Finite Fields and Their Applications 16 (2010) 155–162

Contents lists available at ScienceDirect

Finite Fields and Their Applications

www.elsevier.com/locate/ffa

Fast arithmetic in unramified p-adic fields

Hendrik Hubrechts

Katholieke Universiteit Leuven, Department of Mathematics, Celestijnenlaan 200B, Leuven, Belgium

ARTICLE INFO

Article history:
Received 30 June 2009
Revised 18 December 2009
Available online 6 January 2010
Communicated by Igor Shparlinski

Keywords:
p-Adic fields
Fast arithmetic
Point counting

ABSTRACT

Let p be prime and $\mathbb{Z}_{p^n}$ a degree n unramified extension of the ring of p-adic integers $\mathbb{Z}_p$. In this paper we give an overview of some very fast deterministic algorithms for common operations in $\mathbb{Z}_{p^n}$ modulo p^N. Combining existing methods with recent work of Kedlaya and Umans about modular composition of polynomials, we achieve quasi-linear time algorithms in the parameters n and N, and quasi-linear or quasi-quadratic time in $\log p$, for most basic operations on these fields, including Galois conjugation, Teichmüller lifting and computing minimal polynomials.

1. Introduction

An important topic in computational number theory and algebraic geometry in recent years is the design of point counting algorithms. More specifically, let $\mathbb{F}_{p^n}$ be the field with p^n elements, p prime, and V a variety over $\mathbb{F}_{p^n}$, then the question is how to count the number of $\mathbb{F}_{p^n}$-rational points on V in an efficient manner. In 1985 Schoof gave a first general algorithm for elliptic curves (the SEA-algorithm [32]). Afterwards cryptography and other applications stimulated further research in this direction. In 1999, Satoh [29] proposed the first p-adic algorithm for elliptic curves (for small $p \geqslant 5$), which was based on the canonical lift. This work was extended and improved by many different authors, e.g. [6,35,38,26,30,19,25,9]. Kedlaya [18] initiated an approach based on Monsky–Washnitzer (or rigid) cohomology that works for hyperelliptic curves, and also this line of research turned out to be fruitful, see [5,4,1,10,7,8,21,22,14,11]. Lauder and Wan [24] and Lauder [23] conceived algorithms for computing zeta functions of hypersurfaces of arbitrary dimension based on Dwork cohomology.

Except for the SEA-algorithm all these methods work by lifting the variety along with certain morphisms to a field of characteristic zero, more precisely to finite extensions of $\mathbb{Q}_p$ with residue field $\mathbb{F}_{p^n}$. In this p-adic field one then typically needs Frobenius computations or more generally some calculations concerning an action of the Galois group. In particular, the chain of improvements on

E-mail address: Hendrik.Hubrechts@wis.kuleuven.be.

1071-5797/$ – see front matter
doi:10.1016/j.ffa.2009.12.004

Satoh's algorithm consists mainly of ever improving p-adic arithmetic. For instance, one crucial idea made Harley's result [9] (see also [37, Section 3.10]) possible, which is the fastest known algorithm for computing the number of points on an elliptic curve over $\mathbb{F}_{p^n}$ for small p. In order to achieve this result, Harley gave an efficient algorithm for computing a Teichmüller modulus modulo p^N (see Section 2.5 below). We present an algorithm that has a time complexity of $\mathcal{O}((Nn\log^2 p)^{1+\epsilon})$, whereas Harley's algorithm works also in $\mathcal{O}((Nn)^{1+\epsilon})$ for fixed p, but is at least exponential in p. For curves of higher genus often the norm of not just one element (for which Harley used a special trick), but of a matrix of p-adic numbers has to be computed. Using our results this can be done in essentially linear time apart from an extra factor $\log p$. This leads in our papers [2] (with Castryck and Vercauteren), [13] and [12] to the following substantial improvement: for fixed p and genus the zeta function of a $C_{a,b}$-curve over $\mathbb{F}_{p^n}$ lying in a one parameter family defined over the prime field $\mathbb{F}_p$ can be computed in time $\mathcal{O}(n^{2+\epsilon})$, instead of $\mathcal{O}(n^{2.667})$ as mentioned in those papers.

Other applications of fast p-adic arithmetic include computing zeta functions of certain formal groups over finite fields [31], bounding Picard numbers of surfaces [17] and computing Coleman integrals [16].

The basic result required for this paper is the following. In a recent article, Kedlaya and Umans were able to give an essentially linear time and deterministic algorithm for the problem of modular composition:

Theorem 1. *(See Kedlaya and Umans [15, Theorem 7.1 with parameters $m = 1$, $N = d$].) Let R be a finite ring of cardinality q given as $(\mathbb{Z}/r\mathbb{Z})[Z]/(E(Z))$ for some monic polynomial $E(Z)$. For every constant $\delta > 0$ there is an algorithm that does the following. Given polynomials $f(X)$, $g(X)$ and $h(X)$ over R of degree at most d, such that h has a unit as leading coefficient and that we have access to $d^{1+\delta}$ distinct elements of R whose differences are units in R; then it can compute $f(g(X)) \bmod h(X)$ in at most $d^{1+\delta}\log^{1+o(1)} q$ bit operations.*

Although the main idea of this paper is merely to combine existing algorithms with the above theorem, most results are new. A central source for classical fast algorithms is the book [39], and for more specific p-adic methods we refer to Chapter 12 of [3].

