JOURNAL
de Théorie des Nombres
de BORDEAUX

anciennement Séminaire de Théorie des Nombres de Bordeaux

Jordi GUÀRDIA, Jesús MONTES et Enric NART

Higher Newton polygons in the computation of discriminants and prime ideal decomposition in number fields

Tome 23, n^{o} 3 (2011), p. 667-696.

<http://jtnb.cedram.org/item?id=JTNB_2011__23_3_667_0>

Journal de Théorie des Nombres de Bordeaux **23** (2011), 667-696

Higher Newton polygons in the computation of discriminants and prime ideal decomposition in number fields

par Jordi GUÀRDIA, Jesús MONTES et Enric NART

Résumé. Nous présentons un algorithme pour calculer le discriminant et la décomposition des idéaux d'un corps de nombres en produit d'idéaux premiers. L'algorithme est un raffinement d'une méthode de factorisation p-adique qui utilise polygones de Newton d'ordre supérieur. L'exigence de mémoire et les temps d'exécution sont très bons.

Abstract. We present an algorithm for computing discriminants and prime ideal decomposition in number fields. The algorithm is a refinement of a p-adic factorization method based on Newton polygons of higher order. The running-time and memory requirements of the algorithm appear to be very good.

1. Introduction

The factorization of prime numbers in number fields is a classical problem, whose resolution lays at the foundation of algebraic number theory. Although it is completely understood from the theoretical point of view, the rising of computational number theory in the last decades has renewed the interest on the problem from a practical perspective. In his comprehensive book [2], H. Cohen refers to this problem as one of the main computational tasks in algebraic number theory.

The most common insight in the known solutions of the problem is based on the solution of a more general problem: the determination of a (local) integral basis. There are several efficient methods for this problem, most of them based on variants of the Round 2 and Round 4 routines [15], [16], [3], [4], [1], [5].

The theory of higher order Newton polygons developed in [12], and revised in [7], has revealed itself as a powerful tool for the analysis of the decomposition of a prime p in a number field. Newton polygons of higher order are a p-adic tool, and their computation involves no extension of the ground field but only extensions of the residue field; thus, they constitute an excellent device for a computational treatment of the problem. In this

Manuscrit reçu le 3 novembre 2008.

paper we explain how the theoretical results of [7] apply to yield an algorithm, due to the second author [12, Ch.3], to factor a prime number p in a number field K, in terms of a generating polynomial $f(x)$. The algorithm computes the p-valuation of the index of $f(x)$ as well; in particular, it determines the discriminant of the number field, if one is able to factorize the discriminant $\operatorname{disc}(f)$ of the defining equation.

In many applications, the computation of an integral basis is very useful because it helps to carry out other tasks in the number field. However, if one is interested only in the discriminant or in the factorization of a prime, our direct method has the advantage of being more efficient and it is able to carry out these tasks in number fields of much higher degree. The complexity of our algorithm has been analyzed by Ford-Veres [6] and Pauli [14]. Assuming fast multiplication, these results lead to an estimation of $O(n^{2+\epsilon}v_p(\operatorname{disc}(f))^{2+\epsilon})$ multiplications of integers less than p. Also, the algorithm has an excellent practical performance; the running-times and memory requirements of its implementation appear to be very good.

The outline of the paper is as follows. In section 2 we present the main technical ingredients of the paper: *types* and their *representatives*, and we review the basic algorithm that is obtained by a direct application of the ideas of [7]. In section 3 we characterize the optimal choices of representatives (Theorem 3.1). This result leads to an optimization of the basic algorithm, based on lowering the “order" at which the computations take place. In section 4 we show how to compute generators of the prime ideals lying above p in terms of the output of the algorithm. In section 5 we describe an implementation and in section 6 we present the results of some numerical tests. We construct some worst possible polynomials, that should be especially difficult with respect to the structure of the algorithm; this means that they have a huge index, and this index is sufficiently hidden to force the algorithm to work at a high order. The record is a polynomial of degree 1152 and 2-index 2153184, for which the factorization of 2 requires computations at order seven and it is obtained in less than two seconds. The algorithm, moreover, is highly parallelizable, so that it can raise the bounds of computations on number fields to huge degrees.

Last but not least, one can go the other way round and apply this algorithm to compute an integral basis [8]. This new approach provides a significant improvement in the solution of this problem as well.

Notation. Throughout the paper, K is a number field generated by a monic irreducible polynomial $f(x) \in \mathbb{Z}[x]$, $\theta \in K$ is a root of $f(x)$, and $\mathbb{Z}_K$ is the ring of integers of K.

Let $p \in \mathbb{Z}$ be a prime number and let $\mathbb{F} = GF(p)$ be the finite field with p elements. We fix an algebraic closure $\overline{\mathbb{Q}}_p$ of the field $\mathbb{Q}_p$ of p-adic numbers.

We denote simply by $v\colon \overline{\mathbb{Q}}_p \to \mathbb{Q}\cup\{\infty\}$ the canonical p-adic valuation normalized by $v(p)=1$.

If k is a field and $\varphi(y)$, $\psi(y)\in k[y]$, we write $\varphi\sim\psi$ to indicate that the two polynomials coincide up to multiplication by a nonzero constant in k.

2. Computation of discriminants and prime ideal decomposition

In this section we present a basic algorithm that computes the p-value of the discriminant of K and the prime ideal decomposition of $p\,\mathbb{Z}_K$. This algorithm is obtained by a direct application of the ideas of [7].

2.1. Types and their representatives. The basic tool for the algorithm is the concept of type and its representative, which we recall here with some detail. All results of this section are taken from [7, Sec. 2].

Definition. A type of order zero is a monic irreducible polynomial in $\mathbb{F}[y]$. Let $r\ge 1$ be a natural number. A *type of order r* is a sequence of data:

$$\mathbf{t}=(\psi_0(y);(\phi_1(x),\lambda_1,\psi_1(y));\dots;(\phi_r(x),\lambda_r,\psi_r(y))),$$

where $\phi_1(x),\dots,\phi_r(x)\in\mathbb{Z}[x]$, $\psi_0(y),\dots,\psi_r(y)\in\overline{\mathbb{F}}[y]$ are monic polynomials and $\lambda_1,\dots,\lambda_r\in\mathbb{Q}^-$ are negative rational numbers, that satisfy the following properties:

(1) The polynomial $\psi_0(y)\in\mathbb{F}[y]$ is irreducible and it coincides with the reduction of $\phi_1(y)$ modulo p. We define $\mathbb{F}_1:=\mathbb{F}[y]/(\psi_0(y))$.

(2) For all $1\le i<r$, the Newton polygon of i-th order, $N_i(\phi_{i+1})$, is one-sided, with positive length and slope λ_i.

(3) For all $1\le i<r$, the polynomial $\psi_i(y)\in\mathbb{F}_i[y]$ is irreducible and the residual polynomial of i-th order of ϕ_{i+1} satisfies $R_i(\phi_{i+1})(y)\sim\psi_i(y)$ in $\mathbb{F}_i[y]$. We define $\mathbb{F}_{i+1}:=\mathbb{F}_i[y]/(\psi_i(y))$.

(4) For all $1\le i<r$, the polynomial $\phi_{i+1}(x)$ has minimal degree among all monic polynomials in $\mathbb{Z}[x]$ satisfying (2) and (3).

(5) The polynomial $\psi_r(y)\in\mathbb{F}_r[y]$ is irreducible and $\psi_r(y)\ne y$. We define $\mathbb{F}_{r+1}:=\mathbb{F}_r[y]/(\psi_r(y))$.

Every type carries implicitly a certain amount of extra data, whose notation we fix now. We denote $e_0:=1$, $f_0:=\deg\psi_0=\deg\phi_1(x)$, and for all $1\le i\le r$:

- h_i, e_i are positive coprime integers such that $\lambda_i=-h_i/e_i$,
- ℓ_i, $\ell'_i\in\mathbb{Z}$ are fixed integers such that $\ell_ih_i-\ell'_ie_i=1$,
- $f_i=\deg\psi_i(y)$,
- $m_i=\deg\phi_i(x)=e_{i-1}f_{i-1}m_{i-1}$,
- $z_i=y\pmod{\psi_i(y)}\in\mathbb{F}^*_{i+1}$, $z_0=y\pmod{\psi_0(y)}\in\mathbb{F}^*_1$.

Also, we denote $m_{r+1}=m_re_rf_r$. The type determines a chain of finite extensions of $\mathbb{F}$:

$$\mathbb{F}=:\mathbb{F}_0\subset\mathbb{F}_1\subset\cdots\subset\mathbb{F}_{r+1},$$

such that $\mathbb{F}_{i+1}=\mathbb{F}_i[z_i]=\mathbb{F}[z_0,\dots,z_i]$, for all $0\le i\le r$. Also, for all $1\le i\le r+1$, the type carries certain p-adic discrete valuations $v_i:\mathbb{Q}_p(x)^*\to\mathbb{Z}$ [7, Def. 2.5], and semigroup homomorphisms,

$$\omega_i\colon \mathbb{Z}_p[x]\setminus\{0\}\to\mathbb{Z}_{\ge 0},\qquad P(x)\mapsto \operatorname{ord}_{\psi_{i-1}}(R_{i-1}(P)),$$

where $R_0(P)(y)\in\mathbb{F}[y]$ is the reduction modulo p of $P(y)/p^{v_1(P)}$. The integer $\omega_i(P)$ measures the length of the principal part, $N_i^-(P)$, of the Newton polygon of i-th order of $P(x)$ [7, Lem. 2.17]. The principal part is the polygon determined by the sides of negative slope of $N_i(P)$.

For every negative rational number $\lambda=-h/e$, with h,e positive coprime integers, the type $\mathbf{t}$ determines a residual polynomial operator of order i:

$$R_{\lambda,i}\colon \mathbb{Z}_p[x]\longrightarrow \mathbb{F}_i[y],\qquad P(x)\mapsto R_{\lambda,i}(P)(y),$$

for $1\le i\le r$. By definition, $R_i=R_{\lambda_i,i}$. These operators depend on the choice of the coefficients of the Bézout identity $\ell_ih_i-\ell_i'e_i=1$. However, if we replace ℓ_i by ℓ_i+me_i, then $R_{\lambda,i+1}(P)(y)$ changes into $\tau^uR_{\lambda,i+1}(P)(\tau^{-h}y)$, where $\tau=(z_i)^m\in\mathbb{F}_{i+1}^*$ is an absolute constant and the integer u depends on $P(x)$. We convene to choose $\ell_i=0$ when $e_i=1$.

To avoid confusion when we work simultaneously with different types, we write the type as a superscript in every datum: $\phi_i^{\mathbf{t}}(x)$, $\lambda_i^{\mathbf{t}}$, $e_i^{\mathbf{t}}$, etc.

Definition. We say that $\phi_i(x),\lambda_i,\psi_i(y)$ (and their implicit data) are the i-th *level* of $\mathbf{t}$.

Let $\mathbf{t}_0(f)$ be the set of all types of order zero that divide $f(x)$ modulo p. By Hensel's lemma, each $\mathbf{t}\in\mathbf{t}_0(f)$ determines a monic p-adic factor $f_{\mathbf{t}}(x)\in\mathbb{Z}_p[x]$ of $f(x)$, and

$$f(x)=\prod\nolimits_{\mathbf{t}\in\mathbf{t}_0(f)} f_{\mathbf{t}}(x).$$

Types of order r play an analogous role and provide similar factorizations at higher order. Let us recall some concepts and results in this regard.

Definition. Let $\mathbf{t}$ be a type of order r, and let $P(x)\in\mathbb{Z}_p[x]$ be a monic polynomial.

- We say that $P(x)$ is of type $\mathbf{t}$ if $\deg P=m_{r+1}\omega_{r+1}(P)>0$. This is equivalent to:
  (1) $P(x)\equiv\phi_1(x)^{a_0}\pmod p$, for some positive integer a_0, and
  (2) For all $1\le i\le r$, the Newton polygon $N_i(P)$ is one-sided of slope λ_i, and $R_i(P)(y)\sim\psi_i(y)^{a_i}$ in $\mathbb{F}_i[y]$, for some positive integer a_i.

  Also, $P(x)$ is of type $\mathbf{t}$ if and only if all its irreducible factors in $\mathbb{Z}_p[x]$ are of type $\mathbf{t}$.
- We say that $P(x)$ is divisible by $\mathbf{t}$, or that $\mathbf{t}$ divides $P(x)$, if $\omega_{r+1}^{\mathbf{t}}(P)>0$. Formally, we can think of $\omega_{r+1}^{\mathbf{t}}(P)$ as the exponent with which $\mathbf{t}$ divides $P(x)$.

- If $\mathbf{t}$ divides $P(x)$, we denote by $P_{\mathbf{t}}(x)$ the product of all monic irreducible factors of $P(x)$ in $\mathbb{Z}_p[x]$, which are of type $\mathbf{t}$. We have $\omega_{r+1}^{\mathbf{t}}(P_{\mathbf{t}}) = \omega_{r+1}^{\mathbf{t}}(P)$.
- We say that a set $\mathbf{T}$ of types faithfully represents $P(x)$, if $P(x)$ is divisible by all types in $\mathbf{T}$, and $P(x) = \prod_{\mathbf{t}\in\mathbf{T}} P_{\mathbf{t}}(x)$.

In [7, Sec. 3] we describe a constructive method to enlarge a type of order r into different types of order $r+1$. The crucial step is the construction of a polynomial of type $\mathbf{t}$ and minimal degree. Any such polynomial is called a *representative of the type* $\mathbf{t}$, and it plays the analogous role in order $r+1$ to that played by an irreducible polynomial modulo p in order one.

