show that these dependent random variables yield a reasonable approximation of $\#\mathcal{C}$.

9. Values of the trace of a sparse polynomial

In this section, we present a method to determine the image of a polynomial in a very special case. We consider a polynomial $f \in \mathbb{F}_{q^n}[x_1, \ldots, x_m]$ that is the sum of at most t monomials (i.e., f is *t-sparse*), and let

$$\mathrm{T}(x) = \sum_{0 \le i < n} x^{q^i} \in \mathbb{F}_{q^n}[x]$$

denote the trace polynomial over $\mathbb{F}_q$. We write $q = p^r$, with p prime, and present a deterministic algorithm to determine the set

$$\{(\mathrm{T} \circ f)(u) : u \in \mathbb{F}_{q^n}^m\} \subseteq \mathbb{F}_q$$

of values which the trace of f takes, assuming a primitive root of $\mathbb{F}_{q^n}$ is given.

Our algorithm runs in time $(nt)^{O(mpr)}$ instead of the naive $O\tilde{~}(tq^{nm})$. Thus, for fields of fixed characteristic the running time can be estimated as $(ntq)^{O(m)}$. A trivial lower bound for any such algorithm is the size of the output, which may be about q.

For $f \in \mathbb{F}_q[x_1, \ldots, x_m]$ and $V \subseteq \mathbb{F}_q^m$, we write $f(V) = \{f(v) : v \in V\}$ for the set of values of f on V.

LEMMA 9.1. *Let θ be a primitive root of $\mathbb{F}_q$, and*

$$U = \{0\} \cup \{\theta^e \in \mathbb{F}_q : 0 \le e < s\} \subseteq \mathbb{F}_q. \tag{9.1}$$

Then for any s-sparse $f \in \mathbb{F}_q[x_1, \ldots, x_m]$ we have

$$f(\mathbb{F}_q^m) = \{0\} \iff f(U^m) = \{0\}.$$

PROOF. It is sufficient to prove "$\Longleftarrow$" by induction on m, starting with $m = 1$. We write $f = \sum_{1 \le i \le s} a_i x^{e_i}$, with all $a_i \in \mathbb{F}_q$ and $e_i \in \mathbb{N}$. For $d, e \in \mathbb{N}$ with $d, e \ge 1$, we have

$$(\forall a \in \mathbb{F}_q \quad a^d = a^e) \iff d \equiv e \bmod q - 1.$$

By replacing each $e_i > 0$ by its representative d_i modulo $q - 1$ with $1 \le d_i < q$ and adding coefficients of equal exponents, we obtain $g = \sum_{1 \le i \le t} b_i x^{d_i} \in \mathbb{F}_q[x]$, with $t \leq s$, $b_1, \ldots, b_t \in \mathbb{F}_q$, $0 \leq d_1 < \cdots < d_t < q$, and $f(a) = g(a)$ for all $a \in \mathbb{F}_q$. By assumption, we have

$$g(u) = 0 \text{ for all } u \in U. \tag{9.2}$$

In particular, the constant coefficient of g is zero. Since $\deg g < q$, the elements θ^{d_i} for $1 \leq i \leq t$ are pairwise distinct, and (10.2) corresponds to a Vandermonde system of linear equations. It follows that $b_i = 0$ for all i, and hence $f(a) = 0$ for all $a \in \mathbb{F}_q$.

The induction step is left to the reader. □

THEOREM 9.2. *Let $m, n, t \geq 1$, $q = p^r$, $f \in \mathbb{F}_{q^n}[x_1, \ldots, x_m]$ be t-sparse, $\theta \in \mathbb{F}_{q^n}$ primitive, $s = (nt+1)^{(p-1)r} + 1$, and $U \subseteq \mathbb{F}_{q^n}$ as in (10.1). Then*

$$(T \circ f)(\mathbb{F}_{q^n}^m) = (T \circ f)(U^m).$$

PROOF. For any $a \in \mathbb{F}_q$, there exists $u \in \mathbb{F}_{q^n}^m$ with $(\mathrm{T} \circ f)(u) = a$ if and only if the polynomial

$$g_a = (\mathrm{T} \circ f - a)^{q-1} - 1 \in \mathbb{F}_{q^n}[x_1, \ldots, x_m]$$

assumes a nonzero value on $\mathbb{F}_{q^n}^m$.