Let $\mathbb{Z}_{p^n}$ be the valuation ring of the unramified extension field $\mathbb{Q}_{p^n}$ of degree n of $\mathbb{Q}_p$. All results below for computing in $\mathbb{Z}_{p^n}$ with precision p^N are quasi-linear except for some extra factor $\log p$ arising from computing a p-th power in the finite field $\mathbb{F}_{p^n}$. For example, computing a Teichmüller lift requires time $\mathcal{O}((n\log p\log p^N)^{1+\epsilon})$, whereas the most general algorithm in [3] requires time $\mathcal{O}((n^2\log p\log p^N)^{1+\epsilon})$. We note that any improvement in computing x^p in $\mathbb{F}_{p^n} \cong \mathbb{F}_p[x]/\bar{\varphi}(x)$ over the complexity $\mathcal{O}((n\log^2 p)^{1+\epsilon})$ of repeated squaring would yield a similar improvement for most of our results. Moreover, it is easy to verify that the memory requirements for all results in this paper are essentially linear, and that all algorithms are deterministic.

The structure of the sequel of the paper is quite straightforward: first we prove a corollary to Theorem 1 that allows fast modular composition over p-adic fields. Then in separate subsections we give various results concerning Newton iteration, Galois conjugation, equations involving the Frobenius automorphism, Teichmüller lift, minimal polynomial, trace, norm and Teichmüller modulus.

We note that the use of the exponent $1+\epsilon$ in all our complexity estimates means that for every $\epsilon > 0$ an algorithm exists with this estimate. For most results only logarithmic factors are needed (e.g. $\mathcal{O}(n\log n)$ instead of $\mathcal{O}(n^{1+\epsilon})$), but we adopt a more uniform formulation.

2. Fast arithmetic

We choose for once and for all a prime number p, an extension degree $n \geqslant 1$, a p-adic precision $N \geqslant 1$ and we define $q := p^N$. Recall that there exists up to isomorphism a unique unramified degree n field extension $\mathbb{Q}_{p^n}$ of the p-adic field $\mathbb{Q}_p$, see [20, Section III.3]. We work in the valuation ring $\mathbb{Z}_{p^n}$ of $\mathbb{Q}_{p^n}$, with precision p^N. We may assume that this ring is represented as $\mathbb{Z}_p[x]/\varphi(x)$ for some monic inert (i.e. irreducible modulo p) polynomial $\varphi(x) \in \mathbb{Z}_p[x]$ of degree n and precision p^N. From now on the notation $\mathbb{Z}_{p^n} \bmod p^N$ will be used for this setting (including the implicit polynomial $\varphi(x)$).

It is not in the scope of this text to discuss how to find a (large) prime p and some inert polynomial $\varphi(x)$ of given degree n. However, we note that for finding $\varphi(x)$ it suffices to compute an irreducible polynomial $\bar{\varphi}(x)$ of degree n over $\mathbb{F}_p$, which is an extensively studied problem [34].

It is well known (see e.g. [39]) that basic operations like addition, multiplication and division by units in $\mathbb{Z}_{p^n} \bmod p^N$ can be performed deterministically in time $\mathcal{O}((n \log p^N)^{1+\epsilon})$. In this section we will show that many more operations are possible within similar time constraints, if we use the aforementioned result of Kedlaya and Umans. For our purposes Theorem 1 is not immediately applicable, hence we give a reformulation.

Theorem 2. *Let $f(x)$, $g(x)$ and $h(x)$ be polynomials of degree at most n over $\mathbb{Z}_p[x] \bmod p^N$, with $h(x)$ monic. Recall that $q := p^N$. Then we can compute $f(g(x)) \bmod h(x)$ in time $\mathcal{O}((n \log q)^{1+\epsilon})$.*

Proof. If p is large enough in comparison to n, say $p \geqslant n^2$, we can use Theorem 1 directly because $\mathbb{Z}_p$ contains enough (readily available) elements whose differences are units. Suppose hence $p < n^2$. Shoup gave in [33] a deterministic algorithm that computes an irreducible polynomial $\bar{E}(Y)$ of degree a over $\mathbb{F}_p$ in at most $(\sqrt{p}\, a^4)^{1+\epsilon}$ operations in $\mathbb{F}_p$. It now suffices to take $a := \lceil \log_p n^2 \rceil$ and to note that $\sqrt{p}$ is dominated by n. Let $E(Y)$ be a monic lift of $\bar{E}(Y)$, then the ring $\mathbb{Z}_p[Y]/E(Y)$ has at least n^2 elements whose differences are units and we conclude the proof with Theorem 1. □

We note that by using q instead of p^N in the complexity estimate of the theorem, the result is more general. In particular, the complexity bound holds for fixed p (in which case N and n have to be large enough), and for fixed N (for p and n large enough). Moreover, although in some complexity results below expressions like $\log^2 p + \log q$ will appear, in all these cases the estimates hold also for fixed p or N.