Theorem 2.1. *Let* $\mathbf{t}$ *be a type of order* r*. We can construct a monic polynomial* $\phi_{r+1}(x) \in \mathbb{Z}[x]$ *such that* $\deg \phi_{r+1} = m_{r+1}$ *and* $\omega_{r+1}(\phi_{r+1}) = 1$*. This polynomial is irreducible in* $\mathbb{Z}_p[x]$ *and* $v_{r+1}(\phi_{r+1}) = e_r f_r (e_r v_r(\phi_r) + h_r)$*.*

A representative of $\mathbf{t}$ facilitates its enlargement to types of higher order. We denote by $(\mathbf{t}; (\phi_{r+1}(x), \lambda_{r+1}, \psi_{r+1}(y)))$ the type of order $r+1$ obtained by adding at the $(r+1)$-th level any negative rational number λ_{r+1} and any irreducible monic polynomial $\psi_{r+1}(y) \in \mathbb{F}_{r+1}[y]$, $\psi_{r+1}(y) \neq y$.

2.2. Types versus prime ideals. The basic algorithm. Recall that we fixed a monic irreducible polynomial $f(x) \in \mathbb{Z}[x]$.

Definition. A type $\mathbf{t}$ of order r is said to be f*-complete* if $\omega_{r+1}^{\mathbf{t}}(f) = 1$.

Theorem 2.2 ([7, Cor. 3.8])**.** *Let* $\mathbf{t}$ *be an* f*-complete type of order* r*. Then the* p*-adic factor* $f_{\mathbf{t}}(x)$ *is irreducible in* $\mathbb{Z}_p[x]$*. Moreover, if* $L/\mathbb{Q}_p$ *is the extension generated by* $f_{\mathbf{t}}(x)$*, we have*

$$e(L/\mathbb{Q}_p) = e_1 \cdots e_r, \quad f(L/\mathbb{Q}_p) = f_0 f_1 \cdots f_r.$$

Thus, an f-complete type singles out a prime ideal $\mathfrak{p}$ dividing $p\,\mathbb{Z}_K$, whose ramification index and residual degree can be read in the data of $\mathbf{t}$:

$$e(\mathfrak{p}/p) = e_1 \cdots e_r, \quad f(\mathfrak{p}/p) = f_0 f_1 \cdots f_r.$$

The p-adic factorization process of [7] consists essentially in the construction of a set $\mathbf{T}$ of f-complete types, that faithfully represents $f(x)$. Thus, it can be interpreted as a *basic algorithm*, to determine the prime ideal decomposition of $p\,\mathbb{Z}_K$. The types are built iteratively by means of Theorem 2.1, and the theory of Newton polygons of higher order. We start with the set $\mathbf{T}_0(f) := \mathbf{t}_0(f)$, that faithfully represents $f(x)$. We extend the non-f-complete types of this set to types of order one, in order to construct a set $\mathbf{T}_1(f)$ that, again, faithfully represents $f(x)$, etc. At each order r, the extension process is carried out by a main loop that performs the following operations.

Main loop of the basic algorithm. At the input of a non-f-complete type $\mathbf{t}$ of order $r-1$, for which $\omega_r(f)>0$, and a representative $\phi_r(x)$:

1) Compute the Newton polygon of r-th order, $N_r(f)=S_1+\cdots+S_t$, with respect to $\mathbf{t}$ and $\phi_r(x)$.
2) For every side S_j of negative slope $\lambda_{r,j}<0$, compute the residual polynomial of r-th order, $R_{\lambda_{r,j},r}(f)(y)\in\mathbb{F}_r[y]$, with respect to $\mathbf{t}$, $\phi_r(x)$ and $\lambda_{r,j}$.
3) Factorize this polynomial in $\mathbb{F}_r[y]$ as a product of powers of pairwise different monic irreducible polynomials:
$$R_{\lambda_{r,j},r}(f)(y)\sim\psi_{r,1}(y)^{a_1}\cdots\psi_{r,s}(y)^{a_s}.$$
4) For every factor $\psi_{r,k}(y)$, compute a representative of the following type of order r:
$$\mathbf{t}^{j,k}:=(\mathbf{t};(\phi_r(x),\lambda_{r,j},\psi_{r,k}(y))).$$

For those factors $\psi_{r,k}(y)$ with exponent $a_k=1$, the type $\mathbf{t}^{j,k}$ is complete. For the remaining types we continue the iterative process.

Thus, each non-complete type $\mathbf{t}$ of order $r-1$ sprouts several types of order r, which are called *branches* of $\mathbf{t}$. We have a factorization in $\mathbb{Z}_p[x]$:
$$f_{\mathbf{t}}(x)=\prod\nolimits_{j,k}f_{\mathbf{t}^{j,k}}(x),$$
with $\deg f_{\mathbf{t}^{j,k}}=e_{r,j}f_{r,k}m_r$. Also, $(\omega_{r+1})^{\mathbf{t}^{j,k}}(f)=a_k>0$, for all j,k, and
$$\omega_r^{\mathbf{t}}(f)=\sum_{j,k}e_{r,j}f_{r,k}(\omega_{r+1})^{\mathbf{t}^{j,k}}(f).\tag{2.1}$$

Hence, the invariant $\omega_r^{\mathbf{t}}(f)$ is an upper bound for the number of irreducible factors of $f_{\mathbf{t}}(x)$, and it is a kind of measure of the distance that is left to complete the analysis of the type $\mathbf{t}$ and its branches (or equivalently, to decompose each $f_{\mathbf{t}^{j,k}}(x)$ into a product of irreducible factors). Also, (2.1) shows that, except for the case in which there is only one branch with $e_r=f_r=1$, the branches are always closer to be f-complete than $\mathbf{t}$.

We denote by $\mathbf{t}_r(f)$ the set of types of order r obtained by aplying the main loop to all non-f-complete types of $\mathbf{t}_{r-1}(f)$. We denote by $\mathbf{t}_i(f)^{\mathrm{compl}}$ the subset of the f-complete types of $\mathbf{t}_i(f)$, and we define
$$\mathbf{T}_r(f):=\mathbf{t}_r(f)\cup\left(\bigcup\nolimits_{0\le i<r}\mathbf{t}_i(f)^{\mathrm{compl}}\right).$$

Proposition 2.3 ([7, Sec. 3])**.** *$\mathbf{T}_r(f)$ faithfully represents $f(x)$.*

To show that the basic algorithm deserves this name, we have to prove that, after a finite number of enlargements, all types of $\mathbf{t}_r(f)$ will be complete. To this purpose we introduce another variable to measure how far a type is from being complete, that works even in the unibranch case with $e_r=f_r=1$. This control variable is defined in terms of *higher indices.*

2.3. Indices of higher order. The results of this section are taken from [7, Sec. 4]. Denote

$$\operatorname{ind}(f) := v\left((\mathbb{Z}_K \colon \mathbb{Z}[\theta])\right),$$

and recall the well-known relationship, $v(\operatorname{disc}(f)) = 2\operatorname{ind}(f) + v(\operatorname{disc}(K))$, between $\operatorname{ind}(f)$, the discriminant of $f(x)$ and the discriminant of K.

Definition. Let $N = S_1 + \cdots + S_t$ be a principal polygon with sides ordered by increasing slopes $\lambda_1 < \cdots < \lambda_t < 0$. Denote by $E_i = \ell(S_i)$, $H_i = H(S_i)$, $d_i = d(S_i)$ the respective length, height and degree of each side [7, Sec. 1.1]. We define the *index of the polygon* N to be the nonnegative integer

$$\operatorname{ind}(N) := \sum_{i=1}^{t} \frac{1}{2}(E_iH_i - E_i - H_i + d_i) + \sum_{1\le i<j\le t} E_iH_j.$$

This number is equal to the number of points with integral coordinates that lie below or on the polygon, strictly above the horizontal line that passes through the last point of N and strictly beyond the vertical axis. Hence, $\operatorname{ind}(N) = 0$ if and only if N has a unique side with height $H = 1$, or length $E = 1$.

Definition. Let $\mathbf{t}$ be a type of order $r-1$, and let $\phi_r(x)$ be a representative of $\mathbf{t}$. We define its *f-index* to be the nonnegative integer

$$\operatorname{ind}_{\mathbf{t}}(f) := \operatorname{ind}_{\mathbf{t},\phi_r}(f) := f_0 \cdots f_{r-1} \operatorname{ind}(N_r^-(f)),$$

the Newton polygon of r-th order taken with respect to $\mathbf{t}$ and $\phi_r(x)$.

We say that $\mathbf{t}$ is *f-maximal* if $\mathbf{t}$ divides $f(x)$ and $\operatorname{ind}_{\mathbf{t}}(f) = 0$.

For any natural number $r \ge 1$, we define $\operatorname{ind}_r(f) := \sum_{\mathbf{t}\in\mathbf{t}_{r-1}(f)} \operatorname{ind}_{\mathbf{t}}(f)$.

Since the Newton polygon $N_r^-(f)$ depends on the choice of $\phi_r(x)$, the value $\operatorname{ind}_{\mathbf{t}}(f)$, and the fact of being f-maximal, depends on this choice too.

Proposition 2.4 ([7, Lem. 4.16])**.**

a) *If $\mathbf{t}$ is f-complete, then it is f-maximal.*

b) *If $\mathbf{t}$ is f-maximal, then either $\mathbf{t}$ is f-complete, or the output of the main loop applied to $\mathbf{t}$ is a unique branch of order $r+1$ which is f-complete.*

Thus, the fact that all types of $\mathbf{t}_r(f)$ are complete is essentially equivalent to the fact that they are all maximal. The proof that this will occur after a finite number of iterations is provided by the theorem of the index.

Theorem 2.5 (Theorem of the index [7, Thm. 4.18])**.** *For all $r \ge 1$,*

$$\operatorname{ind}(f) \ge \operatorname{ind}_1(f) + \cdots + \operatorname{ind}_r(f), \tag{2.2}$$

and equality holds if and only if $\operatorname{ind}_{r+1} = 0$.

This theorem shows that after a finite number of iterations all types of $\mathbf{t}_r(f)$ will be f-maximal, because the sum of the right-hand side is bounded by the absolute constant $\operatorname{ind}(f)$. By (b) of Proposition 2.4, either $\mathbf{T}_r(f)$ or $\mathbf{T}_{r+1}(f)$ will contain only f-complete types. By Theorem 2.2 and Proposition 2.3, these complete types determine the complete factorization of $p\,\mathbb{Z}_K$ into a product of prime ideals. At this final stage we have necessarily an equality in (2.2), so that we get a computation of $\operatorname{ind}(f)$ as a by-product.

Remark 2.6. If at the end of step 1 of the main loop of the basic algorithm, we accumulate to a global variable the value $\operatorname{ind}_{\mathbf{t}}(f)$, the final output of this global variable is $\operatorname{ind}(f)$. In particular, $\operatorname{ind}_{\mathbf{t}}(f)$ is an absolute measure of the distance covered by each iteration of the main loop, towards the end of the algorithm.

Theorem 2.7. *Given a number field K, a generating polynomial $f(x) \in \mathbb{Z}[x]$, and a prime number p, we can construct a set $\mathbf{T}$ of f-complete types, that faithfully represents $f(x)$. The types of $\mathbf{T}$ are in 1-1 correspondence with the prime ideals of K lying above p, and the ramification index and residual degree of each ideal can be read from data of the corresponding type. Along the construction of $\mathbf{T}$, the algorithm computes the p-valuation of the index of $f(x)$ as well.*

The theorem of the index and Proposition 2.4 show that the number of iterations of the main loop is bounded by $\operatorname{ind}(f)$. In practice, the number of iterations is much lower.

Remark 2.8. In each iteration of the main loop, $\operatorname{ind}_{\mathbf{t}}(f)$ is usually much bigger than one, due to the abundance of the number of points of integer coordinates below an average Newton polygon with a fixed length $\omega_r^{\mathbf{t}}(f)$, and the fact that these points are counted with weight $f_0 \cdots f_{r-1}$.

In the next section we introduce a crucial optimization. A *refinement process* will control at each iteration wether it is strictly necessary to build a type of higher order, or it is possible to keep working at the same order, to avoid an increase of the recursivity in the computations. For instance, the polynomial $f(x) = (x-2)^2 + 2^{2k}$ would require the construction of types of order $\approx k$ in a strict application of the basic algorithm, while it can be completely analyzed with a refined type of order 1.

3. Optimal representatives of types

3.1. Detection of optimal representatives. The construction of types dividing a given polynomial is not canonical: in the construction of the representatives $\phi_r(x)$ one makes some choices, mainly related to lifting certain polynomials over finite fields to polynomials over $\mathbb{Z}$. A natural question arises: are there optimal choices?

Consider the following trivial example: let $p = 2$, $f(x) = x^2 - 4x + 12$, and $K = \mathbb{Q}(\theta) = \mathbb{Q}(\sqrt{-2})$. The polynomial factorizes modulo 2 as $\overline{f}(y) = y^2$; thus, the type of order zero $\mathbf{t} = \psi_0(y) = y$ is not f-complete. The more natural lifting of ψ_0 to $\mathbb{Z}[x]$ is $\phi_1(x) = x$, and the corresponding Newton polygon and residual polynomial determine a unique extension of $\mathbf{t}$ to a type of order one, $(y; (x, -1, y + 1))$, which is still not complete, so that we must construct a type of (at least) order 2 to determine the factorization of $2\,\mathbb{Z}_K$. If we choose instead, $\phi_1(x) = x - 2$, we have $f(x) = (x - 2)^2 + 2^3$, and the type of order one $(y; (\phi_1(x), -3/2, y + 1))$ is f-complete. Thus, this second choice of $\phi_1(x)$ is better.

While it seems very difficult to predict a priori whether a choice of $\phi_r(x)$ is better than another, it is possible a posteriori to know if our choice was optimal and, if this is not the case, to improve its quality.

