Journal de Théorie des Nombres
de Bordeaux **10** (1998), 243–271

On Computing Subfields. A Detailed Description of the Algorithm

par Jürgen KLÜNERS

Résumé. Soit $\mathbb{Q}(\alpha)$ un corps de nombres défini par le polynôme minimal de α. Nous nous intéressons à déterminer les sous-corps $\mathbb{Q}(\beta) \subset \mathbb{Q}(\alpha)$ de degré donné. Chaque sous-corps est décrit en donnant le polynôme minimal g de β et le plongement de β dans $\mathbb{Q}(\alpha)$ donné par un polynôme h tel que $h(\alpha) = \beta$. Il y a une bijection entre les systèmes de blocs du groupe de Galois de f et les sous-corps de $\mathbb{Q}(\alpha)$. Ces systèmes de blocs sont calculés en utilisant les sous-groupes cycliques du groupe de Galois qui sont obtenus à partir du critère de Dedekind. Lorsqu'un système de blocs est connu, on calcule le sous-corps correspondants par des méthodes p-adiques. Nous présentons ici une description détaillée de l'algorithme.

Abstract. Let $\mathbb{Q}(\alpha)$ be an algebraic number field given by the minimal polynomial f of α. We want to determine all subfields $\mathbb{Q}(\beta) \subset \mathbb{Q}(\alpha)$ of given degree. It is convenient to describe each subfield by a pair $(g, h) \in \mathbb{Z}[t] \times \mathbb{Q}[t]$ such that g is the minimal polynomial of $\beta = h(\alpha)$. There is a bijection between the block systems of the Galois group of f and the subfields of $\mathbb{Q}(\alpha)$. These block systems are computed using cyclic subgroups of the Galois group which we get from the Dedekind criterion. When a block system is known we compute the corresponding subfield using p-adic methods. We give a detailed description for all parts of the algorithm.

1. Introduction

Let $E = \mathbb{Q}(\alpha)$ be an algebraic number field of degree n, where α is a root of a monic irreducible polynomial $f \in \mathbb{Z}[t]$. In this article a method is developed for determining all subfields $L = \mathbb{Q}(\beta)$ of E of fixed degree m over $\mathbb{Q}$. We describe each subfield L by the minimal polynomial g of β and the embedding of β into E, which is given by a polynomial $h \in \mathbb{Q}[t]$ with $h(\alpha) = \beta$.

Manuscrit reçu le 6 mai 1998.

Lemma 1.1. *1. Each subfield L of E has a representation as a pair $(g,h) \in \mathbb{Z}[t] \times \mathbb{Q}[t]$, such that $g \circ h \equiv 0 \bmod f\mathbb{Z}[t]$.*

2. A pair $(g,h) \in \mathbb{Z}[t] \times \mathbb{Q}[t]$ such that $g \circ h \equiv 0 \bmod f\mathbb{Z}[t]$ and g irreducible describes a subfield L of E.

Note that the coefficients of the *embedding polynomial* h are not necessarily integral because the equation order $\mathbb{Z}[\alpha]$ is in general not integrally closed. W.l.o.g. we assume that the degree of h is less than n, otherwise we replace h by its remainder modulo f. The lemma is used to check if a pair (g,h) presents a subfield L of E. Such a subfield L is represented in the form $\mathbb{Q}[t]/g(t)\mathbb{Q}[t]$; hence isomorphic fields are not distinguishable by g alone.

Example 1.2. *We determine all subfields L of $E = \mathbb{Q}(i\sqrt[6]{108})$ of degree 3. There are three subfields with characterizing pairs $(t^3-108, -t^2), (t^3-108, \frac{1}{12}t^5 + \frac{1}{2}t^2)$ and $(t^3-108, -\frac{1}{12}t^5 + \frac{1}{2}t^2)$. In all cases the minimal polynomial of β is the same; however, we are able to distinguish the three isomorphic subfields by their embedding polynomials.*

There are several other algorithms [1, 3, 8, 9, 12, 14, 15] for calculating subfields. In this article we improve the methods described in [12]. The generating polynomials are constructed by factorizations of polynomials over finite fields and Hensel lifting over p-adic fields. We give improved algorithms for the computations in p-adic fields. In the combinatorial part of the algorithm we can reduce the number of possibilities dramatically.