2.1. Root finding (Newton iteration)

Let $f(Y)$ be a polynomial over $\mathbb{Z}_p \bmod p^N$ or over $\mathbb{Z}_{p^n} \bmod p^N$. In this subsection we want to compute an approximation of a root of $f(Y)$. In order to be able to use Newton iteration, we have to require that we already know a single root of the polynomial modulo p. We remark that Proposition 2 is added for completeness, it is not new and its proof does not require Theorem 1.

Proposition 1. *Let $f(Y)$ be a polynomial over $\mathbb{Z}_p \bmod p^N$ of degree m, and let $y_0 \in \mathbb{Z}_{p^n} \bmod p^N$ such that $f(y_0) \equiv 0 \bmod p$ and $\frac{df}{dY}(y_0) \not\equiv 0 \bmod p$. Then we can compute $y \in \mathbb{Z}_{p^n} \bmod p^N$ such that $y \equiv y_0 \bmod p$ and $f(y) \equiv 0 \bmod p^N$ in time $\mathcal{O}(((n+m) \log q)^{1+\epsilon})$.*

Proposition 2. *Let $f(Y)$ be a polynomial over $\mathbb{Z}_{p^n} \bmod p^N$ of degree m, and let $y_0 \in \mathbb{Z}_{p^n} \bmod p^N$ such that $f(y_0) \equiv 0 \bmod p$ and $\frac{df}{dY}(y_0) \not\equiv 0 \bmod p$. Then we can compute $y \in \mathbb{Z}_{p^n} \bmod p^N$ such that $y \equiv y_0 \bmod p$ and $f(y) \equiv 0 \bmod p^N$ in time $\mathcal{O}((nm \log q)^{1+\epsilon})$.*

Proof. Suppose that $y_i \in \mathbb{Z}_{p^n} \bmod p^N$ is such that $f(y_i) \equiv 0 \bmod p^{2^i}$ and $y_i \equiv y_0 \bmod p$. Then $\frac{df}{dY}(y_i)$ is invertible and we can define

$$y_{i+1} := y_i + f(y_i) \cdot \left(\frac{df}{dY}(y_i) \right)^{-1} \bmod p^{2^{i+1}}. \tag{1}$$

By using $f(Y) = f(y_i) + (Y - y_i)\frac{df}{dY}(y_i) + (Y - y_i)^2 \cdots$, see e.g. Lemma 9.20 in [39], one verifies trivially that $f(y_{i+1}) \equiv 0 \bmod p^{2^{i+1}}$ and $y_{i+1} \equiv y_0 \bmod p$. Note that this procedure is just the p-adic analogue of classical Newton iteration. Clearly we have to apply (1) at most $\lceil \log_2(N) \rceil$ times in order to find $y \bmod p^N$ as required in the propositions. In the situation of Proposition 1 each application of (1) takes time no more than $\mathcal{O}((\max(n, m) \cdot \log q)^{1+\epsilon})$ by Theorem 2 with $h(x) = \varphi(x)$, and for Proposition 2 we have $\mathcal{O}((nm \log q)^{1+\epsilon})$ for each step by Horner's rule. □

2.2. Galois conjugates

We denote with σ the p-th power Frobenius automorphism on $\mathbb{Q}_{p^n}$, which is the unique field automorphism that on $\mathbb{Z}_{p^n}/p\mathbb{Z}_{p^n} \cong \mathbb{F}_{p^n}$ reduces to $\bar{\sigma} : x \mapsto x^p$. Note that σ^n is the identity map and that σ generates the Galois group of $\mathbb{Q}_{p^n}$ over $\mathbb{Q}_p$.

Proposition 3. *Let $\alpha \in \mathbb{Z}_{p^n} \bmod p^N$ and $0 < k < n$ be an integer. We can compute $\sigma^k(\alpha)$ in time $\mathcal{O}((n\log^2 p + n\log q)^{1+\epsilon})$.*

Proof. Let $\mathbb{F}_{p^n} \cong \mathbb{F}_p[\bar{x}]/\bar{\varphi}(\bar{x})$ be the 'reduction modulo p' of $\mathbb{Z}_{p^n} \cong \mathbb{Z}_p[x]/\varphi(x)$, with $\bar{\sigma}$ as p-th power Frobenius on it. Clearly we can compute $\bar{\sigma}(\bar{x}) = \bar{x}^p$ in $\mathbb{F}_{p^n}$ in time $\mathcal{O}((n\log^2 p)^{1+\epsilon})$. In order to compute $\bar{\sigma}^k(\bar{x}) = \bar{x}^{p^k}$ we use the following lemma.