Theorem 3.1. *Let $\mathbf{t}_0$ be a type of order $r - 1$ which divides $f(x)$ but it is not f-complete, and let $\phi_r(x)$ be a representative of $\mathbf{t}_0$. Let $\mathbf{t} = (\mathbf{t}_0; (\phi_r(x), \lambda_r, \psi_r(y)))$ be an extension of $\mathbf{t}_0$ to a type of order r still dividing $f(x)$, and let $f_{\mathbf{t}}(x) \in \mathbb{Z}_p[x]$ be the factor of $f(x)$ determined by $\mathbf{t}$. Let $\phi'_r(x) \in \mathbb{Z}[X]$ be another representative of $\mathbf{t}_0$. If $e_r f_r > 1$, then,*

a) *The Newton polygon $N'_r(f_{\mathbf{t}})$, with respect to $\mathbf{t}_0$ and $\phi'_r(x)$, is one-sided with slope $|\lambda'_r| \le |\lambda_r|$, and it has the same right end point as $N_r(f_{\mathbf{t}})$.*
b) *The residual polynomial $R'_r(f_{\mathbf{t}})(y)$, with respect to $\mathbf{t}_0$, $\phi'_r(x)$ and λ'_r, has only one irreducible factor in $\mathbb{F}_r[y]$; that is, $R'_r(f_{\mathbf{t}})(y) \sim \psi'_r(y)^{a'_r}$, for some monic irreducible polynomial $\psi'_r(y) \in \mathbb{F}_r[y]$ and some positive exponent a'_r.*
c) *Let $\mathbf{t}' = (\mathbf{t}_0; (\phi'_r(x), \lambda'_r, \psi'_r(y)))$. If $|\lambda'_r| < |\lambda_r|$, then $e'_r = f'_r = 1$. If $\lambda'_r = \lambda_r$, then $f_{\mathbf{t}}(x) = f_{\mathbf{t}'}(x)$, $e'_r = e_r$, $f'_r = f_r$, and $\omega^{\mathbf{t}'}_{r+1}(f) = \omega^{\mathbf{t}}_{r+1}(f)$.*

According to this theorem, if $e_r f_r > 1$, the representative $\phi_r(x)$ is optimal for the branch $\mathbf{t}$ of order r of $\mathbf{t}_0$. Items a), b) show that for any other choice $\phi'_r(x)$ the branch $\mathbf{t}$ is replaced by a single branch $\mathbf{t}'$, and $f_{\mathbf{t}}(x)$ is always of type $\mathbf{t}'$. Hence, $f_{\mathbf{t}}(x)$ divides $f_{\mathbf{t}'}(x)$, so that there is no choice of a representative of $\mathbf{t}_0$ leading to a proper factorization of $f_{\mathbf{t}}(x)$.

Also, if $\lambda'_r = \lambda_r$, the types $\mathbf{t}$, $\mathbf{t}'$ face the same obstruction for the future development of the factorization algorithm, because $f_{\mathbf{t}}(x) = f_{\mathbf{t}'}(x)$ and $\omega^{\mathbf{t}}_{r+1}(f) = \omega^{\mathbf{t}'}_{r+1}(f)$. However, if $|\lambda'_r| < |\lambda_r|$, the type $\mathbf{t}'$ is worse than $\mathbf{t}$ in this regard. In fact, we shall see in section 3.2 that the condition $e_r f_r > 1$ is also necessary for the optimality of $\phi_r(x)$ with respect to the branch $\mathbf{t}$.

Let us remark that a choice of the representative $\phi_r(x)$ of $\mathbf{t}_0$ can be optimal for some branches $\mathbf{t}$ of $\mathbf{t}_0$ and non-optimal for other branches.

For the proof of Theorem 3.1, we need an auxiliary result. Fix a type $\mathbf{t}_0$ of order $r-1$ dividing $f(x)$. For any $\mathbf{n}=(n_0,\dots,n_{r-1})\in\mathbb{N}^r$, denote $\Phi(\mathbf{n})(x)=p^{n_0}\phi_1(x)^{n_1}\dots\phi_{r-1}(x)^{n_{r-1}}$. Let $\theta\in\overline{\mathbb{Q}}_p$ be a root of $f_{\mathbf{t}_0}(x)$, $L=\mathbb{Q}_p(\theta)$, $\mathcal{O}_L$ the ring of integers of L, $\mathfrak{m}_L$ the maximal ideal and $\mathbb{F}_L=\mathcal{O}_L/\mathfrak{m}_L$ the residue field. We denote simply by $\alpha\mapsto\overline{\alpha}$, the reduction map, $\mathcal{O}_L\to\mathbb{F}_L$. In [7, (27)], an embedding $\mathbb{F}_r\hookrightarrow\mathbb{F}_L$, is defined by

$$\iota_\theta\colon\mathbb{F}_r\hookrightarrow\mathbb{F}_L,\quad z_0\mapsto\bar{\theta},\ z_1\mapsto\overline{\gamma_1(\theta)},\dots,\ z_{r-1}\mapsto\overline{\gamma_{r-1}(\theta)},\tag{3.1}$$

for certain rational functions $\gamma_i(x)\in\mathbb{Q}(x)$ such that $v(\gamma_i(\theta))=0$ [7, Def. 2.13, Cor. 3.2].

Lemma 3.2. *Let $\mathbf{t}_0,\theta,L$ be as above. Let $M(x)\in\mathbb{Z}[x]$ be a polynomial of degree less than m_r. Suppose that $\mathbf{n}\in\mathbb{N}^r$ satisfies $v(M(\theta))=v(\Phi(\mathbf{n})(\theta))$. Then, $\overline{M(\theta)/\Phi(\mathbf{n})(\theta)}\in\mathbb{F}_L^*$ belongs to $\iota_\theta(\mathbb{F}_r)$, and the element $\iota_\theta^{-1}(\overline{M(\theta)/\Phi(\mathbf{n})(\theta)})\in\mathbb{F}_r^*$ is independent of the choice of θ.*

Proof. Let $J:=\{\mathbf{j}=(j_0,\dots,j_{r-1})\in\mathbb{N}^r\mid 0\le j_i<e_if_i,\text{ for }0\le i<r\}$. Since $\deg M<m_r$, this polynomial can be written in a unique way as

$$M(x)=\sum_{\mathbf{j}=(j_0,\dots,j_{r-1})\in J}a_{\mathbf{j}}x^{j_0}\Phi(0,j_1,\dots,j_{r-1})(x),$$

for certain integers $a_{\mathbf{j}}$. We have $v(\theta^{j_0})=0$, and [7, Lem. 4.21] shows that

$$v(a_{\mathbf{j}})\ge\delta_{\mathbf{j}}:=v(M(\theta))-v(\Phi(0,j_1,\dots,j_{r-1})(\theta)),$$

for all $\mathbf{j}\in J$. Let $J_0=\{\mathbf{j}\in J\mid v(a_{\mathbf{j}})=\delta_{\mathbf{j}}\}$. Denote $b_{\mathbf{j}}=a_{\mathbf{j}}/p^{\delta_{\mathbf{j}}}$, and $\mathbf{j}'=(\delta_{\mathbf{j}},j_1,\dots,j_{r-1})$. We can write $M(x)$ as

$$M(x)=\sum_{\mathbf{j}\in J_0}b_{\mathbf{j}}x^{j_0}\Phi(\mathbf{j}')(x)+N(x),$$

where $N(x)\in\mathbb{Z}[x]$ satisfies $v(N(\theta))>v(M(\theta))$. Now,

$$\frac{M(x)}{\Phi(\mathbf{n})(x)}=\sum_{\mathbf{j}\in J_0}b_{\mathbf{j}}x^{j_0}\Phi(\mathbf{j}'-\mathbf{n})(x)+\frac{N(x)}{\Phi(\mathbf{n})(x)}.$$

By hypothesis, $v(\Phi(\mathbf{j}'-\mathbf{n})(\theta))=0$. Since $\omega_{r+1}(\Phi(\mathbf{j}'-\mathbf{n}))=0$ [7, Prop. 2.15], we have $v_r(\Phi(\mathbf{j}'-\mathbf{n})(x))=0$, by [7, Prop. 2.9]. By [7, Lem. 2.16], there exists a sequence $i_1,\dots,i_{r-1}$ of integers, that depend only on $\mathbf{j}'$ and $\mathbf{n}$, such that

$$\Phi(\mathbf{j}'-\mathbf{n})(x)=\gamma_1(x)^{i_1}\cdots\gamma_{r-1}(x)^{i_{r-1}}.$$

Hence, the element of $\mathbb{F}_L^*$,

$$\overline{M(\theta)/\Phi(\mathbf{n})(\theta)}=\textstyle\sum_{\mathbf{j}\in J_0}\bar{b}_{\mathbf{j}}\bar{\theta}^{j_0}\overline{\Phi(\mathbf{j}'-\mathbf{n})(\theta)},$$

belongs to $\iota_\theta(\mathbb{F}_r)$. Since all the ingredients $a_{\mathbf{j}}$, $\delta_{\mathbf{j}}$, $i_1,\dots,i_{r-1}$ etc. depend only on $\mathbf{t}_0$, the element $\iota_\theta^{-1}(\overline{M(\theta)/\Pi(\theta)})\in\mathbb{F}_r^*$ is independent of θ. □

Proof of Theorem 3.1. Let $\theta \in \overline{\mathbb{Q}}_p$ be now a root of $f_{\mathbf{t}}(x)$, and $L = \mathbb{Q}_p(\theta)$. Let us write $\phi'_r(x) = \phi_r(x) + M(x)$, for certain $M(x) \in \mathbb{Z}[x]$ of degree less than m_r. Since $\deg M < m_r$, we have $\omega_{r+1}(M) = 0$ [7, Lem. 2.2]. The theorem of the polygon [7, Thm. 3.1] and [7, Prop. 2.9] show that

(3.2)
$$v(\phi_r(\theta)) = (v_r(\phi_r) + |\lambda_r|)/e_1 \cdots e_{r-1}, \quad v(M(\theta)) = v_r(M)/e_1 \cdots e_{r-1},$$

where $v_r = v_r^{\mathbf{t}_0}$. In particular, $v(\phi_r(\theta))$ and $v(M(\theta))$ are independent of the choice of θ as a root of $f_{\mathbf{t}}(x)$.

Let us prove first that $v(\phi_r(\theta)) \ge v(\phi'_r(\theta))$, by showing that the opposite inequality implies $e_r = f_r = 1$. In fact, if $v(\phi_r(\theta)) < v(\phi'_r(\theta))$, then $v(M(\theta)) = v(\phi_r(\theta))$, and (3.2) shows that $v_r(M) = v_r(\phi_r) + |\lambda_r|$. Hence λ_r is an integer, and $e_r = 1$.

We now use some other rational functions introduced in [7, Def. 2.13]:

$$\gamma_r(x) = \frac{\Phi_r(x)}{\pi_r(x)^{h_r}} = \frac{\phi_r(x)}{\pi_r(x)^{h_r}\pi_{r-1}(x)^{f_{r-1}v_r(\phi_{r-1})}}.$$

Denote $\Pi(x) = \pi_r(x)^{h_r}\pi_{r-1}(x)^{f_{r-1}v_r(\phi_{r-1})}$. Since $v(\gamma_r(\theta)) = 0$, we have $v(\Pi(\theta)) = v(\phi_r(\theta)) = v(M(\theta))$. By [7, (17)], we can write $\Pi(x) = \Phi(\mathbf{n})(x)$, for some $\mathbf{n} \in \mathbb{N}^r$ that depends only on $\mathbf{t}_0$. Hence, if we reduce modulo $\mathfrak{m}_L$ the identity

$$\frac{\phi'_r(\theta)}{\Pi(\theta))} = \gamma_r(\theta) + \frac{M(\theta)}{\Pi(\theta))},$$

Lemma 3.2 shows that $\overline{\gamma_r(\theta)} = -\overline{M(\theta)/\Pi(\theta))}$ belongs to $\iota_\theta(\mathbb{F}_r)$. Since $\overline{\gamma_r(\theta)}$ is a root of $\iota_\theta(\psi_r(y))$ [7, Prop. 3.5], we get $f_r = 1$. This ends the proof of the inequality $v(\phi_r(\theta)) \ge v(\phi'_r(\theta))$, which is valid for all roots θ of $f_{\mathbf{t}}(x)$.

We now prove item a) of the theorem. Since ϕ_r, ϕ'_r are representatives of the same type $\mathbf{t}_0$, Theorem 2.1 shows that $\deg \phi_r = \deg \phi'_r = m_r$ and $v_r(\phi_r) = v_r(\phi'_r)$. Therefore, the Newton polygons $N_r(f_{\mathbf{t}})$ and $N'_r(f_{\mathbf{t}})$ have the same right end point $(\deg f_{\mathbf{t}}/m_r, v_r(\phi_r)\deg f_{\mathbf{t}}/m_r)$.

If we show that $v(\phi'_r(\theta))$ takes the same value for all the roots θ of $f_{\mathbf{t}}(x)$, then, by [7, Thm. 3.1], $N'_r(f_{\mathbf{t}})$ will be one-sided with slope

$$\lambda'_r = v_r(\phi'_r) - e_1 \cdots e_{r-1} v(\phi'_r(\theta)) \ge v_r(\phi_r) - e_1 \cdots e_{r-1} v(\phi_r(\theta)) = \lambda_r,$$

and item a) will be proven. Now, if $v(\phi'_r(\theta)) = v(\phi_r(\theta))$ for all θ, then the value $v(\phi'_r(\theta))$ is constant, because the value $v(\phi_r(\theta))$ is constant. If there is one θ_0 with $v(\phi'_r(\theta_0)) < v(\phi_r(\theta_0))$, then $v(M(\theta_0)) = v(\phi'_r(\theta_0)) < v(\phi_r(\theta_0))$. Hence, $v(M(\theta)) < v(\phi_r(\theta))$ for all θ, because both expressions are independent of θ. Thus, $v(\phi'_r(\theta)) = v(M(\theta))$ is constant too.

We prove items b), c) simultaneously. Suppose first that $|\lambda'_r| < |\lambda_r|$; then, the theorem of the polygon shows that $v(\phi'_r(\theta)) < v(\phi_r(\theta))$. Arguing as above, this implies $e'_r = 1$ and $\overline{\gamma'_r(\theta)} \in \iota_\theta(\mathbb{F}_r)$, with $\eta := \iota_\theta^{-1}(\overline{\gamma'_r(\theta)}) \in \mathbb{F}_r^*$ independent of θ. By [7, Thm. 3.7+Prop. 3.5], if θ runs on all the roots of

$f_{\mathbf{t}}(x)$, then $\overline{\gamma_r'(\theta)}$ runs on all the roots of the irreducible factors of $R_r'(f_{\mathbf{t}})(y)$. Hence, $R_r'(f_{\mathbf{t}})(y) \sim (y-\eta)^{a_r'}$, and $f_r'=1$.