Three other methods [9, 14, 15] need factorizations of polynomials over number fields, respectively factorizations of polynomials over the rational integers of much higher degree than the degree of the given field. The method presented in [1] needs hard numerical computations and lattice reduction algorithms. Although the algorithm in [3] computes subfields it is not guaranteed that all subfields will be found. A comparison of running times is given in section 8.

This paper is organized as follows. In the next section we focus on algorithms for computations in p-adic fields. The block systems of a Galois group and their relation to subfields is presented in section 3. In section 4 and 5 we develop methods to compute generating polynomials resp. the embedding of a subfield via its block system. In the last section we discuss the efficiency of the algorithm and give some examples. This paper contains the results concerning the subfield computation of the dissertation [11] of the author.

2. Unramified p-adic extensions

The subfield computation is based on p-adic methods. Therefore we give a detailed description of the algorithms we are going to use for computation in the p-adic fields.

2.1. **Introduction.** In the following we recall some fundamental properties of unramified p-adic extensions. The proofs can be found e.g. in [2, 17]. In the following let $\mathcal{F}$ and $\mathcal{E}$ be unramified extensions of $\mathbb{Q}_p$ with maximal orders $\mathfrak{o}_{\mathcal{F}}$, $\mathfrak{o}_{\mathcal{E}}$ and prime ideals $\mathfrak{p}$, $\mathfrak{P}$, respectively. The corresponding residue class fields are denoted by $\bar{\mathcal{F}}$ and $\bar{\mathcal{E}}$.

Lemma 2.1. *For every extension $\mathbb{F}_q/\bar{\mathcal{F}}$ there exists an unique unramified extension $\mathcal{E}/\mathcal{F}$ such that $\bar{\mathcal{E}}$ and $\mathbb{F}_q$ are isomorphic. The extension $\mathcal{E}/\mathcal{F}$ is cyclic with* $\mathrm{Gal}(\mathcal{E}/\mathcal{F})$ *is isomorphic to* $\mathrm{Gal}(\bar{\mathcal{E}}/\bar{\mathcal{F}})$.

Lemma 2.2. *Let $\mathcal{E}/\mathcal{F}$ be an unramified p-adic extension of degree s. If $\rho_1, \ldots, \rho_s$ are representatives of a basis of $\bar{\mathcal{E}}/\bar{\mathcal{F}}$, then they are a $\mathfrak{o}_{\mathcal{F}}$-basis of $\mathfrak{o}_{\mathcal{E}}$.*

Using this lemma, it is straightforward to construct a $\mathfrak{o}_{\mathcal{F}}$-basis of $\mathfrak{o}_{\mathcal{E}}$. Let $\bar{\omega} \in \bar{\mathcal{F}}[t]$ be a monic irreducible polynomial of degree s and $\omega \in \mathfrak{o}_{\mathcal{F}}[t]$ with $\omega \equiv \bar{\omega} \bmod \mathfrak{p}$. Then the equation order $\mathfrak{o}_{\mathcal{F}}[\rho] = \mathfrak{o}_{\mathcal{F}} + \mathfrak{o}_{\mathcal{F}}\rho + \cdots + \mathfrak{o}_{\mathcal{F}}\rho^{s-1} = \mathfrak{o}_{\mathcal{E}}$, where $\rho \in \mathfrak{o}_{\mathcal{E}}$ is a zero of ω.

The following lemma gives a method to reduce elements of $\mathfrak{o}_{\mathcal{E}}$ modulo $\mathfrak{P}^k$.

Lemma 2.3. *Let $\mathcal{E}/\mathcal{F}$ be an unramified extension with integral basis $1, \rho, \ldots, \rho^{s-1}$. Let $x = \sum_{i=0}^{s-1} x_i\rho^i \in \mathfrak{o}_{\mathcal{E}}$ $(x_i \in \mathfrak{o}_{\mathcal{F}})$ and $k \in \mathbb{N}$. Then we have $x \in \mathfrak{P}^k$ if and only if $x_i \in \mathfrak{p}^k$ $(1 \le i \le s)$.*

Proof. Since $\mathfrak{P} = \mathfrak{p}\mathfrak{o}_{\mathcal{E}}$ it follows that $\mathfrak{P}^k = \mathfrak{p}^k\mathfrak{o}_{\mathcal{E}}$ and the assertion is an easy consequence. □