Lemma 1. *Given the polynomials $A(\bar{x}) := (\bar{x}^{p^a} \bmod \bar{\varphi}(\bar{x}))$ and $B(\bar{x}) := (\bar{x}^{p^b} \bmod \bar{\varphi}(\bar{x}))$ for some integers $a, b \geqslant 1$, we have that $A(B(\bar{x})) \equiv \bar{x}^{p^{a+b}} \bmod \bar{\varphi}(\bar{x})$, and this composition can be computed in time $\mathcal{O}((n\log p)^{1+\epsilon})$.*

Proof. It is easy to verify that $A(B(\bar{x})) \bmod \bar{\varphi}(\bar{x}) = \bar{x}^{p^{a+b}} \bmod \bar{\varphi}(\bar{x})$, using the fact that $B(\bar{x})$ is a root of $\bar{\varphi}(\bar{x})$. Now Theorem 2 (for $N = 1$) completes the proof of the lemma. □

Proof of Proposition 3 (*continued*)**.** The idea to compute $\bar{\sigma}^k(\bar{x})$ is to use the binary representation of k combined with the lemma. The general algorithm is similar to the classical repeated squaring technique (Algorithm 4.8 in [39]), we explain here only the easier case where $k = 2^m$ for an integer $m \geqslant 1$. The procedure is quite obvious: compute recursively $A_i(\bar{x}) = A_{i-1}(A_{i-1}(\bar{x})) \bmod \varphi((\bar{x}))$ with $A_0(\bar{x}) = \sigma(\bar{x})$. Lemma 1 yields $A_1(\bar{x}) = \bar{x}^{p^2} \bmod \bar{\varphi}(\bar{x})$, $A_2(\bar{x}) = \bar{x}^{p^4} \bmod \bar{\varphi}(\bar{x}), \ldots, A_m(\bar{x}) = \bar{x}^{p^{2^m}} \bmod \bar{\varphi}(\bar{x})$. Only $m = \log_2 k \leqslant \log n$ steps are required, hence if we know $\bar{\sigma}(\bar{x})$, we can compute $\bar{\sigma}^k(\bar{x})$ in time $\mathcal{O}(\log k(n\log p)^{1+\epsilon}) = \mathcal{O}((n\log p)^{1+\epsilon})$.

Because $\sigma^k(x)$ is a root of $\varphi(X)$ and $\bar{\varphi}(X)$ is squarefree, we can now apply Proposition 1 in order to lift $\bar{\sigma}^k(\bar{x})$ to $\sigma^k(x)$ modulo p^N in time $\mathcal{O}((n\log q)^{1+\epsilon})$. For $\alpha(x) \in \mathbb{Z}_p[x]/\varphi(x)$ we have $\sigma^k(\alpha(x)) = \alpha(\sigma^k(x)) \bmod \varphi(x)$, and hence Theorem 2 allows us to compute this last expression with precision $q = p^N$ in time $\mathcal{O}((n\log q)^{1+\epsilon})$, thereby proving the proposition. □

Corollary 1. *Let $\alpha \in \mathbb{F}_{p^n}$, $\bar{\sigma}$ be the Frobenius automorphism and $0 < k < n$. Then we can compute $\bar{\sigma}^k(\alpha)$ in time $\mathcal{O}((n\log^2 p)^{1+\epsilon})$.*

Proof. With $\mathbb{F}_{p^n}$ given as $\mathbb{F}_p[\bar{x}]/\bar{\varphi}(\bar{x})$, we have shown above that $\bar{\sigma}^k(\bar{x})$ can be computed in time $\mathcal{O}((n\log^2 p)^{1+\epsilon})$. Now writing α as $\alpha(\bar{x})$ gives $\bar{\sigma}^k(\alpha(\bar{x})) = \alpha(\bar{\sigma}^k(\bar{x}))$, hence Theorem 2 gives the corollary. □

2.3. Equations with Frobenius

In this section we rephrase some results from [3] using faster Frobenius computations.

Proposition 4. *Let $\alpha, \beta, \gamma \in \mathbb{Z}_{p^n} \bmod p^N$ with $\beta \equiv 0 \bmod p$ and $\alpha \not\equiv 0 \bmod p$. We can compute the (unique) solution y in $\mathbb{Z}_{p^n} \bmod p^N$ of $\alpha\sigma(Y) + \beta Y + \gamma = 0$ in time $\mathcal{O}((n\log^2 p + n\log q)^{1+\epsilon})$.*

Proof. The equation of the proposition is equivalent to $\sigma(Y) = a_1 Y + b_1$, where $a_1 = -\beta/\alpha$ and $b_1 = -\gamma/\alpha$. Applying σ to this equation gives

$$\sigma^2(Y) = \sigma(a_1 Y + b_1) = \sigma(a_1)a_1 Y + \sigma(a_1)b_1 + \sigma(b_1).$$