Suppose now $\lambda_r' = \lambda_r$, so that $v(\phi_r(\theta)) = v(\phi_r'(\theta))$, by the theorem of the polygon. We distinguish two cases. If $v(M(\theta)) = v(\phi_r(\theta))$, arguing as above we get $e_r = 1$ and $\overline{\gamma_r(\theta)} = \overline{\gamma_r'(\theta)} + \iota_\theta(\eta)$, for some $\eta \in \mathbb{F}_r^*$ which is independent of θ. In this case,

$$R_r'(f_{\mathbf{t}})(y) \sim R_r(f_{\mathbf{t}})(y+\eta) \sim \psi_r(y+\eta)^{a_r}, \quad a_r = \omega_{r+1}^{\mathbf{t}}(f) > 0.$$

Hence, $R_r'(f_{\mathbf{t}})(y) \sim \psi_r'(y)^{a_r}$, where $\psi_r'(y) = \psi_r(y+\eta)$ is an irreducible polynomial of degree $f_r' = f_r$. Therefore, $f_{\mathbf{t}}(x)$ is of type $\mathbf{t}'$, so that $f_{\mathbf{t}}(x) \mid f_{\mathbf{t}'}(x)$. Since $e_r'f_r' = e_rf_r > 1$, symmetric arguments show that $f_{\mathbf{t}'}(x)$ is of type $\mathbf{t}$, and we get $f_{\mathbf{t}}(x) = f_{\mathbf{t}'}(x)$. Finally, since $m_{r+1}' = e_r'f_r'm_r = e_rf_rm_r = m_{r+1}$, we have

$$\omega_{r+1}^{\mathbf{t}}(f) = \omega_{r+1}^{\mathbf{t}}(f_{\mathbf{t}}) = \deg f_{\mathbf{t}}/m_{r+1} = \deg f_{\mathbf{t}'}/m_{r+1}' = \omega_{r+1}^{\mathbf{t}'}(f_{\mathbf{t}'}) = \omega_{r+1}^{\mathbf{t}'}(f).$$

If $v(M(\theta)) > v(\phi_r(\theta))$, then $\phi_r(\theta)^{e_r} = \phi_r'(\theta)^{e_r} + N(x)$, where $v(N(\theta)) > v(\phi_r(\theta)^{e_r})$. Arguing as above, we get $\overline{\gamma_r(\theta)} = \overline{\gamma_r'(\theta)}$, and this implies $R_r'(f_{\mathbf{t}})(y) \sim R_r(f_{\mathbf{t}})(y)$. The proof proceeds then as in the previous case. We now have $\psi_r'(y) = \psi_r(y)$. □

3.2. The process of refinement. What can be said when $e_r = f_r = 1$? Suppose we enlarge $\mathbf{t}_0$ to an order r type $\mathbf{t} = (\mathbf{t}_0; (\phi_r(x), -h_r, y-\eta))$, still dividing $f(x)$. Here h_r is a positive integer and $\eta \in \mathbb{F}_r^*$. Let $\phi_{r+1}(x)$ be a representative of $\mathbf{t}$, of degree $m_{r+1} = e_rf_rm_r = m_r$. Let us emphasize a relevant observation.

Remark 3.3. The polynomial $\phi_r'(x) := \phi_{r+1}(x)$ can be taken too as a representative of $\mathbf{t}_0$.

In fact, $\deg \phi_r' = m_{r+1} = m_r$, and $\omega_r(\phi_r') = \deg \phi_r'/m_r = 1$, because ϕ_r' is of type $\mathbf{t}_0$. We shall show that $\phi_r'(x)$ is always a better representative of $\mathbf{t}_0$ than $\phi_r(x)$; thus, in this case $\phi_r(x)$ is never optimal.

The comparison betwen these representatives uses the following plane affine transformation:

$$\mathcal{H}: \mathbb{R}^2 \longrightarrow \mathbb{R}^2, \qquad \mathcal{H}(x,y) = (x, y - h_rx).$$

The vertical lines of the plane are invariant under this transformation, and $\mathcal{H}$ acts as a translation on them. The vertical axis is pointwise invariant. Also, $\mathcal{H}$ preserves points of integer coordinates. If S is a side of negative slope λ, length ℓ, and degree d, then $\mathcal{H}(S)$ is a side of slope $\lambda - h_r$, length ℓ, and degree d.

Definition. Let h be a positive integer, $\mathbf{t}_0$ a type of order $r-1$, $\phi_r(x)$ a representative of $\mathbf{t}_0$, and $P(x) \in \mathbb{Z}[x]$ a polynomial not divisible by $\phi_r(x)$.

We define

$$(3.3)\quad \operatorname{ind}^h_{\mathbf{t}_0}(P) := \operatorname{ind}^h_{\mathbf{t}_0,\phi_r}(P) := f_0 \cdots f_{r-1}\left(\operatorname{ind}(N^h_r(P)) - \frac{1}{2}h\ell(\ell-1)\right),$$

where $N^h_r(P)$ is the part of $N_r(P)$ formed by the sides of slope less than $-h$, and ℓ is the abscissa of the right end point of $N^h_r(P)$.

This number $\operatorname{ind}^h_{\mathbf{t}_0}(P)$ is equal to $f_0 \cdots f_{r-1}$ times the number of points of integer coordinates in the region of the plane determined by the points that lie below (or on) $N^h_r(P)$, strictly above the line L_{-h} of slope $-h$ that passes through the last point of the polygon, and strictly beyond the vertical axis. The term $h\ell(\ell-1)/2$ takes care of the points of integer coordinates in the triangle determined by L_{-h}, the vertical axis and the horizontal line that passes through the last point of $N^h_r(P)$.

Let λ be a negative rational number. The λ-*component* of a Newton polygon N is the set of points (x, y) of the plane that lie on N and $y + x|\lambda|$ has a minimal value [7, Def. 1.5]. If λ is one of the slopes of N, then the λ-component coincides with the side of N of slope λ; otherwise the λ-component is a vertex of N.

Proposition 3.4. *Let $\mathbf{t}_0$ be a type of order $r-1$, and let $\phi_r(x)$ be a representative of $\mathbf{t}_0$. Let $\mathbf{t} = (\mathbf{t}_0; (\phi_r(x), -h_r, y-\eta))$ be a type of order r with $e_r = f_r = 1$, and let $\phi_{r+1}(x)$ be a representative of $\mathbf{t}$. Let $\phi'_r(x) = \phi_{r+1}(x)$ be the same polynomial considered as a representative of $\mathbf{t}_0$. Let $P(x) \in \mathbb{Z}_p[x]$. Denote by $N_{r+1}(P)$ the Newton polygon with respect to $\mathbf{t}$, $\phi_{r+1}(x)$, and denote by $N'_r(P)$ the Newton polygon with respect to $\mathbf{t}_0$, $\phi'_r(x)$. Let $\lambda = -h/e$, with h, e positive coprime integers. Denote by $R_{\lambda,r+1}(P)(y) \in \mathbb{F}_r[y]$ the residual polynomial of order $r+1$ with respect to $\mathbf{t}$, $\phi_{r+1}(x)$, λ, and denote by $R'_{\lambda,r}(P)(y) \in \mathbb{F}_r[y]$ the residual polynomial of order r with respect to $\mathbf{t}_0$, $\phi'_r(x)$, λ. Then,*

a) $(N'_r)^{h_r}(P) = \mathcal{H}(N^-_{r+1}(P))$,

b) $\operatorname{ind}_{\mathbf{t},\phi_{r+1}}(P) = \operatorname{ind}^{h_r}_{\mathbf{t}_0,\phi'_r}(P)$,

c) *Let s be the abscissa of the left end point of the λ-component of $N^-_{r+1}(P)$. There exists $\epsilon \in \mathbb{F}^*_r$ depending only on $\mathbf{t}_0$, such that*

$$R_{\lambda,r+1}(P)(y) = \epsilon^s R'_{\lambda-h_r,r}(P)(\epsilon^e y).$$

Proof. Let v_r, v_{r+1} be the p-adic valuations attached to $\mathbf{t}$. Note that v_r depends only on $\mathbf{t}_0$. Consider the ϕ_{r+1}-adic development of $P(x)$, which is simultaneously its ϕ'_r-adic development:

$$P(x) = \sum_{0\le i} a_i(x)\phi_{r+1}(x)^i = \sum_{0\le i} a_i(x)\phi'_r(x)^i.$$

For any $0 \le i$, let $u_i = v_{r+1}(a_i\phi^i_{r+1})$, $u'_i = v_r(a_i(\phi'_r)^i)$, so that the points (i, u_i) determine the Newton polygon $N_{r+1}(P)$, and the points (i, u'_i)

determine the Newton polygon $N'_r(P)$. Since $\deg a_i < m_r = m_{r+1}$, [7, Lem. 2.2+Prop. 2.7] show that $v_{r+1}(a_i) = e_r v_r(a_i) = v_r(a_i)$. By Theorem 2.1,

$$v_{r+1}(\phi_{r+1}) = e_r f_r(e_r v_r(\phi_r) + h_r) = v_r(\phi_r) + h_r = v_r(\phi'_r) + h_r.$$

This proves a), because $u_i = v_{r+1}(a_i\phi_{r+1}^i) = v_r(a_i(\phi'_r)^i) + ih_r = u'_i + ih_r$.

Item b) is an immediate consequence of a), because $\mathcal{H}$ transforms the horizontal line that passes through the last point of $N^-_{r+1}(P)$ into the line L_{-h_r} of slope $-h_r$ that passes through the last point of $N^h_r(P)$ (cf. Figure 3.1).

Let us prove c). The definition of the residual coefficients and the residual polynomials is given in [7, Defs. 2.20-2.21]. Denote $N' = (N'_r)^{h_r}(P)$. To every integer abscissa, $0 \le i \le \ell(N')$, one attaches a residual coefficient c_i of $N^-_{r+1}(P)$, and a residual coefficient c'_i of N', given by

$$c_i = \begin{cases} z_r^{t_r(i)} R_r(a_i)(z_r), & \text{if } (i, u_i) \text{ lies on } N^-_{r+1}(P), \\ 0, & \text{otherwise.} \end{cases}$$

$$c'_i = \begin{cases} z_{r-1}^{t'_{r-1}(i)} R_{r-1}(a_i)(z_{r-1}), & \text{if } (i, u'_i) \text{ lies on } N', \\ 0, & \text{otherwise.} \end{cases}$$

By a), the points (i, u_i), (i, u'_i), lying on the respective polygons have the same abscissas. Suppose that i is such an abscissa. For $j = r, r-1$, denote by $s_j(a_i)$ the abscissa of the left end point of the λ_j-component of $N_j(a_i)$ [7, Sec. 1.1]. By convention, $\ell_r = 0$, if $e_r = 1$. Since $\deg(a_i) < m_r$, the polygon $N_r(a_i)$ is the single point $(0, v_r(a_i))$. Hence, $t_r(i) := (s_r(a_i) - \ell_r u_i)/e_r = 0$. Also, $R_r(a_i)(y) = z_{r-1}^{t_{r-1}(0)} R_{r-1}(a_i)(z_{r-1}) \in \mathbb{F}_r^*$ is a constant polynomial. By definition, the exponents $t'_{r-1}(i)$, and $t_{r-1}(0)$ are given by

$$t'_{r-1}(i) = \frac{s_{r-1}(a_i) - \ell_{r-1} v_r(a_i(\phi'_r)^i)}{e_{r-1}}, \quad t_{r-1}(0) = \frac{s_{r-1}(a_i) - \ell_{r-1} v_r(a_i)}{e_{r-1}}.$$

Hence, $c_i = \epsilon^i c'_i$, where $\epsilon = (z_{r-1})^{\ell_{r-1} v_r(\phi_r)/e_{r-1}}$. Since $R_{\lambda, r+1}(P)(y) = c_s + c_{s+e} y + \cdots + c_{s+de} y^d$, and $R'_{\lambda - h_r, r}(P)(y) = c'_s + c'_{s+e} y + \cdots + c'_{s+de} y^d$, we get $R_{\lambda, r+1}(P)(y) = \epsilon^s R'_{\lambda - h_r, r}(P)(\epsilon^e y)$. □

Suppose that $\mathbf{t}_0$ divides our polynomial $f(x)$ but it is not f-complete, and the main loop of the basic algorithm applied to $\mathbf{t}_0$ computes $\mathbf{t} = (\mathbf{t}_0; (\phi_r(x), -h_r, y - \eta))$ as one of its branches. Then, Proposition 3.4 applied to $P(x) = f(x)$ shows that $\phi'_r(x)$ is a better representative of $\mathbf{t}_0$ than $\phi_r(x)$, in what the analysis of the branch $\mathbf{t}$ concerns. Theorem 3.1 and Proposition 3.4 inspire the following definition.

Definition. A type $\mathbf{t}$ of order r is called *optimal* if $m_1 < \cdots < m_r$, or equivalently, $e_i f_i > 1$, for all $1 \le i < r$.

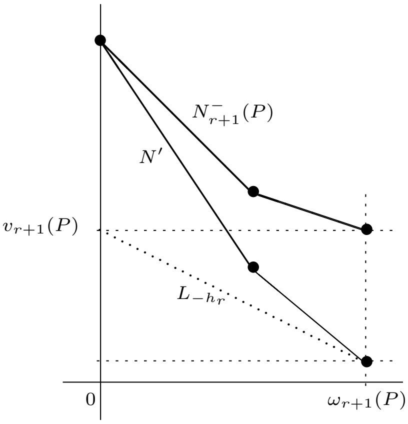

FIGURE 3.1

Actually, Proposition 3.4 proves something stronger: all information obtained by applying to $\mathbf{t}$ the basic algorithm in order $r+1$, can be already obtained at order r if $\phi'_r(x)$ is chosen as a representative of $\mathbf{t}_0$, as long as we analyze $(N'_r)^{h_r}(f)$ instead of the whole $(N'_r)^-(f)$.

