On the computation of norm residue symbols

Bouw, J.

Chapter 3

A computational model for local fields

1. Introduction

Let F be a finite extension of $\mathbf{Q}_p$. This is an uncountable field and hence it is not obvious how to do arithmetic in such a field. Just as in the field $\mathbf{R}$, we need to work with a 'precision' to make all our computations take place in finite sets. In this chapter, we answer the following questions:

- How can one represent F with a finite amount of data?
- How can one represent elements of F in a finite precision?
- How can one do basic arithmetic in F?

We answer the above questions, and compute bit complexities for many of the basic algorithms. In the next section, we discuss the main results. One can use these results as a black box for local fields. In the final section we answer the above questions.

In this chapter, we follow the notation of Chapter 2.

2. Main results

We will now discuss the conventions regarding the complexity of certain algorithms. The complexity of the algorithms below is given in bit complexity (not in terms of field operations in say $\mathbf{F}_p$). We usually use the big O notation, in the parameters e, f, p and N. We also use the $\tilde{O}$ notation as follows: here $h' \in \tilde{O}(h)$ means that there is an integer s such that $h' \in O(h \cdot (\log h)^s)$. In this thesis we use the following convention for complexity. If we write that the complexity is $O((N \log q)^{1[+1]})$ (or briefly just $(N \log q)^{1[+1]}$ in the tables below), it means that the complexity is $O((N \log q)^2)$ and also $\tilde{O}(N \log q)$. The faster complexity is usually obtained by using fast arithmetic.

Let $\mathbb{F}$ be a field, and $\mathcal{D}$ a basis of a finite dimensional vector space V over $\mathbb{F}$. If $T : V \to V$ is a linear map, we denote by $[T]_{\mathcal{D}}$ the matrix of T with respect to the basis $\mathcal{D}$. Furthermore, if $x \in V$ we denote by $[x]_{\mathcal{D}}$ the coordinates of x with respect to the basis $\mathcal{D}$. Finally, if $c \in \mathcal{O}_1$ we denote by $[\cdot c]_{\mathcal{B}}$ the matrix of the linear map $\cdot c : \mathcal{O}_1 \to \mathcal{O}_1$ with $x \to c \cdot x$ with respect to the basis $\mathcal{B}$. The ring of $n \times n$ matrices over a ring R is denoted by $\mathrm{Mat}_n(R)$.

DEFINITION 3.1. Let F be a local field and let $N \in \mathbf{Z}_{\geq 1}$. A *model* of F in precision N is a finite sequence of bits that specifies the ring $\mathcal{O}_N$, together with a representation of its elements; such a representation is defined to be a bijection from a set of finite sequences of bits to $\mathcal{O}_N$.

We remark that all O-constants are absolute, in particular independent of F and N.

THEOREM 3.2. *For every local field F and $N \in \mathbf{Z}_{\geq 1}$ there is a model of F in precision N such that the length of the sequence of bits that specifies $\mathcal{O}_N$ and the lengths of the sequences of bits that represent its elements are $O(N \log q)$, and such that one has the following algorithms for basic arithmetic:*

Algorithm	***Input***	***Output***	***Complexity***
Addition	$\mathcal{O}_N,\ x, y \in \mathcal{O}_N$	$x + y \in \mathcal{O}_N$	$N \log q$
Subtraction	$\mathcal{O}_N,\ x, y \in \mathcal{O}_N$	$x - y \in \mathcal{O}_N$	$N \log q$
Multiplication	$\mathcal{O}_N,\ x, y \in \mathcal{O}_N$	$x \cdot y \in \mathcal{O}_N$	$(N \log q)^{1[+1]}$
Powering	$\mathcal{O}_N,\ x \in \mathcal{O}_N,$ $r \in \mathbf{Z}_{\geq 0}$	$x^r \in \mathcal{O}_N$	$\log(r+2)\cdot$ $(N \log q)^{1[+1]}$
Inversion	$\mathcal{O}_N,\ x \in \mathcal{O}_N^*$	$1/x \in \mathcal{O}_N$	$(N \log q)^{1[+1]}$
Division	$\mathcal{O}_N,\ x \in \mathcal{O}_N,$ $y \in \mathcal{O}_N^*$	$x/y \in \mathcal{O}_N$	$(N \log q)^{1[+1]}$
Equality	$\mathcal{O}_N,\ x, y \in \mathcal{O}_N$	$\begin{cases} \text{True} & \text{if } x = y \\ \text{False} & \text{if } x \neq y \end{cases}$	$N \log q$
Unit?	$\mathcal{O}_N,\ x \in \mathcal{O}_N$	$\begin{cases} \text{True} & \text{if } x \in \mathcal{O}_N^* \\ \text{False} & \text{if } x \notin \mathcal{O}_N^* \end{cases}$	$N \log q$