In general, the eth power of a k-sparse polynomial is k^e-sparse, and if e is a power of the field characteristic p, then this eth power is k-sparse. Furthermore, the product of a k-sparse and an ℓ-sparse polynomial is $k\ell$-sparse. It follows that $(\mathrm{T} \circ f) - a$ is $(nt+1)$-sparse. Taking into account that $q - 1 = (p-1) \cdot (p^{r-1} + \cdots + p + 1)$, we find that polynomial g_a is s-sparse. Using Lemma 10.1, we find for any $a \in \mathbb{F}_q$ that

$$a \in (\mathrm{T} \circ f)(\mathbb{F}_{q^n}^m) \iff g_a(\mathbb{F}_{q^n}^m) \neq \{0\}$$

$$\iff g_a(U^m) \neq \{0\} \iff a \in (\mathrm{T} \circ f)(U^m). \qquad \square$$

COROLLARY 9.3. *Let $m, n, t \geq 1$, $q = p^r$, $f \in \mathbb{F}_{q^n}[x_1, \ldots, x_m]$ be t-sparse. Then the image of $T \circ f : \mathbb{F}_{q^n}^m \to \mathbb{F}_q$ can be calculated with $((nt+1)^{(p-1)r} + 2)^m$ or $(nt)^{O(mpr)}$ evaluations of $T \circ f$.*

Proof. Using the algorithms of Shoup (1992) or Shparlinski (1990) (see also Shparlinski 1992a), we can construct in time $(pn)^{O(1)}$ a set $M \subseteq \mathbb{F}_q$ with cardinality $(pn)^{O(1)}$ containing a primitive root of $\mathbb{F}_q$. Setting

$$U_\mu = \{0\} \cup \{\mu^e \in \mathbb{F}_q : 0 \leq e < s\} \subseteq \mathbb{F}_q$$

for $\mu \in M$, we have from Theorem 10.2 that

$$(\mathrm{T} \circ f)(\mathbb{F}_{q^n}^m) = \bigcup_{\mu \in M} (\mathrm{T} \circ f)(U_\mu^m). \quad \square$$

The simple estimate presented here of the sparsity of a power of a polynomial may allow to improve some results of Grigoryev & Karpinski (1991) in the case of "large" fields of "small" characteristic. If $q = p^r$, the "density" of zeros of a t-sparse polynomial in t variables over $\mathbb{F}_q$ can be estimated from below as $(t+1)^{-(p-1)r^2 \log p}$, and that is $t^{O(-\log^2 q)}$ for a fixed p, say when $p = 2$. It improves their bound $(t+1)^{-(q-1)\log q}$, and a corresponding improvement of their algorithm.

Can the method of Clausen *et al.* (1991), Grigoryev *et al.* (1990) help us to improve Theorem 10.2?

Acknowledgments

Parts of this work were done during visits by the first author to Bonn, supported by the Information Technology Research Centre and the Natural Sciences and Engineering Research Council of Canada, and a sabbatical visit to the Institute for Scientific Computation at ETH Zürich, whose hospitality is gratefully acknowledged. Parts of the work of the first two authors were done while visiting the International Computer Science Institute, Berkeley, California. The second author's research was partially supported by the DFG Grant KA 673/4-1 and by the ESPRIT BR Grant 7097. We are very grateful to Kevin McCurley and to Carl Pomerance for helping us to understand their results on primes in arithmetic progressions. Many thanks also go to Henri Cohen, John Friedlander, Marc Giusti, Dima Grigoryev, Joos Heintz, Mike Luby (who supervised Andrew Quick's 1986 M. Sc. thesis), and Nicolai Vorobjov for helpful discussions.

An Extended Abstract of this paper has appeared in Proc. 25th Ann. ACM Symp. Theory of Computing, San Diego CA, 1993, 805-812.

10. Values of the trace of a sparse polynomial

In this section, we present a method to determine the image of a polynomial in a very special case. We consider a polynomial $f \in \mathbb{F}_{q^n}[x_1, \ldots, x_m]$ that is the sum of at most t monomials (i.e., f is *t-sparse*), and let

$$\mathrm{T} = \sum_{0 \le i < n} x^{q^i} \in \mathbb{F}_{q^n}[x]$$

denote the trace polynomial over $\mathbb{F}_q$. We write $q = p^r$, with p prime, and present a deterministic algorithm to determine the set

$$\{(\mathrm{T} \circ f)(u) : u \in \mathbb{F}_{q^n}^m\} \subseteq \mathbb{F}_q$$

of values which the trace of f takes, assuming a primitive root of $\mathbb{F}_{q^n}$ is given.