This observation leads to an essential optimization of the basic algorithm. Whenever we apply the main loop to a type $\mathbf{t}_0$ and one of the outputs is a non-complete branch $\mathbf{t} = (\mathbf{t}_0; (\phi_r(x), -h_r, y-\eta))$, with $e_r = f_r = 1$, then:

(1) we compute a representative $\phi_{r+1}(x)$ of $\mathbf{t}$,
(2) we replace the pair $(\mathbf{t}, \phi_{r+1}(x))$ by the pair $(\mathbf{t}_0, \phi'_r(x))$, with $\phi'_r(x) = \phi_{r+1}(x)$,
(3) we store the cutting slope $-h_r$ as new data of level r of $\mathbf{t}_0$.
(4) when we apply the main loop to the new pair $(\mathbf{t}_0, \phi'_r(x))$, only the sides of slope less than $-h_r$ of $(N'_r)^-(f)$ will be taken into account.

We call this a *refinement step.* The algorithm that is obtained from the basic algorithm by applying a refinement step to every non-complete branch with $e_r f_r = 1$ is called *Montes' algorithm.*

In order to see that the two algorithms are equivalent we need to show that a non-refined type is equivalent to its refined replacement, as far as the future development of the algorithm is concerned. More precisely, suppose that

$$\mathbf{s} = (\mathbf{t}_0; (\phi_r(x), -h_r, y-\eta); (\phi_{r+1}(x), \lambda_{r+1}, \psi_{r+1}(y))) \tag{3.4}$$

is one of the branches of $\mathbf{t}$ obtained by the basic algorithm. In the Montes algorithm, this type is replaced by:

$$\mathbf{s}' = (\mathbf{t}_0; (\phi'_r(x), \lambda'_r, \psi'_r(y))) \tag{3.5}$$

where $\phi'_r(x) = \phi_{r+1}(x)$, $\lambda'_r = \lambda_{r+1} - h_r$, and $\psi'_r(y) = c^{f_{r+1}}\psi_{r+1}(c^{-1}y)$, for the constant $c = \epsilon^{e_{r+1}} \in \mathbb{F}_r^*$ given in Proposition 3.4.

We denote $\mathbb{F}_r := \mathbb{F}_r^{\mathbf{s}} = \mathbb{F}_r^{\mathbf{s}'} = \mathbb{F}_{r+1}^{\mathbf{s}}$, $e_{r+1} := e_{r+1}^{\mathbf{s}} = e_r^{\mathbf{s}'}$, $h_{r+1} := h_{r+1}^{\mathbf{s}}$. For the type $\mathbf{s}'$ we have $\ell_r^{\mathbf{s}'} h_r^{\mathbf{s}'} - (\ell'_r)^{\mathbf{s}'} e_r^{\mathbf{s}'} = 1$. Since $h_r^{\mathbf{s}'} = h_{r+1} - e_{r+1}h_r$, that Bézout identity can be rewritten as

$$\ell_r^{\mathbf{s}'} h_{r+1} - \left((\ell'_r)^{\mathbf{s}'} + \ell_r^{\mathbf{s}'} h_r\right) e_{r+1} = 1.$$

Therefore, we can choose $\ell_{r+1}^{\mathbf{s}} = \ell_r^{\mathbf{s}'}$, $(\ell'_{r+1})^{\mathbf{s}} = (\ell'_r)^{\mathbf{s}'} + \ell_r^{\mathbf{s}'} h_r$, as coefficients of the Bézout identity $\ell_{r+1}^{\mathbf{s}} h_{r+1} - (\ell'_{r+1})^{\mathbf{s}} e_{r+1} = 1$.

The next proposition shows that the types $\mathbf{s}, \mathbf{s}'$ are equivalent, in the sense that the basic algorithm applied to them yields completely equivalent data.

Proposition 3.5. *Let* $\mathbf{s}$, $\mathbf{s}'$ *be types as in (3.4), (3.5). Then,*

a) $v_{r+2}^{\mathbf{s}} = v_{r+1}^{\mathbf{s}'}$.

b) *The types* $\mathbf{s}$, $\mathbf{s}'$ *have the same representatives.*

Let $\phi(x) \in \mathbb{Z}[x]$ *be a representative of* $\mathbf{s}$ *(and* $\mathbf{s}'$*). For any polynomial* $P(x) \in \mathbb{Z}_p[x]$*, let* $N_{r+2}(P)$ *be the Newton polygon of order* $r+2$ *with respect to* $\mathbf{s}$ *and* $\phi(x)$*. Let* $N'_{r+1}(P)$ *be the Newton polygon of order* $r+1$ *with respect to* $\mathbf{s}'$ *and* $\phi(x)$*.*

c) $N_{r+2}(P) = N'_{r+1}(P)$ *and* $\mathrm{ind}_{\mathbf{s},\phi}(P) = \mathrm{ind}_{\mathbf{s}',\phi}(P)$.

d) $P_{\mathbf{s}}(x) = P_{\mathbf{s}'}(x)$.

Let $\lambda = -h/e$*, with* h, e *coprime positive integers. Let* $R_{\lambda,r+2}(P)(y) \in \mathbb{F}_{r+2}^{\mathbf{s}}[y]$ *be the residual polynomial of order* $r+2$ *with respect to* $\mathbf{s}$*,* $\phi(x)$*,* λ*. Let* $R'_{\lambda,r+1}(P)(y) \in \mathbb{F}_{r+1}^{\mathbf{s}'}[y]$ *be the residual polynomial of order* $r+1$*, with respect to* $\mathbf{s}'$*,* $\phi(x)$*,* λ*. Consider the* $\mathbb{F}_r$*-isomorphism*

$$\iota\colon \mathbb{F}_{r+2}^{\mathbf{s}} = \mathbb{F}_r[y]/\psi_{r+1}(y) \longrightarrow \mathbb{F}_r[y]/\psi'_r(y) = \mathbb{F}_{r+1}^{\mathbf{s}'}, \quad z_{r+1}^{\mathbf{s}} \mapsto c^{-1} z_r^{\mathbf{s}'},$$

and extend it in a natural way to the polynomial ring, $\iota\colon \mathbb{F}_{r+2}^{\mathbf{s}}[y] \to \mathbb{F}_{r+1}^{\mathbf{s}'}[y]$.

e) *Suppose we choose* $\ell_{r+1}^{\mathbf{s}} = \ell_r^{\mathbf{s}'}$*, and let* u *be the ordinate of the left end point of the* λ*-component of* $N_{r+2}(P)$*. Then, there exists a constant* $\tau \in \mathbb{F}_r^*$*, depending only on* $\mathbf{s}'$*, such that*

$$\iota(R_{\lambda,r+2}(P)(y)) = \tau^u R'_{\lambda,r+1}(P)(\tau^{-h} y).$$

Proof. We use the operators N_{r+1}, N'_r, $R_{\lambda,r+1}$, $R'_{\lambda,r}$ as defined is Proposition 3.4, and we denote $R_{r+1} := R_{\lambda_{r+1},r+1}$, $R'_r := R'_{\lambda'_r,r}$.

Let L be the line of slope λ_{r+1} that first touches $N_{r+1}^-(P)$ from below. Let $Q = (0, y)$ be the point of intersection of L with the vertical axis. By definition, $v_{r+2}^{\mathbf{s}}(P) = e_{r+1}^{\mathbf{s}} y$ [7, Def. 2.5]. Similarly, $v_{r+1}^{\mathbf{s}'}(P) = e_r^{\mathbf{s}'} y' = e_{r+1}^{\mathbf{s}} y'$, where $Q' = (0, y')$ is the point of intersection of the vertical axis with the line L' of slope $\lambda'_r = \lambda_{r+1} - h_r$ that first touches $N'_r(P)$ from below. By Proposition 3.4, $\mathcal{H}(N_{r+1}^-(P)) = (N'_r)^{h_r}(P)$; hence, $L' = \mathcal{H}(L)$.

Since the vertical axis is pointwise invariant under the action of $\mathcal{H}$, we get $Q=\mathcal{H}(Q)=Q'$, so that $y=y'$. This proves item a).

Let $\phi(x)\in\mathbb{Z}[x]$ be a representative of $\mathbf{s}$; that is, $\deg\phi=m^{\mathbf{s}}_{r+2}$ and $R_{r+1}(\phi)\sim\psi_{r+1}$ in $\mathbb{F}^{\mathbf{s}}_{r+1}[y]=\mathbb{F}_r[y]$ (Theorem 2.1). Then,

$$\deg\phi=m^{\mathbf{s}}_{r+2}=e^{\mathbf{s}}_{r+1}f^{\mathbf{s}}_{r+1}m^{\mathbf{s}}_{r+1}=e^{\mathbf{s}'}_{r}f^{\mathbf{s}'}_{r}m^{\mathbf{s}'}_{r}=m^{\mathbf{s}'}_{r+1}.$$

Also, Proposition 3.4 shows that $R'_r(\phi)\sim\psi'_r$ in $\mathbb{F}^{\mathbf{s}'}_r[y]=\mathbb{F}_r[y]$. Thus, $\phi(x)$ is a representative of $\mathbf{s}'$ too. The reciprocal statement follows by symmetric arguments. This proves item b).

Item c) is an immediate consequence of a). In order to prove d) we may assume that $P(x)$ is irreducible. Now, an irreducible polynomial $P(x)$ is of type $\mathbf{s}$ if and only if it is divisible by $\mathbf{s}$ [7, Lem. 2.4], or equivalently $N^-_{r+2}(P)$ has positive length. Analogously, $P(x)$ is of type $\mathbf{s}'$ if and only if $(N'_{r+1})^-(P)$ has positive length. Thus, d) is a consequence of c).

Consider the ϕ-adic development $P(x)=\sum_{0\le i}a_i(x)\phi(x)^i$, and denote $u_i=v^{\mathbf{s}'}_{r+1}(a_i\phi^i)=v^{\mathbf{s}}_{r+2}(a_i\phi^i)$, for all $i\ge 0$. Let $\{c_i\}_{i\ge0}$ be the residual coefficients of $N_{r+2}(P)$, and $\{c'_i\}_{i\ge0}$ the residual coefficients of $N'_{r+1}(P)$. Let i be an integer abscissa. If the point (i,u_i) lies above $N_{r+2}(P)=N'_{r+1}(P)$, we have $c_i=c'_i=0$. Suppose that (i,u_i) lies on $N_{r+2}(P)=N'_{r+1}(P)$, so that $c_ic'_i\ne0$. In this case, we have by definition,

$$c_i=(z^{\mathbf{s}}_{r+1})^{t_{r+1}(i)}R_{r+1}(a_i)(z^{\mathbf{s}}_{r+1})\in\mathbb{F}^{\mathbf{s}}_{r+2},\quad c'_i=(z^{\mathbf{s}'}_{r})^{t'_r(i)}R'_r(a_i)(z^{\mathbf{s}'}_{r})\in\mathbb{F}^{\mathbf{s}'}_{r+1}.$$

By a) of Proposition 3.4 applied to $P(x)=a_i(x)$, we have $s^{\mathbf{s}}_{r+1}(a_i)=s^{\mathbf{s}'}_r(a_i)$. By hypothesis, $\ell^{\mathbf{s}}_{r+1}=\ell^{\mathbf{s}'}_r$, and $e^{\mathbf{s}}_{r+1}=e^{\mathbf{s}'}_r$; hence, $t(i):=t_{r+1}(i)=(s^{\mathbf{s}}_{r+1}(a_i)-\ell^{\mathbf{s}}_{r+1}u_i)/e^{\mathbf{s}}_{r+1}=t'_r(i)$. Recall that $\iota(z^{\mathbf{s}}_{r+1})=c^{-1}z^{\mathbf{s}'}_r=\epsilon^{-e^{\mathbf{s}}_{r+1}}z^{\mathbf{s}'}_r$. Therefore, by c) of Proposition 3.4,

$$\iota(c_i)=\epsilon^{-e^{\mathbf{s}}_{r+1}t(i)}(z^{\mathbf{s}'}_r)^{t(i)}\epsilon^{s^{\mathbf{s}}_{r+1}(a_i)}R'_r(a_i)(z^{\mathbf{s}'}_r)=\epsilon^{\ell^{\mathbf{s}}_{r+1}u_i}c'_i=\epsilon^{\ell^{\mathbf{s}'}_r u_i}c'_i.$$

If (s,u) is the left end point of the λ-component of $N_{r+2}(P)=N'_{r+1}(P)$, and $s+de$ is the abscissa of the right end point, we have

$$\begin{aligned}R_{\lambda,r+2}(P)(y)&=c_s+c_{s+e}y+\cdots+c_{s+de}y^d,\\ R'_{\lambda,r+1}(P)(y)&=c'_s+c'_{s+e}y+\cdots+c'_{s+de}y^d.\end{aligned}$$

Since $u_{s+je}=u-jh$, we get $\iota(R_{\lambda,r+2}(P)(y))=\tau^uR'_{\lambda,r+1}(P)(\tau^{-h}y)$, for $\tau=\epsilon^{\ell^{\mathbf{s}'}_r}$. □

The refinement steps cause a strong diminution of the complexity. If we work at a higher order, we introduce new levels of recursivity in all basic tasks of the main loop; thus, if we avoid raising the order, we avoid an increase of the recursivity of the computations. For instance, suppose that along the basic algorithm we find a branch of order $r+n$, with n successive levels with $e_{r+i}f_{r+i}=1$, for $0\le i<n$,

$$\mathbf{s}=(\mathbf{t}_0;(\phi_r(x),\lambda_r,\psi_r(y));\cdots;(\phi_{r+n}(x),\lambda_{r+n},\psi_{r+n}(y))).$$

Starting with $\mathbf{t}_0$, we reach $\mathbf{s}$ by applying the main loop n times at orders $r,r+1,\dots,r+n$. If we refine, $\mathbf{s}$ collapses to $\mathbf{s}'=(\mathbf{t}_0;(\phi'_r(x),\lambda'_r,\psi'_r(x)))$, with $\phi'_r(x)=\phi_{r+n}(x)$, and $\mathbf{s}'$ is computed after applying the main loop n times too but working always at order r.

The refinement steps can also be interpreted as a search for the optimal representatives. The search is carried out by successive applications of the main loop, and the types are not enlarged till an optimal branch is found.