One can obtain constants as follows:

Algorithm	***Input***	***Output***	***Complexity***
$0, 1, \pi, \gamma$	$\mathcal{O}_N$	$0, 1, \overline{\pi}, \overline{\gamma} \in \mathcal{O}_N$	$N \log q$
p, f, N	$\mathcal{O}_N$	p, f, N	$N \log q$
$N > e$?	$\mathcal{O}_N$	$\begin{cases} \text{True} & \text{if } N > e \\ \text{False} & \text{if } N \leq e \end{cases}$	$N \log q$
e	$\mathcal{O}_N$ *with* $N > e$	e	$N \log q$
$\mathcal{O}_M$	$\mathcal{O}_N,\ M \leq N$	$\mathcal{O}_M$	$N \log q$

Additionally, one has the following algorithms:

Algorithm	***Input***	***Output***	***Complexity***
Reducing	$\mathcal{O}_N,\ x \in \mathcal{O}_N,\ M \leq N$	$\mathcal{O}_M,\ \overline{x} \in \mathcal{O}_M$	$N \log q$
Lifting	$M \geq N,\ \mathcal{O}_M,\ \mathcal{O}_N$ $x \in \mathcal{O}_N$	$x' \in \mathcal{O}_M$ *with* $\overline{x'} = x$	$M \log q$
σ_{N-1}^{-1}	$\mathcal{O}_N$ *with* $N \geq 2,$ $x \in \mathcal{O}_N \cap \overline{U_{N-1}}$	$\mathcal{O}_1,\ \sigma_{N-1}^{-1}(x) \in \mathcal{O}_1$	$N \log q$
σ_{N-1}	$\mathcal{O}_N$ *with* $N \geq 2,\ c \in \mathcal{O}_1$	$\sigma_{N-1}(c) \in \mathcal{O}_N$	$N \log q$
u_0	$\mathcal{O}_N$ *with* $N > e$	$\mathcal{O}_{N-e},\ \overline{u_0} \in \mathcal{O}_{N-e}$	$N \log q +$ $((N-e) \log q)^{1[+1]}$
Teichmüller	$\mathcal{O}_N,\ c \in \mathcal{O}_1$	$\overline{\omega(c)} \in \mathcal{O}_N$	$\big(N + ((N/e) \log q)^{1[+1]}\big) \cdot$ $\log q$

Furthermore, one has the following algorithms regarding $k = \mathcal{O}_1$:

Algorithm	***Input***	***Output***	***Complexity***
$[x]_{\mathcal{B}}$	$\mathcal{O}_1,\ x \in \mathcal{O}_1$	$(a_b)_{b\in\mathcal{B}} \in \mathbf{F}_p^f$ $s.t.\ x = \sum_{b\in\mathcal{B}} a_b b$	$\log q$
$[\cdot c]_{\mathcal{B}}$	$\mathcal{O}_1,\ c \in \mathcal{O}_1$	$[\cdot c]_{\mathcal{B}} \in \mathrm{Mat}_f(\mathbf{F}_p)$	$f(\log q)^{1[+1]}$
$[x \mapsto x^p]_{\mathcal{B}}$	$\mathcal{O}_1$	$[x \mapsto x^p]_{\mathcal{B}} \in \mathrm{Mat}_f(\mathbf{F}_p)$	$(f+\log p)(\log q)^{1[+1]}$

The proof of the above theorem can be found in Section 4.

REMARK 3.3. Given a model $\mathcal{O}_N$, it is not the case that we can reconstruct F up to isomorphism. For example, if $N \leq e$ the ring $\mathcal{O}_N$ can come from different fields with different e. If N is big enough, then at least the isomorphism class of the field F is uniquely determined (Lemma 3.6). Hence properties of F can be read off from $\mathcal{O}_N$ for large enough N.

Let us explain how we handle the non-uniqueness of F in certain algorithms. One of the algorithms outputs $\overline{u_0} \in \mathcal{O}_{N-e}$, when given $\mathcal{O}_N$ with $N > e$ as input. Note that in this specific algorithm, we lose some precision. This means that our algorithm computes $\overline{u_0}$, and that the answer does not depend on the possible choice of F giving rise to $\mathcal{O}_N$.