Our algorithm runs in time $(nt)^{O(mpr)}$ instead of the naive $O\tilde{~}(tq^{nm})$. Thus, for fields of fixed characteristic the running time can be estimated as $(ntq)^{O(m)}$. A trivial lower bound for any such algorithm is the size of the output, which may be about q.

For $f \in \mathbb{F}_q[x_1, \ldots, x_m]$ and $V \subseteq \mathbb{F}_q^m$, we write $f(V) = \{f(v) : v \in V\}$ for the set of values of f on V.

LEMMA 10.1. *Let θ be a primitive root of $\mathbb{F}_q$, and*

$$U = \{0\} \cup \{\theta^e \in \mathbb{F}_q : 0 \le e < s\} \subseteq \mathbb{F}_q. \tag{10.1}$$

Then for any s-sparse $f \in \mathbb{F}_q[x_1, \ldots, x_m]$ we have

$$f(\mathbb{F}_q^m) = \{0\} \iff f(U^m) = \{0\}.$$

PROOF. It is sufficient to prove "$\Longleftarrow$" by induction on m, starting with $m = 1$. We write $f = \sum_{1 \le i \le s} a_i x^{e_i}$, with all $a_i \in \mathbb{F}_q$ and $e_i \in \mathbb{N}$. For $d, e \in \mathbb{N}$ with $d, e \ge 1$, we have

$$(\forall a \in \mathbb{F}_q \quad a^d = a^e) \iff d \equiv e \bmod q - 1.$$

By replacing each $e_i > 0$ by its representative d_i modulo $q - 1$ with $1 \le d_i < q$ and adding coefficients of equal exponents, we obtain $g = \sum_{1 \le i \le t} b_i x^{d_i} \in \mathbb{F}_q[x]$, with $t \leq s$, $b_1, \ldots, b_t \in \mathbb{F}_q$, $0 \leq d_1 < \cdots < d_t < q$, and $f(a) = g(a)$ for all $a \in \mathbb{F}_q$. By assumption, we have

$$g(u) = 0 \text{ for all } u \in U. \tag{10.2}$$

In particular, the constant coefficient of g is zero. Since $\deg g < q$, the elements θ^{d_i} for $1 \leq i \leq t$ are pairwise distinct, and (10.2) corresponds to a Vandermonde system of linear equations. It follows that $b_i = 0$ for all i, and hence $f(a) = 0$ for all $a \in \mathbb{F}_q$.

The induction step is left to the reader. □

THEOREM 10.2. *Let $m, n, t \geq 1$, $q = p^r$, $f \in \mathbb{F}_{q^n}[x_1, \ldots, x_m]$ be t-sparse, $\theta \in \mathbb{F}_{q^n}$ primitive, $s = (nt+1)^{(p-1)r} + 1$, and $U \subseteq \mathbb{F}_{q^n}$ as in (10.1). Then*

$$(T \circ f)(\mathbb{F}_{q^n}^m) = (T \circ f)(U^m).$$

PROOF. For any $a \in \mathbb{F}_q$, there exists $u \in \mathbb{F}_{q^n}^m$ with $(\mathrm{T} \circ f)(u) = a$ if and only if the polynomial

$$g_a = (\mathrm{T} \circ f - a)^{q-1} - 1 \in \mathbb{F}_{q^n}[x_1, \ldots, x_m]$$

assumes a nonzero value on $\mathbb{F}_{q^n}^m$.

In general, the eth power of a k-sparse polynomial is k^e-sparse, and if e is a power of the field characteristic p, then this eth power is k-sparse. Furthermore, the product of a k-sparse and an ℓ-sparse polynomial is $k\ell$-sparse. It follows that $(\mathrm{T} \circ f) - a$ is $(nt+1)$-sparse. Taking into account that $q - 1 = (p-1) \cdot (p^{r-1} + \cdots + p + 1)$, we find that g_a is s-sparse. Using Lemma 10.1, we find for any $a \in \mathbb{F}_q$ that

$$a \in (\mathrm{T} \circ f)(\mathbb{F}_{q^n}^m) \iff g_a(\mathbb{F}_{q^n}^m) \neq \{0\}$$
$$\iff g_a(U^m) \neq \{0\} \iff a \in (\mathrm{T} \circ f)(U^m). \qquad \square$$