Summing up, Montes' algorithm has the same number of iterations as the basic algorithm but a much lower complexity. It works always at the minimum order possible till an optimal representative of each type is found, and only then it passes to work at a higher order. The output is a set $\mathbf{T}$ of f-complete and optimal types, that faithfully represents $f(x)$.

Rather surprisingly, this optimization motivated by purely computational reasons, yields an output with unexpected canonical properties. In the paper [9] we prove that the f-complete and optimal types computed by the Montes algorithm determine certain canonical invariants of each of the irreducible p-adic factors of the input polynomial $f(x)$. Also, in the paper [11] we find upper bounds for the total number of refinement steps, based on these invariants. Some of these canonical invariants had been introduced by K. Okutsu in [13].

3.3. Computation of the index with Montes' algorithm. By Theorem 2.5, $\operatorname{ind}(f)$ is obtained as the sum of all $\operatorname{ind}_{\mathbf{t}}(f)$ for $\mathbf{t}$ running on all types considered along the flow of the basic algorithm (cf. Remark 2.6).

By b) of Proposition 3.4 and c) of Proposition 3.5, $\operatorname{ind}(f)$ is obtained as well as the sum of all $\operatorname{ind}_{\mathbf{t}}^{H_r}(f)$ for $\mathbf{t}$ running on all types considered along the flow of Montes' algorithm. Here H_r is the cutting slope of $\mathbf{t}$ at its higher level (see section 5.1).

4. Generators of the prime ideals

In this section we compute generators of the prime ideals lying above p in terms of the output of Montes' algorithm: a list $\mathbf{T}=\{\mathbf{t}_{\mathfrak{p}_1},\dots,\mathbf{t}_{\mathfrak{p}_g}\}$, of f-complete optimal types, which are in 1-1 correspondence with the prime ideals $\mathfrak{p}_1,\dots,\mathfrak{p}_g$ of K dividing $p\mathbb{Z}_K$. We choose a root $\theta\in K$ of $f(x)$, and denote by $\theta_{\mathfrak{p}}\in\overline{\mathbb{Q}}_p$ the root of $f_{\mathbf{t}_{\mathfrak{p}}}(x)$, image of θ under a fixed topological embedding $K\hookrightarrow\overline{\mathbb{Q}}_p$.

If $\mathbf{t}\in\mathbf{T}$ has order zero, and $\phi(x)$ is a representative of $\mathbf{t}$, then the corresponding prime ideal is generated by $(p,\phi(x))$ by Kummer's criterion. If $\mathbf{t}\in\mathbf{T}$ has order one and its truncation of order zero, $\mathbf{t}_0=\psi_0(y)$, has

$\mathrm{ind}_{\mathbf{t}_0}(f) = 0$, then the program computes generators of the corresponding prime ideal by using Dedekind's criterion.

From now on, we fix a type $\mathbf{t}_\mathfrak{p}$, corresponding to a prime ideal $\mathfrak{p}$ not falling in those special cases. We omit the superscript $(\ )^{\mathbf{t}_\mathfrak{p}}$ for the data of $\mathbf{t}_\mathfrak{p}$. Let r be the order of $\mathbf{t}_\mathfrak{p}$, and recall that $e(\mathfrak{p}/p) = e_1 \cdots e_r$. We want to compute an integral element $\alpha_\mathfrak{p} \in \mathbb{Z}_K$ satisfying

$$v_\mathfrak{p}(\alpha_\mathfrak{p}) = 1; \quad v_\mathfrak{q}(\alpha_\mathfrak{p}) = 0,\ \forall \mathfrak{q} \mid p,\ \mathfrak{q} \neq \mathfrak{p},$$

so that the ideal $\mathfrak{p}$ is generated by p and $\alpha_\mathfrak{p}$.

Let us first construct an element $\beta_\mathfrak{p} \in K$ such that $v_\mathfrak{p}(\beta_\mathfrak{p}) = 1$. To this end we compute first a representative $\phi_{r+1}(x)$ of $\mathbf{t}_\mathfrak{p}$. Since $\mathbf{t}_\mathfrak{p}$ is f-complete, the Newton polygon $N_{r+1}(f)$, with respect to $\mathbf{t}_\mathfrak{p}$ and $\phi_{r+1}(x)$, has length equal to $\omega_{r+1}(f) = 1$; hence, it is one-sided with integer slope $-h_{r+1}$. By the theorem of the polygon and Theorem 2.1,

$$\begin{aligned} v_\mathfrak{p}(\phi_{r+1}(\theta)) &= e(\mathfrak{p}/p)v(\phi_{r+1}(\theta_\mathfrak{p})) = v_{r+1}(\phi_{r+1}) + h_{r+1} \\ &= e_r f_r(e_r v_r(\phi_r) + h_r) + h_{r+1}, \\ v_\mathfrak{p}(\phi_r(\theta)) &= e(\mathfrak{p}/p)v(\phi_r(\theta_\mathfrak{p})) = e_r(v_r(\phi_r) + (h_r/e_r)) = e_r v_r(\phi_r) + h_r. \end{aligned}$$

Therefore, the element $\beta_\mathfrak{p} := \phi_{r+1}(\theta)/\phi_r(\theta)^{e_r f_r} \in K$ has $v_\mathfrak{p}(\beta_\mathfrak{p}) = h_{r+1}$. Our aim is to find a kind of worst possible representative $\phi_{r+1}(x)$ of $\mathbf{t}_\mathfrak{p}$, satisfying $h_{r+1} = 1$. To this end, we compute a blind $\phi_{r+1}(x)$. If $h_{r+1} = 1$ we are done; if $h_{r+1} > 1$ we use a subroutine based on [7, Prop. 2.10], to construct a polynomial $P(x) \in \mathbb{Z}[x]$ with the following properties:

$$\deg P < m_{r+1}, \quad v_{r+1}(P) = v_{r+1}(\phi_{r+1}) + 1, \quad R_r(P)(y) = 1.$$

Now, $\phi'_{r+1}(x) := \phi_{r+1}(x) + P(x)$ is another representative of $\mathbf{t}_\mathfrak{p}$ and it has $h'_{r+1} = 1$, or equivalently, $v_\mathfrak{p}(\phi'_{r+1}(\theta)) = v_{r+1}(\phi_{r+1}) + 1$. In fact, $\deg \phi'_{r+1} = \deg \phi_{r+1}$ and [7, Prop. 2.8] shows that $\omega_{r+1}(\phi'_{r+1}) = \omega_{r+1}(\phi_{r+1}) = 1$; hence, ϕ'_{r+1} is a representative of $\mathbf{t}_\mathfrak{p}$. Finally, since $\deg P < m_{r+1}$, we have

$$\begin{aligned} e(\mathfrak{p}/p)v(P(\theta_\mathfrak{p})) &= v_{r+1}(P) = v_{r+1}(\phi_{r+1}) + 1 \\ &< v_{r+1}(\phi_{r+1}) + h_{r+1} = e(\mathfrak{p}/p)v(\phi_{r+1}(\theta_\mathfrak{p})). \end{aligned}$$

Thus, $v_\mathfrak{p}(\phi'_{r+1}(\theta)) = v_\mathfrak{p}(P(\theta)) = v_{r+1}(\phi_{r+1}) + 1$.

Therefore, we may assume that $\beta_\mathfrak{p} = \phi_{r+1}(\theta)/\phi_r(\theta)^{e_r f_r}$ has $v_\mathfrak{p}(\beta_\mathfrak{p}) = 1$. Our next step is to compute the values $v_\mathfrak{q}(\beta_\mathfrak{p})$, for all other prime ideals $\mathfrak{q} \neq \mathfrak{p}$ lying above p.

Definition. We say that $\mathbf{t}_\mathfrak{q}$ dominates $\mathbf{t}_\mathfrak{p}$, and we write $\mathbf{t}_\mathfrak{q} > \mathbf{t}_\mathfrak{p}$, if $\mathbf{t}_\mathfrak{q}$ is a branch of a type originated from a side of $N_{\mathbf{t}_\mathfrak{p},\phi_r}(f)$ of slope $\lambda < \lambda_r$. In this case we denote $\lambda_\mathfrak{q}^\mathfrak{p} = \lambda$ and we call it the dominating slope of $\mathbf{t}_\mathfrak{q}$ over $\mathbf{t}_\mathfrak{p}$.

The next proposition is a consequence of an explicit formula for $v_\mathfrak{q}(\phi_i(\theta))$ for all $\mathfrak{q}$ and all $1 \le i \le r+1$, that can be found in [10, Prop. 3.8].

Proposition 4.1. *Let $\mathfrak{q}$ be a prime ideal of K lying above p, $\mathfrak{q} \neq \mathfrak{p}$. Let s be the order of $\mathbf{t}_\mathfrak{q}$. Then,*

$$v_\mathfrak{q}(\beta_\mathfrak{p}) = \begin{cases} e_r f_r (e_r^{\mathbf{t}_\mathfrak{q}} \cdots e_s^{\mathbf{t}_\mathfrak{q}})(\lambda_\mathfrak{q}^\mathfrak{p} - \lambda_r) < 0, & \textit{if } \mathbf{t}_\mathfrak{q} > \mathbf{t}_\mathfrak{p}, \\ 0, & \textit{otherwise} . \end{cases}$$

Now, for the maximal types with respect to the ordering "$>$", we take $\alpha_\mathfrak{p} := \beta_\mathfrak{p}$. For the rest of the types we compute recurrently:

$$\alpha_\mathfrak{p} := \beta_\mathfrak{p} \prod\nolimits_{\mathbf{t}_\mathfrak{q} > \mathbf{t}_\mathfrak{p}} \alpha_\mathfrak{q}^{-v_\mathfrak{q}(\beta_\mathfrak{p})}.$$

These elements are not far from generating the $\mathfrak{p}_i$, since:

$$v_\mathfrak{q}(\alpha_\mathfrak{p}) = \begin{cases} 1 & \text{if } \mathfrak{q} = \mathfrak{p}, \\ 0 & \text{otherwise.} \end{cases}$$

Unfortunately, they could be non-integral at primes of $\mathbb{Z}_K$ not dividing $p\,\mathbb{Z}_K$. This can be easily arranged; we write each $\alpha_\mathfrak{p}$ in the form $\alpha_\mathfrak{p} = g(\theta)/b$, with $g(x) \in \mathbb{Z}[x]$ and $b \in \mathbb{Z}$ coprime with the content of $g(x)$. Then, we modify $\alpha_\mathfrak{p}$ by taking $\alpha_\mathfrak{p} := g(\theta)/p^{v(b)}$.

The complexity of the computation of $\alpha_\mathfrak{p}$ is dominated by the inversion of $\phi_r(\theta)$ in K, which may be a hard task if the degree of $\phi_r(x)$ is large ot it has large coefficients. In the forthcoming paper [10] we present an algorithm to compute generators of the prime ideals which requires no inversions in K.

5. Computational issues

We recall that Montes' algorithm is the optimization of the basic algorithm of section 2.2 as a result of the application of the refinement process of section 3.2, with the intention of dealing only with optimal types. In this section we give a more detailed description of this algorithm and we discuss some computational aspects.

5.1. Outline of the algorithm. The goal of Montes' algorithm is the computation of $\operatorname{ind}(f)$ and the construction of a set $\mathbf{T}$ of f-complete optimal types, that faithfully represents $f(x)$.

By the recursive nature of its construction, many of the types generated by the algorithm will share many of their levels, so that most of the computations necessary to enlarge them will be the same. Hence it is convenient to organize their computation in such a way that we can take profit of as much previous computations as possible. The simplest way to organize the computation of types is to store all non-complete types being built by the algorithm in a list, which we call `STACK`. Complete types are stored in a second list called `COMPLETETYPES`.

The variable `STACK`, as its name suggests, is a LIFO stack, which in practice determines the flow of the algorithm: the main loop of the algorithm extracts the last type from the `STACK` and works it out to produce a number of enlarged types. The complete ones are added to `COMPLETETYPES` and the non-complete ones are added to the top of the `STACK`. The program finishes when the `STACK` is empty.

The types are built along the algorithm as the branches of a tree. The root of the tree is a node corresponding to a type of order 0; that is, an irreducible factor modulo p of the input polynomial $f(x)$. Every division of the branch into new subbranches is generated by multiple sides of a Newton polygon of $f(x)$ and by multiple irreducible factors of the residual polynomial attached to each side.

In order to homogenize the flow of Montes' algorithm, we introduce data ϕ_r, H_r, ω_r at the r-th level of each type $\mathbf{t}_0$ of order $r-1$. The variable ϕ_r stores a representative of $\mathbf{t}_0$. The variable H_r stores the absolute value of the *cutting slope*; it tells us that we must compute only the Newton polygon $N_r^{H_r}(f)$ gathering the sides of slope less than $-H_r$ of $N_r(f)$. Finally, ω_r stores the length of $N_r^{H_r}(f)$. By [7, Lem. 2.17] this length can be precomputed as $\omega_r = \operatorname{ord}_{\psi_{r-1}} R_{r-1}(f)$ if $\mathbf{t}_0$ is non-refined ($H_r = 0$ and $N_r^{H_r}(f) = N_r^-(f)$), or as $\omega_r = \omega_{r+1}^{\mathbf{t}}$ if $\mathbf{t}_0$ is the refinement of an order r type $\mathbf{t}$ (by Proposition 3.4).

In this way, the main loop does not need to distinguish between refined and non-refined types. At the input of a non-complete type $\mathbf{t}_0$ of order $r-1$, having all these data at level r, we compute only the first ω_r+1 coefficients $a_0(x), \dots, a_{\omega_r}(x)$ of the ϕ_r-development of $f(x)$; then, the Newton polygon of the cloud of points $(i, v_r(a_i))$, $0 \le i \le \omega_r$, is already $N_r^{H_r}(f)$.