2.2. **Arithmetic in unramified p-adic extensions.** Using Lemmata 2.2 and 2.3 we are able to generate p-adic extensions, such that their equation orders are maximal. Now we explain how to compute the sum and the product of p-adic numbers. This will be done in the same way as the arithmetic in algebraic number fields. In the following let $x = \sum_{i=0}^{s-1} x_i\rho^i$ and $y = \sum_{i=0}^{s-1} y_i\rho^i$ be elements of $\mathfrak{o}_{\mathcal{E}}$ $(x_i, y_i \in \mathfrak{o}_{\mathcal{F}}$ $(0 \le i < s))$. Then we have:

$$x + y = \sum_{i=0}^{s-1}(x_i + y_i)\rho^i. \tag{1}$$

The product of x and y can be easily described via polynomial operations. Let $P_x(t) := \sum_{i=0}^{s-1} x_i t^i \in \mathfrak{o}_{\mathcal{F}}[t]$ and $P_y(t) := \sum_{i=0}^{s-1} y_i t^i \in \mathfrak{o}_{\mathcal{F}}[t]$. It follows that $xy = P_x(\rho)P_y(\rho)$. We have to solve the problem to find a basis representation of xy. We define $P_{xy} := P_x P_y \bmod \omega$. Since $\omega(\rho) = 0$ it follows that $P_{xy}(\rho) = xy$ and $\deg(P_{xy}) < s$. From

$$P_{xy}(t) = \sum_{i=0}^{s-1} z_i t^i \quad \text{we get} \quad xy = \sum_{i=0}^{s-1} z_i \rho^i. \tag{2}$$

We note that we need no divisions to reduce the product modulo ω since ω is monic. We need s^2 multiplications and additions in $\mathfrak{o}_{\mathcal{F}}$ to compute $P_x P_y$ and $s(s-1)$ multiplications and additions in $\mathfrak{o}_{\mathcal{F}}$ to reduce the result modulo ω. This leads to the following lemma.

Lemma 2.4. *The sum of two numbers $x, y \in \mathfrak{o}_{\mathcal{E}}$ can be computed using s additions in $\mathfrak{o}_{\mathcal{F}}$. The product of two numbers $x, y \in \mathfrak{o}_{\mathcal{E}}$ can be computed using $2s^2 - s$ multiplications and additions in $\mathfrak{o}_{\mathcal{F}}$.*

We remark that it is possible to divide two numbers $x, y \in \mathfrak{o}_{\mathcal{E}}$ in an analogue way to the number field case.

Now we explain how to reduce a p-adic number modulo p^k, where p is an odd prime and $k \in \mathbb{N}$. Let

$$x = \sum_{i=0}^{\infty} x_i p^i \in \mathbb{Z}_p \text{ with } x_i \in \{\frac{1-p}{2}, \ldots, \frac{p-1}{2}\}.$$

Then we define $x \bmod p^k$ as $\sum_{i=0}^{k-1} x_i p^i$, which can be interpreted as an integer in $\{\frac{1-p^k}{2}, \ldots, \frac{p^k-1}{2}\}$. Since we need frequently to embed small (negative) integers into the p-adic field, we chose the symmetric residue system. In our applications this yields usually smaller representatives. Using Lemma 2.3 we are able to reduce arbitrary p-adic numbers modulo prime ideal powers.

2.3. **Hensel lifting.** Let $f \in \mathbb{Z}[t]$ be a monic irreducible polynomial and $p \nmid \operatorname{disc}(f)$ be a prime. Let $\tilde{f}$ be the image of f under the canonical embedding from $\mathbb{Z}[t]$ to $\mathbb{Z}_p[t]$. Our aim is to factorize $\tilde{f}$ over an unramified extension $\mathcal{F}/\mathbb{Q}_p$. Since $f \bmod p$ has no multiple factors we know that $\tilde{f}$ has no multiple factors in $\mathcal{F}[t]$. The factorization can be done up to an arbitrary p-adic precision using the following lemma.

Lemma 2.5. (Hensel lemma)
Let R be a commutative ring with 1 and $\mathfrak{b}$ an ideal of R. Let $f, f_{1,0}$, and $f_{2,0}$ be monic, non-constant polynomials with the following properties:

1. $f \equiv f_{1,0} f_{2,0} \bmod \mathfrak{b}[t]$
2. *There exist $a_{i,0} \in R[t]$, $i = 1, 2$, $a_{0,0} \in \mathfrak{b}[t]$ with $a_{1,0} f_{1,0} + a_{2,0} f_{2,0} = 1 + a_{0,0}$.*