COROLLARY 10.3. *Let $m, n, t \geq 1$, $q = p^r$, $f \in \mathbb{F}_{q^n}[x_1, \ldots, x_m]$ be t-sparse, and suppose that a primitive element of $\mathbb{F}_{q^n}$ is given. Then the image of $T \circ f : \mathbb{F}_{q^n}^m \to \mathbb{F}_q$ can be calculated with $((nt+1)^{(p-1)r} + 2)^m$ or $(nt)^{O(mpr)}$ evaluations of $T \circ f$.*

PROOF. By Theorem 10.2, it is sufficient to evaluate $\mathrm{T} \circ f$ at each $u \in U^m$. □

The simple estimate presented here of the sparsity of a power of a polynomial may allow to improve some results of Grigoryev & Karpinski (1991) in the case of "large" fields of "small" characteristic.

Can the method of Clausen *et al.* (1991), Grigoryev *et al.* (1990) help us to improve Theorem 10.2?

Acknowledgments

Parts of this work were done during visits by the first author to Bonn, supported by the Information Technology Research Centre and the Natural Sciences and Engineering Research Council of Canada, and a sabbatical visit to the Institute for Scientific Computation at ETH Zürich, whose hospitality is gratefully acknowledged. Parts of the work of the first two authors were done while visiting the International Computer Science Institute, Berkeley, California. The second author's research was partially supported by the DFG Grant KA 673/4-1 and by the ESPRIT BR Grant 7097. We are very grateful to Kevin McCurley and to Carl Pomerance for helping us to understand their results on primes in arithmetic progressions. Many thanks also go to Henri Cohen, John Friedlander, Marc Giusti, Dima Grigoryev, Joos Heintz, Mike Luby, and Nicolai Vorobjov for helpful discussions.

An Extended Abstract of this paper has appeared in Proc. 25th Ann. ACM Symp. Theory of Computing, San Diego CA, 1993, 805-812.

References

W. R. ALFORD, A. GRANVILLE, AND C. POMERANCE, There are infinitely many Carmichael numbers. Mathematical preprint series, University of Georgia, 1993.

E. BACH, Weil bounds for singular curves. Preprint, 1993.

J. BIRCH AND H. P. F. SWINNERTON-DYER, Note on a problem of Chowla. *Acta Arith.* **5** (1959), 417–423.

E. BOMBIERI, On exponential sums in finite fields. *Amer. J. Math.* **88** (1966), 71–105.

A. L. Chistov and D. Yu. Grigoryev, Polynomial-time factoring of the multivariable polynomials over a global field. LOMI preprint E-5-82, 1982.

M. Clausen, A. Dress, J. Grabmeier, and M. Karpinski, On zero testing and interpolation of k-sparse multivariate polynomials over finite fields. *Theor. Computer Science* **84** (1991), 151–164.

D. Coppersmith and S. Winograd, Matrix multiplication via arithmetic progressions. *J. Symb. Comp.* **9** (1990), 251–280.

H. Davenport, *Multiplicative Number Theory.* Springer-Verlag, Second edition, 1980.

M. D. Fried and M. Jarden, *Field Arithmetic.* Springer-Verlag, 1986.

M. R. Garey and D. S. Johnson, *Computers and intractability: A guide to the theory of NP-Completeness.* W. H. Freeman, San Francisco CA, 1979.

J. von zur Gathen, Tests for permutation polynomials. *SIAM J. Comput.* **20**(3) (1991), 591–602.

J. von zur Gathen and E. Kaltofen, Factorization of multivariate polynomials over finite fields. *Math. Comp.* **45** (1985), 251–261.

J. von zur Gathen and V. Shoup, Computing Frobenius maps and factoring polynomials. *Comput complexity* **2** (1992), 187–224.

J. von zur Gathen and I. E. Shparlinski, Components and projections of curves over finite fields. Manuscript, 1994.

D. Yu. Grigoryev and M. Karpinski, An approximation algorithm for the number of zeros of arbitrary polynomials over $GF(q)$. In *Proc. 20th IEEE Symp. Foundations of Computer Science*, 1991, 662–669.