REMARK 3.4. Once we can work with the rings $\mathcal{O}_N$, we can also work with F^*/U_N for any $N \in \mathbf{Z}_{\geq 1}$ as follows. By Corollary 2.6 one has $F^*/U_N \cong \mathbf{Z} \times \mathcal{O}_N^* \cong \mathbf{Z} \times k^* \times U_1/U_N$. Furthermore, we have an inclusion $U_1/U_N \to \mathcal{O}_N$. If $x = \pi^M \omega(c) v \pmod{U_N}$ corresponds to the (M, c, v), and y corresponds to (M', c', v'), then xy corresponds to $(M+M', cc', vv')$. The complexity of various operations, such as multiplication, now directly follows from the complexity of the operations in Theorem 3.2. With operations like addition, one has to be careful, since precision might be lost. Later in this thesis we usually work in quotients $F^*/(F^*)^m$, which are actually finite groups and hence we will not spend too much time on working out complexities for F^*/U_N.

3. Representing local fields

In this section we explain which data are used to represent a local field, and this will later motivate our construction for representing $\mathcal{O}_N$. We make use of two propositions, the first of which reads as follows.

PROPOSITION 3.5. *Let p be a prime number, e and f positive integers and let $g \in \mathbf{Z}_p[X]$ and $h \in \mathbf{Z}_p[X, Y]$ be polynomials with the following properties.*

i. *g is monic in X of degree f and irreducible modulo p.*

ii. *h has the form*

$$h = Y^e + \sum_{j=0}^{f-1} \sum_{i=0}^{e-1} h_{ij} X^j Y^i$$

with $h_{ij} \in p\mathbf{Z}_p$ for all i, j and $h_{0j} \notin p^2\mathbf{Z}_p$ for at least one j.

Then $F = \mathbf{Q}_p[X, Y]/(g, h)$ is a field, and $F/\mathbf{Q}_p$ has ramification index e and residue class degree f and $E = \mathbf{Q}_p[X]/(g)$ is the largest unramified subfield of F. One has $\mathcal{O}_E = \mathbf{Z}_p[X]/(g)$ and $\mathcal{O}_F = \mathbf{Z}_p[X, Y]/(g, h)$. Finally, set $\gamma = \overline{X}$ and $\pi = \overline{Y}$. Then π

is a prime element of F and $\mathcal{B} = \{1, \overline{\gamma}, \ldots, \overline{\gamma}^{f-1}\}$ forms a basis of the residue field of F over $\mathbf{F}_p$.

PROOF. The ideal (g) is a prime ideal in $\mathbf{Q}_p[X]$ because g is irreducible modulo p. It follows that $E = \mathbf{Q}_p[X]/(g)$ is a field. This field E is an unramified extension of $\mathbf{Q}_p$ of degree f (see [**24**, section 3.2, Theorem 3–2–6]. The field E has $\mathcal{O}_E = \mathbf{Z}_p[\gamma] \cong \mathbf{Z}_p[X]/(g)$ as its ring of integers (see [**24**, Ch. 3, section 3–2, Theorem 3–2–6(ii)]). The polynomial $h(\gamma, Y) \in E[Y]$ is an Eisenstein polynomial, so $F = E[Y]/(h)$ is a field. The field extension F/E is totally ramified of degree e (see [**24**, Theorem 3–3–1]). The field F has $\mathcal{O}_F = \mathcal{O}_E[\pi] = \mathbf{Z}_p[\gamma, \pi]$ as ring of integers (see [**24**, Ch. 3, Corollary 3–3–2]). So we have $\mathcal{O}_F = \mathbf{Z}_p[X, Y]/(g, h)$. Finally, π is a prime element of F (see [**24**, Ch. 3, section 3-3, Theorem 3–3–1(ii)]). The last statement follows easily (see [**24**, Ch. 3, Theorem 3-2-6]). □

We will now show that any local field F can be represented as in Proposition 3.5, and that we can make the defining coefficients small. Before we state and prove the second proposition we treat a lemma. We will alter apply the lemma below to $E = \mathbf{Q}_p[X]/(g)$ from Proposition 3.5.

LEMMA 3.6. *Suppose the field E is an unramified extension of $\mathbf{Q}_p$ and h_1 and h_2 are monic Eisenstein polynomials of degree e in $\mathcal{O}_E[Y]$ where $\mathcal{O}_E$ is the ring of integers of E. Suppose further that l is the largest positive integer such that $p^l \mid (p \cdot e)^2$. Then, if $p^l \mid h_1 - h_2$, we have $E[Y]/(h_1) \cong E[Y]/(h_2)$.*

PROOF. Suppose $\pi \in \overline{E}$, an algebraic closure of E, is a zero of the polynomial h_1. Since h_1 is Eisenstein, π is a prime element of $E(\pi)$.