Then for every $k \in \mathbb{N}$ there exist polynomials $f_{1,k}, f_{2,k}, a_{1,k}, a_{2,k}$, and $a_{0,k} \in R[t]$ with $f_{i,k}$ monic and non-constant, and $\deg(a_{i,k}) < \deg(f_{3-i,k})$ $(i = 1,2)$ and $a_{0,k} \in \mathfrak{b}^{2^k}[t]$ such that the following conditions hold:

1. $f \equiv f_{1,k} f_{2,k} \bmod \mathfrak{b}^{2^k}[t]$
2. $f_{i,k} \equiv f_{i,0} \bmod \mathfrak{b}[t]$
3. $a_{1,k} f_{1,k} + a_{2,k} f_{2,k} = 1 + a_{0,k}$.

A proof can be found in e.g. [18]. In our examples the ideal $\mathfrak{b}$ is always a prime ideal and $R/\mathfrak{b}$ a finite field. Therefore we can compute the $a_{i,0}$ using the extended Euclidean algorithm for polynomials over finite fields. An algorithm to compute the Hensel lifting can be found in [4, 18].

These algorithms are only formulated for two factors. Using induction this can be extended to more factors. This will be demonstrated by the following example.

Example 2.6. *Let $f \equiv f_1 f_2 f_3 \bmod p$. Define $f_{1,2} := f_1 f_2$ and determine the following factorization using Hensel's lemma: $f \equiv \tilde{f}_{1,2} \tilde{f}_3 \bmod p^k$. Now compute $\tilde{f}_{1,2} \equiv \tilde{f}_1 \tilde{f}_2 \bmod p^k$. Combining these results we get $f \equiv \tilde{f}_1 \tilde{f}_2 \tilde{f}_3 \bmod p^k$.*

Now we are able to give a first method to factorize a polynomial $f \in \mathbb{Z}_p[t]$ over an extension $\mathcal{F}$ modulo $\mathfrak{p}^k$:

1. Factorize $f \equiv f_1 \cdots f_r \bmod \mathfrak{p}$.
2. Compute $f \equiv \tilde{f}_1 \cdots \tilde{f}_r \bmod \mathfrak{p}^k$ using Hensel lifting.

The disadvantage of this method is that all computations are carried out in $\mathcal{F}$. Assuming the degree of $\mathcal{F}/\mathbb{Q}_p$ is 10, we need 190 multiplications in $\mathbb{Z}_p$ to multiply two elements of $\mathfrak{o}_{\mathcal{F}}$. We improve this approach by computing some factorizations over smaller fields. This will be demonstrated in the following procedure:

1. Factorize $f \equiv f_1 \cdots f_s \bmod p$.
2. Compute $f \equiv \tilde{f}_1 \cdots \tilde{f}_s \bmod p^k$ using Hensel lifting in $\mathbb{Z}_p[t]$.
3. For $i = 1, \ldots, s$ do:
   (a) Factorize $\tilde{f}_i \equiv f_{i,1} \cdots f_{i,r_i} \bmod \mathfrak{p}$.
   (b) Compute $\tilde{f}_i \equiv \tilde{f}_{i,1} \cdots \tilde{f}_{i,r_i} \bmod \mathfrak{p}^k$ using Hensel lifting in $\mathfrak{o}_{\mathcal{F}}[t]$.
4. Combine the results: $f \equiv \tilde{f}_{1,1} \cdots \tilde{f}_{s,r_s} \bmod \mathfrak{p}^k$.

We demonstrate this by the following example:

Example 2.7. *Let $f(t) = t^{12} + t^{11} - 28t^{10} - 40t^9 + 180t^8 + 426t^7 + 89t^6 - 444t^5 - 390t^4 - 75t^3 + 27t^2 + 11t + 1$. We want to factorize f over an unramified extension $\mathcal{F}/\mathbb{Q}_3$ of degree 3 modulo $\mathfrak{p}^2$. In a first step we compute the factorization modulo 9 and get:*

$$f \equiv (t^3 + 2t - 2)(t^3 + t^2 - 3t + 2)(t^3 + 4t^2 - 4t - 2)(t^3 - 4t^2 + 2t - 1) \bmod 3^2.$$