D. Yu. Grigoryev, M. Karpinski, and M. F. Singer, Fast parallel algorithms for sparse multivariate polynomial interpolation over finite fields. *SIAM J. Comp.* **19** (1990), 1059–1063.

R. Hartshorne, *Algebraic Geometry.* Springer-Verlag, 1977.

M.-D. Huang and D. Ierardi, Counting rational points on curves over finite fields. In *Proc. 34th Ann. IEEE Sympos. Foundations of Computer Science*, Palo Alto CA, 1993, 616–625.

E. Kaltofen, Fast parallel absolute irreducibility testing. *J. Symb. Computation* **1** (1985), 57–67.

E. Kaltofen, Greatest common divisors of polynomials given by straight-line programs. *J. Assoc. Comput. Mach.* **35** (1988), 231–264.

R. M. Karp, M. Luby, and N. Madras, Monte-Carlo approximation algorithms for enumeration problems. *J. Algorithms* **10**(3) (1989), 429–448.

M. Karpinski and M. Luby, Approximating the number of solutions of a $GF[2]$ polynomial. *J. Algorithms* **14** (1993), 280–287.

A. K. Lenstra, Factoring multivariate polynomials over finite fields. *J. Comput. System Sci.* **30** (1985), 235–248.

R. Lidl and H. Niederreiter, *Finite Fields*, vol. 20 of *Encyclopedia of Mathematics and its Applications.* Addison-Wesley, Reading MA, 1983.

K. Ma and J. von zur Gathen, Counting value sets of functions and testing permutation functions. In *Abstracts of Int. Conf. Number Theoretic and Algebraic Methods in Computer Science*, Moscow, 1993, 62–65.

K. S. McCurley, Explicit zero-free regions for Dirichlet L-functions. *J. Number Theory* **19** (1984), 7–32.

R. Pila, Frobenius maps of Abelian varieties and finding roots of unity in finite fields. *Math. Comp.* **55** (1990), 745–763.

D. A. Plaisted, Sparse complex polynomials and polynomial reducibility. *J. Comp. and System Sciences* **14** (1977), 210–221.

D. A. Plaisted, New NP-hard and NP-complete polynomial and integer divisibility problems. *Theor. Computer Science* **31** (1984), 125–138.

A. Quick, Some GCD and divisibility problems for sparse polynomials. Technical Report 191/86, Department of Computer Science, University of Toronto, 1986.

J. B. Rosser and L. Schoenfeld, Approximate formulas for some functions of prime numbers. *Ill. J. Math.* **6** (1962), 64–94.

R. J. Schoof, Elliptic curves over finite fields and the computation of square roots mod p. *Math. Comp.* **44**(170) (1985), 483–494.

J. T. Schwartz, Fast probabilistic algorithms for verification of polynomial identities. *J. Assoc. Computing Machinery* **27** (1980), 701–717.

J.-P. Serre, Sur le nombre des points rationels d'une courbe algébrique sur un corps fini. *C.R. Acad. Sci. Paris, Ser. 1* **296** (1983a), 397–402.

J.-P. Serre, Nombre de points des courbes algébriques sur $\mathbb{F}_q$. *Sémin. de Théorie des Nombres de Bordeaux* (1983b), 1–8.

I. R. Shafarevich, *Basic algebraic geometry*. Grundlehren Band 213. Springer Verlag, 1974.

V. Shoup, Searching for primitive roots in finite fields. *Math. Comp.* **58**(197) (1992), 369–380.

I. E. Shparlinski, On primitive elements in finite fields and on elliptic curves. *Mat. Sbornik* **181**(9) (1990), 1196–1206. Math. USSR Sbornik **71** (1991), 41-50.

I. E. Shparlinski, *Computational and algorithmic problems in finite fields*, vol. 88 of *Mathematics and its applications*. Kluwer Academic Publishers, 1992a.

I. E. Shparlinski, A deterministic test for permutation polynomials. *Comput complexity* **2** (1992b), 129–132.

I. E. Shparlinski and A. N. Skorobogatov, Exponential sums and rational points on complete intersections. *Mathematika* **37** (1990), 201–208.

I. M. Vinogradov, Основы теории чисел *(Elements of number theory)*. Nauka, Moscow, 2nd edition, 1981.

S. G. Vladuts and V. G. Drinfeld, Number of points on an algebraic curve. *Analis i Prilogenija* **17**(1) (1983), 68–69. In Russian.