In the main loop, when we deal with a non-complete order r branch $\mathbf{t}$ of $\mathbf{t}_0$, we compute first a representative $\phi(x)$ of $\mathbf{t}$. Then, if $e_r f_r > 1$, we assign

$$\phi_{r+1}^{\mathbf{t}} \leftarrow \phi(x), \quad H_{r+1}^{\mathbf{t}} \leftarrow 0, \quad \omega_{r+1}^{\mathbf{t}} \leftarrow \operatorname{ord}_{\psi_r} R_r(f),$$

and the order r type $\mathbf{t}$ is added to the `STACK`. On the other hand, if $e_r f_r = 1$, we take a copy $\mathbf{t}'$ of $\mathbf{t}_0$ of order $r-1$, we assign

$$\phi_r^{\mathbf{t}'} \leftarrow \phi(x), \quad H_r^{\mathbf{t}'} \leftarrow |\lambda_r^{\mathbf{t}}|, \quad \omega_r^{\mathbf{t}'} \leftarrow \omega_{r+1}^{\mathbf{t}},$$

and we add $\mathbf{t}'$ to the `STACK`.

Every time we compute a Newton polygon with respect to a type $\mathbf{t}$ of order $r-1$, we add the number $\operatorname{ind}_{\mathbf{t}}^{H_r}(f)$, given by the formula (3.3), to the variable `TOTALINDEX`. The output value of this variable is $\operatorname{ind}(f)$, as explained in section 3.3.

We now give a detailed outline of Montes' algorithm, using standard pseudo-code.

MONTES' ALGORITHM

INPUT:

– A monic irreducible polynomial $f(x) \in \mathbb{Z}[x]$.

– A prime number p.

INITIALIZATION STEPS

```
1 Factor $f(y) \pmod{p} = \prod_\varphi \varphi(y)^{a_\varphi}$, into a product of powers of pairwise different monic irreducible polynomials $\varphi(y) \in \mathbb{F}[y]$.
2 Initialize empty lists STACK and COMPLETETYPES. Set TOTALINDEX$\leftarrow 0$.
3 FOR every polynomial $\varphi(y)$ do
    4 Create a type $\mathbf{t}$ of order zero with $\psi_0(y)^{\mathbf{t}} \leftarrow \varphi(y)$. Set $\omega_1^{\mathbf{t}} \leftarrow a_\varphi$.
    5 Take a monic $g(x) \in \mathbb{Z}[x]$ such that $g(y) \pmod{p} = \varphi(y)$.
      Set $\phi_1(x)^{\mathbf{t}} \leftarrow g(x)$.
    6 If $a_\varphi = 1$ add $\mathbf{t}$ to COMPLETETYPES.
      Otherwise, set $H_1^{\mathbf{t}} \leftarrow 0$ and add $\mathbf{t}$ to STACK.
```

MAIN LOOP

```
WHILE the STACK is non-empty do:
1 Extract the last type $\mathbf{t}_0$ from STACK. Let $r-1 \ge 0$ be its order.
2 Compute the first $\omega_r + 1$ coefficients $a_0(x), \dots, a_{\omega_r}(x)$ of the $\phi_r$-adic expansion of $f(x)$, and compute the Newton polygon $N$ of the cloud of points $(i, v_r(a_i(\phi_r)^i))$, for $0 \le i \le \omega_r$.
3 Compute $\operatorname{ind}_{\mathbf{t}_0}^{H_r}(f)$ by using (3.3), and add this number to TOTALINDEX.
4 FOR every side $S$ of $N$ do
    5 Set $\lambda_r^{\mathbf{t}_0} \leftarrow$ slope of $S$. Compute and factorize the $r$-th order residual polynomial $R_r(f)(y) \in \mathbb{F}_r[y]$.
    6 FOR every monic irreducible factor $\psi(y)$ of $R_r(f)(y)$ do
        7 Make a copy $\mathbf{t}$ of the type $\mathbf{t}_0$, and extend it to order $r$ by setting
              $\psi_r^{\mathbf{t}}(y) \leftarrow \psi(y), \quad \omega_{r+1}^{\mathbf{t}} \leftarrow \operatorname{ord}_\psi R_r(f)$.
        8 Compute a representative $\phi(x) \in \mathbb{Z}[x]$ of $\mathbf{t}$. Set $\phi_{r+1}^{\mathbf{t}}(x) \leftarrow \phi(x)$.
        9 If $\omega_{r+1}^{\mathbf{t}} = 1$ (the type is complete), add $\mathbf{t}$ to COMPLETETYPES and continue to the next factor $\psi(y)$.
        10 If $\deg \psi = 1$ and $\lambda_r^{\mathbf{t}} \in \mathbb{Z}$ (the type must be refined), make a copy $\mathbf{t}'$ of the type $\mathbf{t}_0$, set
              $\phi_r^{\mathbf{t}'}(x) \leftarrow \phi(x), \quad H_r^{\mathbf{t}'} \leftarrow |\lambda_r^{\mathbf{t}}|, \quad \omega_r^{\mathbf{t}'} \leftarrow \omega_{r+1}^{\mathbf{t}},$
           add $\mathbf{t}'$ to the STACK and continue to the next factor $\psi(y)$.
        11 (Store a higher order type) Set $H_{r+1}^{\mathbf{t}} \leftarrow 0$, add $\mathbf{t}$ to the STACK and continue to the next factor $\psi(y)$.
```

OUTPUT:

– The p-valuation of the index of $f(x)$ in $\mathbb{Z}_K$ is the value of TOTALINDEX.

– The list $\{\mathbf{t}_1,\dots,\mathbf{t}_g\}$ of `COMPLETETYPES` is in 1-1 correspondence with the prime ideals of K dividing p.

For every output type $\mathbf{t}$, the ramification index and residual degree of the corresponding ideal $\mathfrak{p}$ are given by:

$$e(\mathfrak{p}/p)=e_1^{\mathbf{t}}\cdots e_r^{\mathbf{t}},\quad f(\mathfrak{p}/p)=m_1^{\mathbf{t}}f_1^{\mathbf{t}}\cdots f_r^{\mathbf{t}},$$

where r is the order of $\mathbf{t}$. Also, the finite field $\mathbb{F}_{r+1}^{\mathbf{t}}$ is a computational representation of the residue field $\mathbb{Z}_K/\mathfrak{p}$. Moreover, the polynomial $\phi_{r+1}^{\mathbf{t}}(x)$ is an Okutsu approximation to the p-adic irreducible factor of $f(x)$ canonically attached to $\mathfrak{p}$ [9].

5.2. Some examples.

Example 1. Let us consider the irreducible polynomial

$$\begin{aligned}f(x):=x^{12}&-588x^{10}+476x^9+130095x^8-172872x^7-12522636x^6\\&+24745392x^5+486721116x^4-1583408736x^3-641009376x^2\\&+10978063488x+59914669248,\end{aligned}$$

whose discriminant is

$$\operatorname{disc}(f)=2^{84}\cdot 3^{64}\cdot 7^{52}\cdot 79^4\cdot 14159^2\cdot 644173^2\cdot 3352073^2\,.$$

We apply the algorithm to find the decomposition of the prime $p=2$ in the ring of integers $\mathbb{Z}_K$ of the number field $K=\mathbb{Q}(\theta)$ generated by any root of the polynomial $f(x)$. Since

$$f(y)\equiv y^8\,(y+1)^4\pmod 2,$$

we find two types of order zero dividing $f(x)$: $\mathbf{t}_0=y$, $\mathbf{t}_1=y+1$, with representatives $\phi_1^{\mathbf{t}_0}(x)=x$, $\phi_1^{\mathbf{t}_1}(x)=x+1$, respectively. The Newton polygon $N_1^{\mathbf{t}_1}(f)$ has two sides, with slopes $-3/2$ and $-1/2$ respectively, which single out two prime ideals $\mathfrak{p}_1,\mathfrak{p}_2$, with $e(\mathfrak{p}_1/2)=e(\mathfrak{p}_2/2)=2$ and $f(\mathfrak{p}_1/2)=f(\mathfrak{p}_2/2)=1$. We can take $\phi_2^{\mathbf{t}_{\mathfrak{p}_1}}=(x+1)^2+8$, $\phi_2^{\mathbf{t}_{\mathfrak{p}_2}}=(x+1)^2+2$ as representatives of these complete types. These polynomials are Okutsu approximations to the p-adic irreducible factors of $f(x)$ canonically attached to $\mathfrak{p}_1$, $\mathfrak{p}_2$, respectively.

The Newton polygon $N_1^{\mathbf{t}_0}(f)$ has again two sides with slopes -1, $-1/2$, and residual polynomials $R_{-1,1}(f)=(y+1)^4$, $R_{-1/2,1}(f)=(y+1)^2$, respectively. Hence, the type $\mathbf{t}_0$ extends to two non-complete types of order one: $\mathbf{t}'=(y;(x,-1/2,y+1))$, $\mathbf{t}''=(y;(x,-1,y+1))$. The type $\mathbf{t}'$ admits x^2+2 as a representative, and it is ready to be enlarged to an order 2 type. The type $\mathbf{t}''$ admits $x+2$ as a representative, and it must be refined; to this end, we take $x+2$ as a new representative of $\mathbf{t}_0$ and a cutting slope $H_1=1$. The Newton polygon of $f(x)$ with respect to $x+2$ has only one side with slope smaller than -1; the slope is $-3/2$ and the residual polynomial $(y+1)^2$; hence, this type admits $(x+2)^2+8$ as a representative, and it may be enlarged too to an order 2 type. Now, we have two types of order one, ready to be enlarged to order 2:

$$\begin{aligned} \mathbf{t} &= (y; (x+2, -3/2, y+1)), & \phi_2^{\mathbf{t}}(x) &= (x+2)^2+8, \\ \mathbf{t}' &= (y; (x, -1/2, y+1)), & \phi_2^{\mathbf{t}'}(x) &= x^2+2. \end{aligned}$$

The Newton polygon $(N_2^{\mathbf{t}})^-(f)$ has a unique side with slope -4 and residual polynomial $(y+1)^2$, so that this type must be refined. We take, for instance, $\phi_2^{\mathbf{t}}(x) = (x+2)^2+40$ and cutting slope $H_2^{\mathbf{t}} = 4$. The new polygon $(N_2^{\mathbf{t}})^4(f)$ has two sides of length one, with slopes -9 and -5; thus, it determines two new prime ideals $\mathfrak{p}_3, \mathfrak{p}_4$ dividing $2\mathbb{Z}_K$, with $e(\mathfrak{p}_3/2) = e(\mathfrak{p}_4/2) = 2$, $f(\mathfrak{p}_3/2) = e(\mathfrak{p}_4/2) = 1$. Their associate complete types are:

$$\begin{aligned} \mathbf{t}_{\mathfrak{p}_3} &= (y; (x+2, -3/2, y+1); ((x+2)^2+40, -9, y+1)), \\ \mathbf{t}_{\mathfrak{p}_4} &= (y; (x+2, -3/2, y+1); ((x+2)^2+40, -5, y+1)). \end{aligned}$$

with representatives $\phi_3^{\mathbf{t}_{\mathfrak{p}_3}}(x) = (x+2)^2+64(x+2)+40$, $\phi_3^{\mathbf{t}_{\mathfrak{p}_4}}(x) = (x+2)^2+16(x+2)+40$, respectively.

The Newton polygon $(N_2^{\mathbf{t}'})^-(f)$ has a unique side with slope -4 and residual polynomial $(y+1)^2$; thus, we must refine. Take $\phi_2^{\mathbf{t}'}(x) = x^2+10$, $H_2 = 4$. The next Newton polygon has again a unique side with slope -5 and residual polynomial $(y+1)^2$. We refine again, taking $\phi_2^{\mathbf{t}'}(x) = x^2+10+8x$. This representative leads to a new $(N_2^{\mathbf{t}'})^-(f)$ having two sides of length one with slopes -8 and -7; thus it singles out two prime ideals $\mathfrak{p}_5, \mathfrak{p}_6$, with $e(\mathfrak{p}_5/2) = e(\mathfrak{p}_6/2) = 2$, $f(\mathfrak{p}_5/2) = e(\mathfrak{p}_6/2) = 1$. Their associate complete types are:

$$\begin{aligned} \mathbf{t}_{\mathfrak{p}_5} &= (y; (x, -1/2, y+1); (x^2+8x+10, -8, y+1)), \\ \mathbf{t}_{\mathfrak{p}_6} &= (y; (x, -1/2, y+1); (x^2+8x+10, -7, y+1)). \end{aligned}$$

with representatives $\phi_3^{\mathbf{t}_{\mathfrak{p}_5}}(x) = x^2+8x+42$, $\phi_3^{\mathbf{t}_{\mathfrak{p}_6}}(x) = x^2+24x+10$, respectively.

Summing up, $2\mathbb{Z}_K = (\mathfrak{p}_1 \cdots \mathfrak{p}_6)^2$. The 2-index of $f(x)$ is $\operatorname{ind}_2(f) = 33$, and $v(\operatorname{disc}(K)) = 18$.

Example 2. Take $p = 2$ and consider the irreducible polynomial

$$f(x) := (x^3+x+5)^{50} + 2^{89}(x^3+x+5)^{25} + 2^{178}.$$

The algorithm takes initially $\phi_1(x) = x^3+x+1$, and finds a unique side with slope -2 and residual polynomial $(y+1)^{50}$. A refinement leads to $\phi_1(x) = x^3+x+5$, and a Newton polygon with one side, with slope $-89/25$ and irreducible residual polynomial y^2+y+1. Hence, in the number field K defined by any root of $f(x)$, we have $2\mathbb{Z}_K = \mathfrak{p}^{25}$, with $f(\mathfrak{p}/2) = 6$. The 2-index of the polynomial is 13011. While this computation is almost instantaneous, the determination with Pari of a 2-integral basis of K takes about 190 seconds, and needs an amount of 244 Mb of memory.

5.3. Some remarks on the complexity. The complexity of the Montes algorithm has been analyzed by Ford-Veres [6] and Pauli [14]. These results lead to an estimation of $O(n^{2+\epsilon}v_p(\mathrm{disc}(f))^{2+\epsilon})$ multiplications of integers less than p, if we assume fast multiplication.

In practice the algorithm has an excellent performance. We provide some arguments to explain its good practical behaviour.