First we will prove that $\mathrm{ord}_{E(\pi)} h_2(\pi) > 2 \cdot \mathrm{ord}_{E(\pi)} h_2'(\pi)$ where h_2' denotes the derivative of h_2. Since $p^l \mid h_1 - h_2$, we have

$$\mathrm{ord}_{E(\pi)} h_2(\pi) = \mathrm{ord}_{E(\pi)}(h_2 - h_1)(\pi) \geq e \cdot l.$$

Further we have

$$\mathrm{ord}_{E(\pi)} h_2'(\pi) \leq \mathrm{ord}_{E(\pi)}(e \cdot \pi^{e-1}),$$

because all terms of $h_2'(\pi)$ that are unequal to zero have different valuations. Hence we have

$$\begin{aligned} 2 \cdot \mathrm{ord}_{E(\pi)} h_2'(\pi) &\leq 2e \cdot \mathrm{ord}_p(e) + 2(e-1) \\ &< 2e \cdot (\mathrm{ord}_p(e) + 1) = 2e \cdot \mathrm{ord}_p(pe) \\ &= e \cdot l \leq \mathrm{ord}_{E(\pi)} h_2(\pi). \end{aligned}$$

With Newton's method and π as initial value we can now compute a zero π^* of h_2 in $E(\pi)$ (see [**24**, section 3-1]). We have $E(\pi^*) \subset E(\pi)$. Because the polynomials h_1 and h_2 are irreducible of the same degree, we conclude that the field extensions $E(\pi)/E$ and $E(\pi^*)/E$ have the same degree too. So $E(\pi) = E(\pi^*)$. This proves the assertion. □

The second proposition gives not only the converse of Proposition 3.5 but also includes the statement that we may choose the coefficients of g and h from a bounded interval in $\mathbf{Z}$ instead of from $\mathbf{Z}_p$.

PROPOSITION 3.7. *Let p be a prime number and F a finite extension of $\mathbf{Q}_p$ with ramification index e and residue class degree f. Suppose l is the largest positive integer for which p^l divides $(pe)^2$. Then there exist polynomials $g \in \mathbf{Z}[X]$ and $h \in \mathbf{Z}[X,Y]$ such that*

i. *g is monic in X of degree f and irreducible modulo p, and the coefficients g_i of g satisfy $0 \leq g_i \leq p-1$,*
ii. *h has the form*

$$h = Y^e + \sum_{j=0}^{f-1} \sum_{i=0}^{e-1} h_{ij} X^j Y^i$$

with $h_{ij} \in p\mathbf{Z}$ and $0 \leq h_{ij} \leq p^l - 1$ for all i, j, and $h_{0j} \notin p^2\mathbf{Z}$ for at least one j,
iii. *$F \cong \mathbf{Q}_p[X,Y]/(g,h)$.*

PROOF. Let $g = \sum_{i=0}^{f} g_i X^i \in \mathbf{Z}_p[X]$ of degree f which is irreducible modulo p and satisfies condition (i) (such g exists by the theory of finite fields). It is well known that the maximal unramified subextension E of F is isomorphic to $\mathbf{Q}_p[X]/(g)$. Fix such an isomorphism. Then pick a prime element π of F and note that $E(\pi) = F$. Consider the minimum polynomial of π over E, viewed over $\mathbf{Q}_p[X]/(g)$. This minimum polynomial h is an Eisenstein polynomial of the form as in (ii), except that the h_{ij} are in $p\mathbf{Z}_p$. Apply Lemma 3.6 to replace h with a polynomial of the required form. □

EXAMPLES 3.8. The following example of a field F illustrates how we present a field. Let $F \supset \mathbf{Q}_2$ be the field given by the triple $(p,g,h) = (2, X^2+X+1, Y^2-(2+2X)Y-2X)$. We denote the unramified part of F by $E = \mathbf{Q}_2(\gamma)$, where γ is a zero of $g(X) = X^2+X+1$. If we adjoin a zero of the Eisenstein polynomial $h(\gamma, Y)$ to E we obtain our field F, which is a totally ramified extension of E. Throughout this thesis we give examples where F is the field from this example.

If we choose a prime number p and polynomials $g = X$ and $h = Y - p$, we obtain the field $F_1 = \mathbf{Q}_p$.

The next example shows that one may naturally encounter polynomials that do not satisfy the conditions on their coefficients. Let F_2 be the cyclotomic field $\mathbf{Q}_p(\zeta_{p^k})$, with k a positive integer. This extension is totally ramified of degree $e = p^{k-1}(p-1)$ and $\zeta_{p^k} - 1$ is a prime element. The integer l from Proposition 3.7 satisfies $l = 2k$. One has $F \cong \mathbf{Q}_p[X]/(g,h)$ where $g(X) = X$ and

$$h(Y) = \frac{(Y+1)^{p^k}-1}{(Y+1)^{p^{k-1}}-1} = \sum_{i=0}^{p-1} (Y+1)^{ip^{k-1}} = Y^e + \ldots + (\sum_{j=0}^{p-1} jp^{k-1})Y + p.$$

For almost all pairs (p,k), the coefficient of the term of the polynomial h with $Y^{\frac{e}{2}}$ (if $p \neq 2$ or $k > 1$) fails to satisfy the inequality from Proposition 3.7ii. This is illustrated by choosing for example $p = 2$ and $k = 5$ because then the coefficients of the terms Y^t of $h(Y)$ with $4 \leq t \leq 12$ are bigger than $2^{10} - 1$.