In a second step we compute the Hensel lifting of each factor in $\mathcal{F}[t]$ modulo $\mathfrak{p}^2$, where $\mathcal{F}$ is generated by a zero ρ of $v(t) = t^3 - t + 1$. We use the notation $[a, b, c]$ for $a + b\rho + c\rho^2$ and get:

$$\begin{aligned}
t^3 + 2t - 2 &\equiv (t + [3, -4, 0])(t + [-2, 2, 3])(t + [-1, 2, -3]) \mod \mathfrak{p}^2,\\
t^3 + t^2 - 3t + 2 &\equiv (t + [3, -2, -4])(t + [3, -1, -4])(t + [4, 3, -1]) \mod \mathfrak{p}^2,\\
t^3 + 4t^2 - 4t - 2 &\equiv (t + [2, -3, -1])(t + [4, 1, -4])(t + [-2, 2, -4]) \mod \mathfrak{p}^2,\\
t^3 - 4t^2 + 2t - 1 &\equiv (t + [-4, 2, 4])(t + [-4, 1, 4])(t + [4, -3, 1]) \mod \mathfrak{p}^2.
\end{aligned}$$

We have factorized a polynomial of degree 12 but we have applied the Hensel lifting over $\mathcal{F}$ only for polynomials of degree 3. An important fact was that it was easily possible to embed $\mathbb{Z}_p$ in $\mathfrak{o}_{\mathcal{F}}$. Suppose we have a polynomial of degree 4 over $\mathbb{Z}_p$ and we want to factorize it over an unramified extension $\mathcal{E}$ of degree 4. Since the unique extension $\mathcal{E}$ is cyclic over $\mathbb{Q}_p$, f factorizes in four linear factors over $\mathcal{E}$. We know that $\mathcal{E}$ has a subfield $\mathcal{F}$ of degree 2. We know that f splits over $\mathfrak{o}_{\mathcal{F}}$ into two quadratic factors. A natural idea is first to factorize f over $\mathfrak{o}_{\mathcal{F}}$ and then to factorize the factors over $\mathfrak{o}_{\mathcal{E}}$. If we want to do this we have to solve the problem in which way we can embed the elements of $\mathfrak{o}_{\mathcal{F}}$ in $\mathfrak{o}_{\mathcal{E}}$. It suffices to give an image of all elements of a basis of $\mathfrak{o}_{\mathcal{E}}/\mathfrak{o}_{\mathcal{F}}$.

Instead of simple p-adic extensions of $\mathbb{Q}_p$ we rather consider towers of extensions. Doing this we can embed trivially the elements of $\mathfrak{o}_{\mathcal{F}}$ in $\mathfrak{o}_{\mathcal{E}}$.

Definition 2.8. *Let $p \in \mathbb{P}$, $n = p_1 \cdots p_r$ with $p_i \in \mathbb{P}$ and $p_1 \le p_2 \le \ldots \le p_r$. We call an extension $\mathcal{F} = \mathcal{F}_r$ over $\mathbb{Q}_p = \mathcal{F}_0$ successively generated, if $\mathcal{F}_i = \mathcal{F}_{i-1}(\tau_i)$, where τ_i is a zero of an irreducible and monic polynomial $v_i \in \mathfrak{o}_{\mathcal{F}_{i-1}}[t]$ of degree p_i $(1 \le i \le r)$. We denote the prime ideals in $\mathfrak{o}_{\mathcal{F}_i}$ with $\mathfrak{p}_i$.*

The following lemma follows immediately.

Lemma 2.9. *Let $p \in \mathbb{P}$ and $\mathcal{F}_r/\mathbb{Q}_p$ be successively generated. Then there is a canonical embedding from $\mathfrak{o}_{\mathcal{F}_{i-1}}$ to $\mathfrak{o}_{\mathcal{F}_i}$ $(1 \le i \le r)$.*

Now we give the whole algorithm.

Algorithm 2.10. (Hensel lifting)

<u>Input</u>: $p \in \mathbb{P}, k \in \mathbb{N},\ f \in \mathbb{Z}_p[t],\ \mathcal{F}_r/\mathbb{Q}_p$ *successively generated.*

<u>Output</u>: *Factorization of f in $\mathfrak{o}_{\mathcal{F}_r}[t]$ modulo $\mathfrak{p}_r^k$.*

<u>Step 1</u>: *Compute the factorization of* $f \equiv f_{0,1} \cdots f_{0,s_0} \mod p^k$.