We saw in section 2.3 that the number of iteration of the main loop is bounded by $\mathrm{ind}(f)$ and that each iteration covers $\mathrm{ind}_{\mathbf{t}}(f)$ steps from the total value of $\mathrm{ind}(f)$. Now, in practice $\mathrm{ind}_{\mathbf{t}}(f)$ is usually much bigger than one in each iteration, as mentioned in Remark 2.8. On the other hand, the iterations of the main loop are more expensive for high order types than for low order types. However, this is balanced by the following fact: the higher is the order, the smaller is $\omega_{r+1}(f)$, and this invariant tells the number of coefficients of the ϕ_r-adic development of $f(x)$ that must be computed, and it is an upper bound for the degrees of the residual polynomials.

Finally, the degree of the polynomials $\phi_k^{\mathbf{t}}(x)$ appearing in an f-complete type $\mathbf{t}$ is a divisor of the product $e(\mathfrak{p}_{\mathbf{t}}/p)f(\mathfrak{p}_{\mathbf{t}}/p)$. Every time the type is enlarged, the degree of the new $\phi_{k+1}^{\mathbf{t}}(x)$ is multiplied by $e_k^{\mathbf{t}}f_k^{\mathbf{t}} > 1$. Hence, if a polynomial $f(x)$ is divisible by a type of high order, its degree must be really huge. This explains why the algorithm works well for polynomials of high degree: the maximum of the orders of the types of a polynomial grows slowly in comparison with the degree.

The low memory requirement of the algorithm is another of its strong advantages: it is only necessary to store the levels $(\phi_k(x), \lambda_k, \psi_k(x))$ of the different types. This makes possible the treatment of polynomials of very high degree with scarce computational resources.

5.4. Implementation of the algorithm. The first implementation of Montes' algorithm was programmed by the first author in 1997 as a part of his Ph.D. It was written for Mathematica 3.0, and it included a specific package to work with finite fields. It is still available on request to the author. Ten years later, we started a collaboration to make a full upgrade of the algorithm, with many optimizations both theoretical and computational, including a completely new implementation in Magma.

The main data type used by the program is a specifically designed record which contain all the relevant data of a type at a given order. To avoid massive replication of the types being computed, most of the routines access them by memory address.

The program is included in a package that contains routines to construct types, and polynomials with prescribed attached types. The package and its documentation can be downloaded from the site

`http://themontesproject.blogspot.com`

which also contains updated information and further applications.

6. Running times

We devote this section to illustrate the performance of (our implementation of) Montes' algorithm with several polynomials chosen to force its capabilities at maximum in three directions: polynomials with a unique associate type of large order, polynomials which require a lot of refinements, and polynomials with many different types. We have also included some test polynomials found in the literature.

The computations in these examples have been done with Magma v2.15-11 in a Linux server, with two Intel Quad Core processors, running at 3.0 Ghz, with 32Gb of RAM memory.

All running times are expressed in seconds.

Example 1. Take $p=2$. Consider the irreducible polynomials

$$\begin{aligned}
&\phi_1 = x^2+2^2x+2^4,\\
&\phi_2 = \phi_1^2+2^4x\phi_1+2^{12},\\
&\phi_3 = \phi_2^4+2^{23}(x+2^2)\phi_2^2+2^{42}x\phi_1,\\
&\phi_4 = \phi_3^2+2^{12}x\phi_2^3\phi_3+2^{72}\phi_1\phi_2^2+2^{101}x,\\
&\phi_5 = \phi_4^3+2^{34}\phi_1\phi_2\phi_3\phi_4^2\\
&\qquad\quad +2^{215}((x(\phi_1+2^6)(\phi_2^3+2^{25}\phi_2)+2^{27}\phi_2^2)\phi_3+2^{64}(x\phi_1\phi_2^2+2^{33})),\\
&\phi_6 = \phi_5^6+2^{883}x\phi_3\phi_5^3+2^{1736}((x+4)\phi_1+2^8)\phi_2^2\phi_4,\\
&\phi_7 = \phi_6^2+2^{2810}\phi_5^3.
\end{aligned}$$

For each j, the polynomial ϕ_j has a unique associate complete type of order j, so that in the corresponding number field K_j the ideal $2\mathbb{Z}_{K_j}$ is the power of a unique prime ideal $\mathfrak{p}_j$. The following table contains the degree and 2-index of ϕ_j, the ramification index e_j and residual degree f_j of $\mathfrak{p}_j$, and the time $\mathtt{t}$ used by the program to compute them.

ϕ_j	$\deg\phi_j$	$\operatorname{ind}(\phi_j)$	e_j	f_j	$\mathtt{t}$
ϕ_1	2	2	1	2	0.00
ϕ_2	4	16	1	4	0.00
ϕ_3	16	360	2	8	0.00
ϕ_4	32	1544	2	16	0.01
ϕ_5	96	14616	2	48	0.03
ϕ_6	576	537120	6	96	0.31
ϕ_7	1152	2153184	12	96	1.47

Example 2. Let $f_k(x)=(x^2+x+1)^2-p^{2k+1}$, with $p\equiv 1 \pmod 3$ a prime number. Montes' algorithm finds two types of order zero dividing $f_k(x)$, with liftings $\phi_1(x)\in\mathbb{Z}[x]$ of degree one. For both of them the Newton polygon of the first order is one-sided of slope -1, it has end points $(2,0)$

and $(0, 2)$, and the residual polynomial is the square of a linear factor. After approximately $2k$ refinements, both types become f_k-complete. The ideal $p\,\mathbb{Z}_K$ splits as the product of two prime ideals with ramification index 2 and residual degree 1, and the p-index of $f_k(x)$ is $2k$.

This is almost the illest-conditioned quartic polynomial for the algorithm. The index of every type is increased a unit per refinement in general, and the total p-index of $f_k(x)$ is $2k$. Thus, about $2k$ iterations of the main loop are required. In the following table we show the running time of the programm for different values of k and p.

p	ind(f_k)	`t`	p	ind(f_k)	`t`
7	1000	0.29	43	10000	35.06
7	2000	0.68	103	10000	48.06
7	4000	1.94	1009	1000	0.52
7	8000	7.59	1009	2000	1.88
7	16000	37.46	1009	4000	9.00
7	20000	65.51	$10^9 + 9$	1000	1.51
13	1000	0.32	$10^9 + 9$	2000	7.97
13	2000	0.81	$10^9 + 9$	4000	46.18
13	10000	19.23	$10^{69} + 9$	100	0.80
19	10000	23.65	$10^{69} + 9$	200	1.74
31	10000	30.84	$10^{69} + 9$	400	5.30
37	10000	33.02	$10^{69} + 9$	1000	37.34

Example 3. Take $p = 13$. We now consider a polynomial with several different types. Let

$$\begin{aligned}
&\phi_1(x) = x^2 + 13^2x + 13^4 \cdot 3;\\
&\phi_2(x) = \phi_1(x)^3 + 13^{18} \cdot 2;\\
&\phi_3(x) = \phi_2(x)^{10} + 13^{89}(x + 13^2)\phi_2(x)^5 + 13^{176}\phi_1(x);\\
&\phi_4(x) = \phi_3(x)^2 - 13^{248}((x + 13^2)\phi_1(x) + 13^8)\phi_2(x)^6 - 13^{335}\phi_1(x)^2\phi_2(x);\\
&f_j(x) = \prod\nolimits_{k=0}^{j} \phi_4(x + k) + 13^{5000}, \qquad j = 0, \dots, 12.
\end{aligned}$$

In the number field K_j defined by $f_j(x)$, we have the factorization

$$13\,\mathbb{Z}_L = \mathfrak{p}_1^5 \cdots \mathfrak{p}_j^5, \qquad f(\mathfrak{p}_1/13) = \cdots = f(\mathfrak{p}_j/13) = 24.$$

Each prime ideal corresponds to an order 4 type. The 13-index of $f_j(x)$ is $21576j$. In this example, the addition of more factors to the product defining the $f_j(x)$ implies a significant growing in the size of the coefficients of the polynomial, which has a certain impact in the running times of the algorithm, shown in the table below.

j	$\deg f_j$	$\mathrm{ind}(f_j)$	t
0	120	21576	0.06
1	240	43152	0.13
2	360	64728	0.33
3	480	86304	0.70
4	600	107880	1.29
5	720	129456	2.16
6	840	151032	3.50
7	960	172608	5.19
8	1080	194184	7.38
9	1200	215760	10.43
10	1320	237336	14.00
11	1440	258912	18.52
12	1560	280488	23.97

Example 4. We applied the algorithm to the 32 polynomials $f_1, \dots, f_{32}$ appearing in [5, appendix D]. The total running time for altogether was less than 0.2 seconds. We then applied the algorithm to the polynomials $F_i = f_i^2 + p_i^{1000}$, where p_i is the prime specified in *loc.cit.* for every polynomial. In the table below we display the p_i-index of these polynomials and the running times of the algorithm.

f	p	$\mathrm{ind}(f)$	t	f	p	$\mathrm{ind}(f)$	t
F_1	2	4510	1.09	F_{17}	2	7016	1.00
F_2	2	4507	0.84	F_{18}	7	7002	14.39
F_3	3	4502	2.05	F_{19}	71	7502	172.85
F_4	3	5006	1.19	F_{20}	3	7510	3.95
F_5	2	5005	0.43	F_{21}	5	7502	29.81
F_6	2	5009	0.66	F_{22}	3	7500	4.25
F_7	2	5510	1.28	F_{23}	3	7510	4.09
F_8	5	5502	83.99	F_{24}	2	8072	1.62
F_9	2	5514	0.70	F_{25}	47	10520	41.33
F_{10}	1289	6002	173.79	F_{26}	61	6090	73.85
F_{11}	2	6014	1.03	F_{27}	2	8084	2.44
F_{12}	3	6000	5.12	F_{28}	3	8152	1.75
F_{13}	11	6502	82.86	F_{29}	3	12040	5.10
F_{14}	17	6502	58.45	F_{30}	2	16156	6.83
F_{15}	2	6527	2.27	F_{31}	2	16476	15.60
F_{16}	2	7009	1.59	F_{32}	2	20204	18.22

References

[1] J.A. Buchmann, H.W. Lenstra, *Approximating ring of integers in number fields.* J. Théorie des Nombres de Bordeaux **6** (1994), no. 2, 221–260.

[2] H. Cohen, *A course in Computational Number Theory.* Graduate Texts in Mathematics, vol. **138**, Springer V., Berlin, 2000, fourth printing.

[3] D. Ford, *The construction of maximal orders over a Dedekind domain.* Journal of Symbolic Computation **4** (1987), 69–75.

[4] D. Ford, P. Letard, *Implementing the Round Four maximal order algorithm.* J. Théorie des Nombres de Bordeaux **6** (1994), no. 1, 39–80.

[5] D. Ford, S. Pauli, X. Roblot, *A fast algorithm for polynomial factorization over $\mathbb{Q}_p$.* J. Théorie des Nombres de Bordeaux **14** (2002), no. 1, 151–169.

[6] D. Ford, O. Veres, *On the Complexity of the Montes Ideal Factorization Algorithm.* In G. Hanrot and F. Morain and E. Thomé, Algorithmic Number Theory, 9th International Symposium, ANTS-IX, Nancy, France, July 19-23, 2010, LNCS, Springer Verlag 2010.

[7] J. Guàrdia, J. Montes, E. Nart, *Newton polygons of higher order in algebraic number theory.* Trans. Amer. Math. Soc. **364** (2012), no. 1, 361–416.

[8] J. Guàrdia, J. Montes, E. Nart, *Higher Newton polygons and integral bases.* ArXiv: 0902.3428v2 [math.NT].

[9] J. Guàrdia, J. Montes, E. Nart, *Okutsu invariants and Newton polygons.* Acta Arithmetica **145** (2010), 83–108.

[10] J. Guàrdia, J. Montes, E. Nart, *A new computational approach to ideal theory in number fields.* ArXiv:1005.1156v3 [math.NT].

[11] J. Guàrdia, E. Nart, S. Pauli, *Single-factor lifting and factorization of polynomials over local fields.* Journal of Symbolic Computation, to appear, arXiv:1104.3181v1 [math.NT].

[12] J. Montes, *Polígonos de Newton de orden superior y aplicaciones aritméticas.* Tesi Doctoral, Universitat de Barcelona 1999.

[13] K. Okutsu, *Construction of integral basis I, II.* Proc. Japan Acad. **58** (1982), Ser. A, 47–49, 87–89.

[14] S. Pauli, *Factoring polynomials over local fields, II.* In G. Hanrot and F. Morain and E. Thomé, Algorithmic Number Theory, 9th International Symposium, ANTS-IX, Nancy, France, July 19-23, 2010, LNCS, Springer Verlag 2010.

[15] H. Zassenhaus, *Ein Algorithmus zur Berechnung einer Minimalbasis über gegebener Ordnung.* Funktionalanalysis, Approximationstheorie, Numerische Mathematik (Oberwolfach, 1965), Birkhäuser, Basel, 1967, 90–103.

[16] H. Zassenhaus, *On the Second Round or the Maximal Order Program.* In Applications of number theory to numerical analysis, Academic Press, New York, 1972, 398–431.

Jordi Guàrdia
Departament de Matemàtica Aplicada IV
Escola Politècnica Superior d'Enginyera de Vilanova i la Geltrú
Av. Víctor Balaguer s/n.
E-08800 Vilanova i la Geltrú, Catalonia
E-mail: guardia@ma4.upc.edu

Jesús Montes
Departament de Ciències Econòmiques i Empresarials
Facultat de Ciències Socials, Universitat Abat Oliba CEU
Bellesguard 30
E-08022 Barcelona, Catalonia, Spain
Departament de Matemàtica Econòmica, Financera i Actuarial
Facultat d'Economia i Empresa, Universitat de Barcelona
Av. Diagonal 690
E-08034 Barcelona, Catalonia, Spain
E-mail: montes3@uao.es,jesus.montes@ub.edu

Enric Nart
Departament de Matemàtiques
Universitat Autònoma de Barcelona
Edifici C
E-08193 Bellaterra, Barcelona, Catalonia, Spain
E-mail: nart@mat.uab.cat