More generally the recurrence relations (for $i \geqslant 1$)

$$a_{i+1} := \sigma(a_i)a_1, \qquad b_{i+1} := \sigma(a_i)b_1 + \sigma(b_i) \tag{2}$$

imply that $\sigma^i(Y) = a_i Y + b_i$. Let $y \in \mathbb{Z}_{p^n}$ be a solution of the original equation, then we find $\sigma^n(y) = a_n y + b_n$. Now $\sigma^n(y) = y$, hence equivalently $y = b_n/(1-a_n) \in \mathbb{Z}_{p^n} \bmod p^N$. Indeed, from (2) and the fact that $a_1 \equiv \beta \equiv 0 \bmod p$ we see that $1 - a_n$ is a unit. Lercier and Lubicz gave in [25] an efficient divide and conquer algorithm for computing a_n and b_n and hence y (see also Section 12.6.1 of [3]). The dominating cost of this algorithm is the computation of σ^k for $\mathcal{O}(\log_2 n)$ different values of k, and Proposition 3 gives then that we can compute y in time $\mathcal{O}(\log n(n\log^2 p + n\log q)^{1+\epsilon})$. □

Proposition 5. *Let $\phi(Y, Z)$ be a polynomial over $\mathbb{Z}_{p^n} \bmod p^N$ for which the evaluation of ϕ, $\partial\phi/\partial Y$ and $\partial\phi/\partial Z$ in any $(\alpha, \beta) \in (\mathbb{Z}_{p^n} \bmod p^N)^2$ requires at most ψ arithmetic operations in $\mathbb{Z}_{p^n} \bmod p^N$. Suppose we have $y_0 \in \mathbb{Z}_{p^n} \bmod p^N$ such that $\phi(y_0, \sigma(y_0)) \equiv 0 \bmod p^{2k+1}$ with $k := \operatorname{ord}_p(\frac{\partial\phi}{\partial Z}(y_0, \sigma(y_0))) < N$. Then we can compute $y \in \mathbb{Z}_{p^n} \bmod p^{N+k}$ such that $\phi(y, \sigma(y)) \equiv 0 \bmod p^{N+k}$ and $y \equiv y_0 \bmod p^{k+1}$ in time $\mathcal{O}((n\log^2 p + \psi n\log q)^{1+\epsilon})$.*

Proof. Algorithm 12.23 from [3] gives a multivariate generalization of the Newton iteration used in the proof of Proposition 2. This algorithm reduces solving the equation $\phi(Y, \sigma(Y)) = 0$ to solving equations of the type considered in Proposition 4. We refer to [3] for details of the algorithm. Except for $\mathcal{O}(\log N)$ times an evaluation of ϕ, $\partial\phi/\partial Y$ and $\partial\phi/\partial Z$, its complexity is the same as the one given in Proposition 4 above. Hence the total complexity is bounded by $\mathcal{O}(\psi \log N(n\log q)^{1+\epsilon} + (n\log^2 p + n\log q)^{1+\epsilon})$. □

2.4. Teichmüller lift

In the point counting algorithms mentioned in the beginning that are based on Monsky–Washnitzer cohomology and deformation (e.g. [21,13]), it is necessary to apply σ to a formal parameter Γ in such a way that σ behaves like the Frobenius automorphism after the substitution of a well-chosen p-adic number γ for Γ. This can be achieved by taking $\sigma(\Gamma) := \Gamma^p$ and for γ a *Teichmüller lift*. Let $\bar{\beta} \in \mathbb{F}_{p^n} \cong \mathbb{Z}_{p^n}/p\mathbb{Z}_{p^n}$, then the Teichmüller lift β of $\bar{\beta}$ in $\mathbb{Z}_{p^n}$ is by definition the unique root of unity congruent to $\bar{\beta}$ modulo p. It is easy to verify that for such β the relation $\sigma(\beta) = \beta^p$ holds.

Proposition 6. *Given $\alpha \in \mathbb{Z}_{p^n} \bmod p$, we can compute the Teichmüller lift of $(\alpha \bmod p)$ in time $\mathcal{O}((n\log p\log q)^{1+\epsilon})$.*

Proof. As pointed out in Section 12.8.1 of [3], we can use Proposition 5 for the polynomial $\phi(Y, Z) = Y^p - Z$ with $x_0 = \alpha$ and $k = 0$. Indeed, the unique root of $\phi(Y, \sigma(Y)) = Y^p - \sigma(Y) = 0$ congruent to α modulo p is the Teichmüller lift of α. Evaluating ϕ, $\partial\phi/\partial Y$ and $\partial\phi/\partial Z$ requires $\mathcal{O}(\log p)$ elementary operations in $\mathbb{Z}_{p^n} \bmod p^N$ and we find the proposition. □

2.5. Minimal polynomial, trace and norm

As $\mathbb{Q}_{p^n}$ has degree n over $\mathbb{Q}_p$, every element α of $\mathbb{Q}_{p^n}$ is the root of a unique irreducible monic polynomial over $\mathbb{Q}_p$ of degree a divisor of n. When $\alpha \in \mathbb{Z}_{p^n}$, this minimal polynomial $f(X)$ is also defined over $\mathbb{Z}_{p^n}$. Indeed, with $A := \{\sigma^k(\alpha) \mid k = 0, \ldots, n-1\}$ as the set of conjugates of α, we have

$$f(X) = \prod_{\beta \in A} (X - \beta).$$

Proposition 7. *Let $\alpha \in \mathbb{Z}_{p^n} \bmod p^N$ and suppose that $\alpha \bmod p$ has degree n over $\mathbb{F}_p$. We can compute the minimal polynomial modulo p^N of α over $\mathbb{Z}_p$ in time $\mathcal{O}((n\log q)^{1+\epsilon})$.*