REMARK 3.9. Let p, g, h and F be as in Proposition 3.7. Furthermore let d be the extension degree of the field F over $\mathbf{Q}_p$ and let L be the bit length of p, g, and h. Then we have

i. $L \geq d$.
ii. $L = O(d \log(pd))$.
iii. $L = O(d \log(2d))$ if F contains a primitive p-th root of unity.

Assertion (i) follows from the fact that we have to write down h and for each of its $d+1$ coefficients at least one bit is needed.

The f coefficients of the polynomial g can be written down in at most $f \cdot \log_2 p \leq d \cdot \log_2 p$ bits. The coefficients of the polynomial h are integers in the interval $[0, p^l - 1]$ with l as in Proposition 3.7. Hence h can be written down using at most $O(e \cdot f \cdot \log(p^l)) \leq O(d \cdot \log((pe)^2)) \leq O(d \cdot \log((pd)^2)) = O(d \cdot \log(pd))$ bits. Because the prime number p can be written down by $O(\log p)$ bits, we obtain the inequality $L = O(d \log(pd))$ bits. This proves assertion (ii).

If F contains a primitive p-th root of unity we have $d = [F : \mathbf{Q}_p] \geq p - 1$ and so $p \leq d + 1 \leq 2d$. If we take this into account, we obtain $L = O(d \log(2d))$. This proves assertion (iii).

4. Proof of main theorem

4.1. Representing $\mathcal{O}_N$ and its elements. Let F be a local field and let $N \in \mathbf{Z}_{\geq 1}$. Let us now discuss the data which define $\mathcal{O}_N = \mathcal{O}/\mathfrak{m}^N$. We call N the *precision* of the ring $\mathcal{O}_N$. Note that F can be given as in Proposition 3.5 by a triple (p, g, h), and we will define $\mathcal{O}_N$ with only a part of this information. Recall that $g \in \mathbf{Z}[X]$ and $h = Y^e + \sum_{j=0}^{f-1} \sum_{i=0}^{e-1} h_{ij} X^j Y^i \in \mathbf{Z}[X, Y]$.

The data for $\mathcal{O}_N$ for $N \geq 1$ are the following. The first part of the data is p and N. The second part of the information is a bit telling whether $N \leq e$ or $N > e$.The third part of the data is

$$g_N \equiv g \pmod{p^{\lceil \frac{N}{e} \rceil}} \in \left(\mathbf{Z}/p^{\lceil \frac{N}{e} \rceil}\mathbf{Z}\right)[X]$$

(if $N \leq e$, this is a polynomial in $(\mathbf{Z}/p\mathbf{Z})[X]$). Additionally, if $N > e$, we are given:

$$h_N \equiv h \pmod{p^{\lceil \frac{N}{e} \rceil}} = Y^e + \sum_{j=0}^{f-1} \sum_{i=0}^{e-1} h_{ij} X^j Y^i \bmod p^{\lceil \frac{N}{e} \rceil} \in \left(\mathbf{Z}/p^{\lceil \frac{N}{e} \rceil}\mathbf{Z}\right)[X, Y].$$

Proposition 3.10. *One has:*

$$\mathcal{O}_N \cong \mathbf{Z}_p[X, Y]/(g, h, Y^N) \cong \begin{cases} \left(\mathbf{Z}/p^{\lceil \frac{N}{e} \rceil}\mathbf{Z}\right)[X, Y]/(g_N, h_N, Y^N) & \text{if } N > e \\ (\mathbf{Z}/p\mathbf{Z})[X, Y]/(g_N, Y^N) & \text{if } N \leq e \end{cases}$$

Proof. The first isomorphism follows since $\overline{Y}$ is a prime element. The second isomorphism follows since we know that $p^{\lceil \frac{N}{e} \rceil} \in \mathfrak{m}^N$. Note that in the second case h_N is already in the ideal generated by g_N and Y^N. □

The data representing $\mathcal{O}_N$ in all cases have $O(N \log q)$ bits.