Proof. We follow an idea of [28] and [36] as explained in Section 3 of [34]. Define the linear operator $P : \mathbb{Z}_p[x]/\varphi(x) \to \mathbb{Z}_p$ by $P(1) := 1$ and $P(x) = P(x^2) = \cdots = P(x^{n-1}) = 0$. We can compute—using the fast modular power projection of Theorem 7.7 in [15]—the sequence $P(1), P(\alpha), \ldots, P(\alpha^{2n-1})$ with precision p^N in essentially linear time $\mathcal{O}((n\log q)^{1+\epsilon})$. The monic minimal polynomial $c(X)$ of $\{P(\alpha^i)\}_{i\geqslant 0}$ is a divisor of the minimal polynomial of α. As this last one is irreducible modulo p^N (even when reduced modulo p), they are equal. Step 2 of Shoup's algorithm refers to the fact that one can obtain the minimal polynomial of $\{P(\alpha^i)\}$ from the (fast) extended Euclidean algorithm for

$$g(X) = \sum_{i=0}^{2n-1} P\big(\alpha^i\big)X^{2n-1-i} \quad \text{and} \quad f(X) = X^{2n}.$$

Indeed, knowing a Euclidean expansion $c(X)g(X) + q(X)f(X) = r(X)$ for some remainder $r(X)$ of degree at most $n-1$ and with $c(X)$ of minimal degree, implies that $c(X)$ is the minimal polynomial of $\{P(\alpha^i)\}$. □

It is interesting to discuss why we have to require that the reduction of α modulo p has degree n over $\mathbb{F}_p$. It is not hard to see that the concept of minimal polynomial cannot be defined in a satisfying way in general when working with finite precision p-adic fields. For example, take $p = n = N = 2$. The minimal polynomial of $1 + 2x$ in $\mathbb{Z}_{2^2} \cong \mathbb{Z}_2[x]/(x^2 - x - 1)$ is then $X^2 - 4X - 1$, which modulo p^2 reduces to $X^2 - 1$. This polynomial is not even irreducible, and in the above algorithm the sequence $P(\alpha^i) \bmod p^N$ would be identically 1 and have minimal polynomial $X - 1$.

The (absolute) trace $\mathrm{Tr}(\alpha)$ and norm $\mathrm{N}(\alpha)$ of $\alpha \in \mathbb{Q}_{p^n}$ are defined as respectively $\sum_{k=0}^{n-1}\sigma^k(\alpha)$ and $\prod_{k=0}^{n-1}\sigma^k(\alpha)$. Suppose for a moment that $n = 2^m$, then by using the recursion relation $\alpha_0 := \sigma(\alpha)$, $\alpha_i := \sigma^{2^{i-1}}(\alpha_{i-1}) + \alpha_{i-1}$—so that $\alpha_m = \mathrm{Tr}(\alpha)$—it is easy to see that the trace (and similarly the norm) of α can be computed in time $\mathcal{O}((n\log^2 p + n\log q)^{1+\epsilon})$. It is however an easy corollary of Proposition 7 that this can be done faster.

Corollary 2. *Let $\alpha \in \mathbb{Z}_{p^n} \bmod p^N$. We can compute the trace $\mathrm{Tr}(\alpha)$ and norm $\mathrm{N}(\alpha)$ over $\mathbb{Z}_p \bmod p^N$ in time $\mathcal{O}((n\log q)^{1+\epsilon})$.*

Proof. If the reduction of α modulo p has degree n over $\mathbb{F}_p$, we can compute its minimal polynomial $f(X) \in \mathbb{Z}_p[X] \bmod p^N$ using Proposition 7. Its constant term equals $(-1)^n\mathrm{N}(\alpha)$ and the coefficient of X^{n-1} equals $-\mathrm{Tr}(\alpha)$. Suppose hence that $\alpha \bmod p$ has degree less than n over $\mathbb{F}_p$. Recall that $\mathbb{Z}_{p^n}$ is given as $\mathbb{Z}_p[x]/\varphi(x)$. Clearly the reductions of x, α/x and $\alpha - x$ modulo p have degree n over $\mathbb{F}_p$, so that their trace and norm can be computed using Proposition 7. Now the equalities $\mathrm{Tr}(\alpha) = \mathrm{Tr}(x) + \mathrm{Tr}(\alpha - x)$ and $\mathrm{N}(\alpha) = \mathrm{N}(x)\cdot\mathrm{N}(\alpha/x)$ allow us to compute the norm and trace of α. We note that in order to determine whether $\alpha \bmod p$ has degree n over $\mathbb{F}_p$, one can work as follows. The technique explained in the proof of Proposition 7 works always when we are working over $\mathbb{F}_p$ instead of $\mathbb{Z}_p \bmod p^N$, regardless of the degree of α. This hence gives the minimal polynomial of the reduction of α in time $\mathcal{O}((n\log p)^{1+\epsilon})$. □