We will now discuss how elements of $\mathcal{O}_N$ are represented. Let π be the class of Y and γ be the class of X in $\mathcal{O}_N$. Note that any $x \in \mathcal{O}_N$ can be written uniquely as $\sum_{i=0}^{N-1} c_i \pi_i$ (recall Definition 2.3) with $c_i \in \mathcal{C}$, that is, we write $c_i = \sum_{j=0}^{f-1} d_{ij} \gamma^j$ with $0 \leq d_{ij} < p$. This is how we represent elements of $\mathcal{O}_N$ in $O(N \log q)$ bits.

4.2. Algorithms for a local field. In this section, we will explain the algorithms in Theorem 3.2. We assume that $\mathcal{O}_N$ is given as in the previous subsection, in $O(N \log q)$ bits. Hence elements in $\mathcal{O}_N$ are written as $\sum_{h=0}^{N-1} c_h \pi_h$ with $c_h \in \mathcal{C}$ and take up $O(N \log q)$ bits.

REMARK 3.11. In the rest of this thesis, we use that we can compute determinants and reduced row echelon forms, basis of kernel, cokernel, inverse, image of an $n \times n$ matrix over $\mathbf{F}_p$ in complexity $n^C (\log p)^{1[+1]}$, with $2 \leq C < 3$, where C is a "feasible matrix multiplication exponent"(see [**8**, Chapter 12], section 1).

Furthermore, we will use that we can do addition and subtraction in $\mathbf{Z}/p^m\mathbf{Z}$ in $O(\log(p^m))$ bit operations and multiplication and inversion in time $O((\log(p^m))^{1[+1]})$ bit operations (see [**8**, Chapter 5]).

Finally, we can compute determinants of $n \times n$ matrices over $\mathbf{Z}/p^m\mathbf{Z}$ in time $n^3 (\log(p^m))^{1[+1]}$ (by using row reductions). The latter can be improved, but we leave this to the reader.

The next lemma treats the complexity of some of the easy algorithms in Theorem 3.2.

LEMMA 3.12. *There algorithms for the following entries Theorem 3.2 run in the time as in Theorem 3.2:*

- *Equality;*
- *Unit?;*
- $0, 1, \pi, \gamma$*;*
- p, f, N*;*
- $N > e$*?;*
- e*;*
- $\mathcal{O}_M$*.*

PROOF. Only two algorithms require an explanation. For 'Unit?', an element $x = \sum_{h=0}^{N-1} c_h \pi_h \in \mathcal{O}_N$ is a unit if and only if $c_0 \neq 0$. For '$\mathcal{O}_M$', reduce the equations of $\mathcal{O}_N$ modulo the right power of p to obtain the model of $\mathcal{O}_M$. □

We have some other easy algorithms.

LEMMA 3.13. *There algorithms for the following entries Theorem 3.2 run in the time as in Theorem 3.2:*

- *Reducing;*
- *Lifting;*
- σ_{N-1}^{-1}*;*
- σ_{N-1}*.*

PROOF. Lifting and reducing are easy. The map σ_{N-1}^{-1} just sends $1 + c_{N-1}\pi_{N-1}$ to c_{N-1}. The map σ_{N-1} sends c to $1 + c\pi_{N-1}$. □

The next Lemma summarizes the discussion in [**8**, Chapter 2], on arithmetic operations in polynomial rings.

LEMMA 3.14. *Let R be a finite ring whose elements can be represented as finite sequences of bits and for which there are algorithms for the operations addition, subtraction and multiplication. Let $z \in R[T]$ be a monic polynomial of degree l. If an upper bound for the number of bit operations of an addition/subtraction and a multiplication in R is respectively denoted by t and u, then an addition/subtraction and a multiplication in $R[T]/(z)$, can be performed in respectively $O(lt)$ and $O(l^{1[+1]}(t+u))$ bit operations.*

PROOF. It is an easy verification that adding two elements of $R[T]/(z)$ comes down to l additions in R or $O(lt)$ bit operations. A multiplication of two elements of $R[T]/(z)$ requires $O(l^2)$ multiplications and additions of elements of R or $O(l^2(t+u))$ bit operations. Moreover the result of such a multiplication is a polynomial of degree at most $2l-2$ which is reduced by polynomial division by z. This division requires $l(l-1)(t+u)$ bit operations. Therefore the total cost of a multiplication in $R[T]/(z)$ is $O(l^2(t+u))$ bit operations. Using fast arithmetic one can reduce the factor l^2 in the runtime to $l^{1[+1]}$. □

The above lemma and its proof give a (standard) algorithm for computing in quotient rings and we apply this algorithm in our situation. We get the following result.

PROPOSITION 3.15. *There is an algorithm which on input $x, y \in \mathcal{O}_N$ computes $x+y \in \mathcal{O}_N$ and $x-y \in \mathcal{O}_N$ in time $O(N \log q)$, and $x \cdot y \in \mathcal{O}_N$ in time $O((N \log q)^{1[+1]})$.*

PROOF. Recall that

$$\mathcal{O}_N = \begin{cases} \left(\mathbf{Z}/p^{\lceil \frac{N}{e} \rceil}\mathbf{Z}\right)[X,Y]/(g_N, h_N, Y^N) & \text{if } N > e \\ \mathbf{Z}/p\mathbf{Z}[X,Y]/(g_N, Y^N) & \text{if } N \leq e. \end{cases}$$

In the second case, we can apply Lemma 3.14 twice to obtain the result.

In the first case, the situation is a bit trickier. We consider the ring

$$\mathcal{O}_{e\lceil \frac{N}{e} \rceil} = \left(\mathbf{Z}/p^{\lceil \frac{N}{e} \rceil}\mathbf{Z}\right)[X,Y]/(g_N, h_N, Y^{e\lceil \frac{N}{e} \rceil}) = \left(\mathbf{Z}/p^{\lceil \frac{N}{e} \rceil}\mathbf{Z}\right)[X,Y]/(g_N, h_N).$$

Lemma 3.14 allows us to do addition in time $O(N \log q)$ and multiplication in time $O((N \log q)^{1[+1]})$. Truncating the computations (reducing modulo Y^N, i.e. throwing away terms of the form $c_i \pi_i$ when $i \geq N$) allows us to do computations in $\mathcal{O}_N$ in the required time. □

Using repeated squaring, one can now compute the powers ('powering') of elements in $\mathcal{O}_N$ in the stated time.

We will now discuss an algorithm for computing inverses, with the help of a Newton iteration.

ALGORITHM 3.16 (Inverses).
Input: $u \in \mathcal{O}_N^*$.
Output: $u^{-1} \in \mathcal{O}_N$.
Steps:

i. Set $\overline{u} \in \mathcal{O}_1$.
ii. Compute $v_0 = \overline{u}^{-1} \in \mathcal{O}_1$ with the extended Euclidean algorithm.

iii. Compute $v_i \in \mathcal{O}_{\min(2^i,N)}$ for $1 \leq i \leq \lceil \log_2 N \rceil = j$ by $v_i = v'_{i-1} \cdot (2 - u \cdot v'_{i-1}) \in \mathcal{O}_{\min(2^i,N)}$ where v'_{i-1} is a lift of v_{i-1} to $\mathcal{O}_{\min(2^i,N)}$.
iv. Return $v = v_j \in \mathcal{O}_N$.

PROPOSITION 3.17. *Algorithm 3.16 is correct and has bit complexity* $O((N \log q)^{1[+1]})$.

PROOF. Computing $\overline{u}$ costs $O(N \log q)$ by Lemma 3.13. Applying the extended Euclidean Algorithm costs $O((\log q)^{1[+1]})$ bit operations. We refer to [**8**, Corollary 4.6] for this. In [**8**, Theorem 9.2] we find the proof that we can compute the inverse of a unit u by applying Newton iteration to the expression $f(x) = \frac{1}{ux} - 1$. The iteration gives the formula as in step iii and v_i is the inverse of u modulo $\mathfrak{m}^{\min(N,2^i)}$. The complexity of step iii is $O(\sum_{i=1}^{\lceil \log_2 N \rceil} (\min(2^i, N) \cdot \log q)^{1[+1]}) = O((N \log q)^{1[+1]})$ (Proposition 3.15, Lemma 3.13). This gives the required complexity. □

Note that for $x \in \mathcal{O}_N$, $y \in \mathcal{O}_N^*$ one has $x/y = x \cdot 1/y$. Hence we can now do division in the claimed time as well.

Recall that u_0 is defined by $p = -u_0 \pi^e$.

ALGORITHM 3.18 (u_0).
Input: $\mathcal{O}_N$ with $N > e$.
Output: $\overline{u_0} \in \mathcal{O}_{N-e}$.
Steps:

i. Compute $w = \sum_{i=0}^{e-1} \sum_{j=0}^{f-1} \frac{h_{ij}}{p} \gamma^j \pi^i \in \mathcal{O}_{N-e}$.
ii. Return $\overline{u_0} = w^{-1}$.

PROPOSITION 3.19. *Algorithm 3.18 is correct and its complexity is* $O(N \log q + ((N-e) \log q)^{1[+1]})$.