We note that for computing $\mathrm{N}(\alpha)$ a much more elegant algorithm was given by Harley, see Section 12.8.5.c in [3]. Namely, if we write α as $\alpha(x)$, the resultant formula $\mathrm{N}(\alpha) = \mathrm{Res}_X(\varphi(X), \alpha(X))$ can be computed in the same amount of time as in Corollary 2, using a variant of Moenck's extended gcd algorithm [27].

A Teichmüller modulus is the minimal polynomial $\Phi(X)$ of a Teichmüller lift α (see Section 12.1 of [3]). Equivalently we can say that $\Phi(X)$ is the unique divisor in $\mathbb{Z}_p[X]$ of $X^{p^n} - X$ which reduces modulo p to the minimal polynomial of $\alpha \bmod p$. Combining Propositions 6 and 7 we then find:

Corollary 3. *Given $\mathbb{F}_{p^n} \cong \mathbb{F}_p[x]/\bar{\varphi}(x)$, we can compute a Teichmüller modulus $\Phi(X)$ modulo p^N which equals $\bar{\varphi}(X)$ modulo p in time $\mathcal{O}((n\log^2 p + n\log q)^{1+\epsilon})$.*

References

[1] W. Castryck, J. Denef, F. Vercauteren, Computing zeta functions of nondegenerate curves, IMRP Int. Math. Res. Pap. (2006), Art. ID 72017, 57.

[2] Wouter Castryck, Hendrik Hubrechts, Frederik Vercauteren, Computing zeta functions in families of $C_{a,b}$ curves using deformation, in: Algorithmic Number Theory, in: Lecture Notes in Comput. Sci., vol. 5011, Springer, Berlin, 2008, pp. 296–311.

[3] Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen, Frederik Vercauteren (Eds.), Handbook of Elliptic and Hyperelliptic Curve Cryptography, Discrete Math. Appl. (Boca Raton), Chapman & Hall/CRC, Boca Raton, FL, 2006.

[4] Jan Denef, Frederik Vercauteren, Counting points on C_{ab} curves using Monsky–Washnitzer cohomology, Finite Fields Appl. 12 (1) (2006) 78–102.

[5] Jan Denef, Frederik Vercauteren, An extension of Kedlaya's algorithm to hyperelliptic curves in characteristic 2, J. Cryptology 19 (1) (2006) 1–25;
Jan Denef, Frederik Vercauteren, Errata for "An extension of Kedlaya's algorithm to hyperelliptic curves in characteristic 2", and related papers, available on http://www.wis.kuleuven.be/algebra/denef_papers/ErrataPointCounting.pdf.

[6] Mireille Fouquet, Pierrick Gaudry, Robert Harley, An extension of Satoh's algorithm and its implementation, J. Ramanujan Math. Soc. 15 (4) (2000) 281–318.

[7] Pierrick Gaudry, Nicolas Gürel, An extension of Kedlaya's point-counting algorithm to superelliptic curves, in: Advances in Cryptology—ASIACRYPT 2001 (Gold Coast), in: Lecture Notes in Comput. Sci., vol. 2248, Springer, Berlin, 2001, pp. 480–494.

[8] Pierrick Gaudry, Nicolas Gürel, Counting points in medium characteristic using Kedlaya's algorithm, Experiment. Math. 12 (4) (2003) 395–402.

[9] Robert Harley, Asymptotically optimal p-adic point-counting, December 2002, e-mail to NMBRTHRY list.

[10] David Harvey, Kedlaya's algorithm in larger characteristic, Int. Math. Res. Not. IMRN 22 (2007), Art. ID rnm095, 29.

[11] Hendrik Hubrechts, Memory efficient hyperelliptic curve point counting, available on http://wis.kuleuven.be/algebra/hubrechts/, submitted for publication.

[12] Hendrik Hubrechts, Point counting in families of hyperelliptic curves in characteristic 2, LMS J. Comput. Math. 10 (2007) 207–234 (electronic).

[13] Hendrik Hubrechts, Point counting in families of hyperelliptic curves, Found. Comput. Math. 8 (1) (2008) 137–169.

[14] Hendrik Hubrechts, Quasi-quadratic elliptic curve point counting using rigid cohomology, J. Symbolic Comput. 44 (9) (2009) 1255–1267.

[15] K. Kedlaya, C. Umans, Fast polynomial factorization and modular composition, preprint, available on http://math.mit.edu/~kedlaya/papers/, SIAM J. Comput. (STOC 2008 Special Issue), in press.

[16] Kiran Kedlaya, Numerical computation of Coleman integrals, lecture available on http://math.mit.edu/~kedlaya/papers/renyi.pdf, in preparation.

[17] Kiran Kedlaya, Timothy G. Abbott, David Roe, Bounding Picard numbers of surfaces using p-adic cohomology, in: Arithmetic, Geometry and Coding Theory (AGCT 2005), in: Sémin. Congr., vol. 21, Societé Mathématique de France, 2009, in press.

[18] Kiran S. Kedlaya, Counting points on hyperelliptic curves using Monsky–Washnitzer cohomology, J. Ramanujan Math. Soc. 16 (4) (2001) 323–338.

[19] H.Y. Kim, J.Y. Park, J.H. Cheon, J.H. Park, J.H. Kim, S.G. Hahn, Fast elliptic curve point counting using Gaussian normal basis, in: Algorithmic Number Theory Symposium – ANTS V, in: Lecture Notes in Comput. Sci., vol. 2369, 2002, pp. 292–307.

[20] Neal Koblitz, p-Adic Numbers, p-Adic Analysis, and Zeta-Functions, second ed., Grad. Texts in Math., vol. 58, Springer-Verlag, New York, 1984.

[21] Alan G.B. Lauder, Deformation theory and the computation of zeta functions, Proc. London Math. Soc. (3) 88 (3) (2004) 565–602.

[22] Alan G.B. Lauder, A recursive method for computing zeta functions of varieties, LMS J. Comput. Math. 9 (2006) 222–269 (electronic).

[23] Alan G.B. Lauder, Counting solutions to equations in many variables over finite fields, Found. Comput. Math. 4 (3) (2004) 221–267.

[24] Alan G.B. Lauder, Daqing Wan, Counting points on varieties over finite fields of small characteristic, in: Algorithmic Number Theory: Lattices, Number Fields, Curves and Cryptography, in: Math. Sci. Res. Inst. Publ., vol. 44, Cambridge Univ. Press, Cambridge, 2008, pp. 579–612.

[25] Reynald Lercier, David Lubicz, Counting points in elliptic curves over finite fields of small characteristic in quasi-quadratic time, in: Advances in Cryptology—EUROCRYPT 2003, in: Lecture Notes in Comput. Sci., vol. 2656, Springer, Berlin, 2003, pp. 360–373.

[26] Jean-François Mestre, Lettre adressée à Gaudry et Harley, available on http://www.math.jussieu.fr/~mestre/.

[27] R.T. Moenck, Fast computation of GCDs, in: Fifth Annual ACM Symposium on Theory of Computing, Austin, TX, 1973, Assoc. Comput. Mach., New York, 1973, pp. 142–151.

[28] Josep Rifà, Joan Borrell, Improving the time complexity of the computation of irreducible and primitive polynomials in finite fields, in: Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, New Orleans, LA, 1991, in: Lecture Notes in Comput. Sci., vol. 539, Springer, Berlin, 1991, pp. 352–359.

[29] Takakazu Satoh, The canonical lift of an ordinary elliptic curve over a finite field and its point counting, J. Ramanujan Math. Soc. 15 (4) (2000) 247–270.
[30] Takakazu Satoh, Berit Skjernaa, Yuichiro Taguchi, Fast computation of canonical lifts of elliptic curves and its application to point counting, Finite Fields Appl. 9 (1) (2003) 89–101.
[31] Takakazu Satoh, Yuichiro Taguchi, Computing zeta functions for ordinary formal groups over finite fields, Discrete Appl. Math. 130 (1) (2003) 51–60, The 2000 Com^{2}MaC Workshop on Cryptography (Pohang).
[32] René Schoof, Counting points on elliptic curves over finite fields, J. Théor. Nombres Bordeaux 7 (1) (1995) 219–254, Les Dix-huitièmes Journées Arithmétiques (Bordeaux, 1993).
[33] Victor Shoup, New algorithms for finding irreducible polynomials over finite fields, Math. Comp. 54 (189) (1990) 435–447.
[34] Victor Shoup, Fast construction of irreducible polynomials over finite fields, J. Symbolic Comput. 17 (5) (1994) 371–391.
[35] Berit Skjernaa, Satoh's algorithm in characteristic 2, Math. Comp. 72 (241) (2003) 477–487 (electronic).
[36] J.-A. Thiong Ly, Note for computing the minimum polynomial of elements in large finite fields, in: Coding Theory and Applications, Toulon, 1988, in: Lecture Notes in Comput. Sci., vol. 388, Springer, New York, 1989, pp. 185–192.
[37] Frederik Vercauteren, Computing zeta functions of curves over finite fields, PhD thesis, K.U. Leuven, Belgium, 2003.
[38] Frederik Vercauteren, Bart Preneel, Joos Vandewalle, A memory efficient version of Satoh's algorithm, in: Advances in Cryptology—EUROCRYPT 2001 (Innsbruck), in: Lecture Notes in Comput. Sci., vol. 2045, Springer, Berlin, 2001, pp. 1–13.
[39] Joachim von zur Gathen, Jürgen Gerhard, Modern Computer Algebra, Cambridge University Press, Cambridge, 2003.