PROOF. If $h = Y^e + \sum_{j=0}^{f-1} \sum_{i=0}^{e-1} h_{ij} X^j Y^i$, then one has

$$1/u_0 = -\pi^e/p = \sum_{i=0}^{e-1} \sum_{j=0}^{f-1} \frac{h_{ij}}{p} \gamma^j \pi^i.$$

This formula allows us to compute $1/\overline{u_0} \in \mathcal{O}_{N-e}$ in time $O(N \log q)$ (we lose precision because of the division by p). We then invert $1/\overline{u_0}$ to get $\overline{u_0}$ in time $O(((N-e) \log q)^{1[+1]})$ (Algorithm 3.16).

□

Let us now discuss the complexity of the algorithms regarding the field k.

LEMMA 3.20. *There are algorithms for* $[x]_{\mathcal{B}}$, $[\cdot c]_{\mathcal{B}}$ *and* $[x \mapsto x^p]_{\mathcal{B}}$ *as in Theorem 3.2 which run in the times as stated in Theorem 3.2.*

PROOF. Since we work with digits, $[x]_{\mathcal{B}}$ is easy to compute.

To compute $[\cdot c]_{\mathcal{B}}$, we compute $c\overline{\gamma}^i$ for $i = 0, \ldots, f-1$ using f multiplications in $\mathcal{O}_1 = k$, in time $f(\log q)^{1[+1]}$. After that we compute $[c\overline{\gamma}^i]_{\mathcal{B}}$ for $i = 0, \ldots, f-1$ in time $f \log q$.

To compute $[x \mapsto x^p]_{\mathcal{B}}$, one raises $\overline{\gamma}$ to the p-th power and then compute $(\overline{\gamma}^p)^i$ for $i = 0, 1, \ldots, f-1$. This requires $f + \log p$ multiplications in k and this costs $O((f + \log p)(\log q)^{1[+1]})$. □

Let us finally discuss how to do Teichmüller lifts. To compute $\omega(c) \in \mathcal{O}_N$, it suffices to do computations in the unramified part E of F, that is, in the ring

$$\mathcal{O}_{E,\lceil \frac{N}{e} \rceil} = \left(\mathbf{Z}/p^{\lceil \frac{N}{e} \rceil}\mathbf{Z}\right)[X]/(g_N) = \left(\mathbf{Z}/p^{\lceil \frac{N}{e} \rceil}\mathbf{Z}\right)[X,Y]/(g_N, Y-p) \subseteq \mathcal{O}_N.$$

Algorithm 3.21 (Teichmüller).
Input: $c \in \mathcal{O}_1$.
Output: $\overline{\omega(c)} \in \mathcal{O}_N$.
Steps:

i. If $c = 0$ or $N \leq e$ return $\sum_{h=0}^{N-1} c_h \pi_h$ with $c_0 = c$ and $c_h = 0$ for $1 \leq h \leq N-1$ and terminate.
ii. Compute $(1-q)^{-1} = \sum_{i=0}^{\lceil \frac{N}{ef} \rceil - 1} p^{if} \in \mathcal{O}_{E,\lceil \frac{N}{e} \rceil}$.
iii. Put $x_0 = c \in k$ and for $1 \leq i \leq \lceil \log_2(N/e) \rceil = l$ compute $x_i = \frac{x'^q_{i-1} - q x'_{i-1}}{1-q} \in \mathcal{O}_{E,\min(2^i, N/e)}$ where x'_{i-1} is a lift of x_{i-1} to $\mathcal{O}_{E,\min(2^i,N/e)}$.
iv. Return $x_l \in \mathcal{O}_{E,\lceil \frac{N}{e} \rceil} \subset \mathcal{O}_N$.

Proposition 3.22. *Algorithm 3.21 is correct and its bit complexity is* $O\left(\left(N + ((N/e)\log q)^{1[+1]}\right) \cdot \log q\right)$.

Proof. If $N \leq e$ or $c = 0$, then $\overline{\omega(c)}$ is a lift of c to $\mathcal{O}_N$, which can be computed in time $O(N \log q)$. If $N > e$ step ii costs $O((N/e)\log q)$, step iv costs $O(N \log q)$ and the complexity of this algorithm is dominated by the third step. For the Newton iteration procedure with $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$ we choose $f(x) = 1 - x^{1-q}$ and obtain the formula of the third step of the algorithm. In every step the precision doubles so x_{i+1} is computed modulo $p^{2^{i+1}}$. The complexity of the last iteration x_l of the third step of Algorithm 3.21 dominates the cost of all the other iterations together and for this iteration we compute a q-th power requiring $O(((N/e)\log q)^{1[+1]} \cdot \log q)$ bit operations. The rest of this step has smaller complexity.

We conclude that Algorithm 3.21 has a complexity of $O(\left(N + ((N/e)\log q)^{1[+1]}\right) \cdot \log q)$ bit operations. □