<u>Step 2</u>: *For $i = 1, \ldots, r$ do*

1. *Compute the factorizations of $f_{i-1,j}$ in $\mathfrak{o}_{\mathcal{F}_i} \bmod \mathfrak{p}_i^k$ $(1 \le j \le s_{i-1})$.*

2. Combine the results: $f \equiv f_{i,1} \cdots f_{i,s_i} \mod \mathfrak{p}_i^k$.

Step 3: *Print the result:* $f \equiv f_{r,1} \cdots f_{r,s_r} \mod \mathfrak{p}_r^k$.

2.4. **The generalized Newton lifting.** Let F be an algebraic number field and R an order in F. The special case $F = \mathbb{Q}$ and $R = \mathbb{Z}$ is possible. Let $f, g \in R[t]$ be irreducible polynomials of degree n resp. m. A zero α of f generates the number field $E = F(\alpha)$. Furthermore we know a modulo p-approximation $\beta_0 \in R$ of a zero of g, that means $g(\beta_0) \equiv 0 \mod pR$.

In the following we use the notation mod p^k instead of mod $p^k R$. We denote with $\mathfrak{d}(f)$ the principal ideal in R generated by $\operatorname{disc}(f)$. We choose a prime p such that $\gcd(pR, \mathfrak{d}(f)\mathfrak{d}(g)) = R$. Using the extended Euclidean algorithm we can compute an element $\omega_0 \in \mathfrak{o}_E$ such that $\omega_0 g'(\beta_0) \equiv 1 \mod p$ holds. In the following we construct elements β_k, ω_k with the following properties:

$$\beta_{k+1} \equiv \beta_k \mod p^{2^k} \tag{3}$$

$$\omega_{k+1} \equiv \omega_k \mod p^{2^k} \tag{4}$$

$$g(\beta_k) \equiv 0 \mod p^{2^k} \tag{5}$$

$$\omega_k g'(\beta_k) \equiv 1 \mod p^{2^k}. \tag{6}$$

We use the following double iteration:

$$\beta_{k+1} \equiv \beta_k - \omega_k g(\beta_k) \mod p^{2^{k+1}} \tag{7}$$

$$\omega_{k+1} \equiv \omega_k [2 - \omega_k g'(\beta_{k+1})] \mod p^{2^{k+1}}. \tag{8}$$

The correctness can be easily verified [10]. We remark that it is possible to solve our problem using the following iteration:

$$\beta_{k+1} \equiv \beta_k - \frac{g(\beta_k)}{g'(\beta_k)} \mod p^{2^{k+1}}.$$

The disadvantage of this approach is that divisions are much more complicated to compute. If we analyze the double iteration we notice that the evaluations of $g(\beta_k)$ and $g'(\beta_k)$ are the most expensive steps. Using Horner's scheme we need $(m-1) + (m-2) = 2m - 3$ multiplications of algebraic numbers of degree n where $m = \deg(g)$. We need $2n^2 - n$ multiplications in R to multiply two numbers in E. Therefore we need $(2m-3)(2n^2-n)$ multiplications in R to compute their evaluations.

It is better to first compute $1, \beta_k, \ldots, \beta_k^m$ which can be done using $m-1$ multiplications in E. After this we need $m + (m-1)$ multiplications of elements of F with elements of E to compute the evaluations. Altogether we need $(m-1)(2n^2-n) + n(2m-1) = (2m-2)n^2 + mn$ multiplications in R. Using this approach we save about half of the multiplications. We have not looked at the size of the coefficients. Practical experience shows that the second approach is about 50% faster than the first one.

Algorithm 2.11. (Newton lifting)

Input: $p \in \mathbb{P}, k \in \mathbb{N},\ f, g \in R[t],\ \beta_0 \in E$ *with* $g(\beta_0) \equiv 0 \bmod p$.

Output: $\beta_k \in E$ *with* $g(\beta_k) \equiv 0 \bmod p^{2^k}$ *and* $\beta_k \equiv \beta_0 \bmod p$.

Step 1: *Compute* $\omega_0 \in E$ *with* $\omega_0 g'(\beta_0) \equiv 1 \bmod p$.

Step 2: *Compute* $\beta_1 \equiv \beta_0 - \omega_0 g(\beta_0) \bmod p^2$.

Step 3: *For* $i = 1, \ldots, k-1$ *do:*

1. *Compute* $1, \beta_i, \ldots, \beta_i^m \bmod p^{2^{i+1}}$.
2. *Compute* $\omega_i \equiv \omega_{i-1}[2 - \omega_{i-1} g'(\beta_i)] \bmod p^{2^i}$.
3. *Compute* $\beta_{i+1} \equiv \beta_i - \omega_i g(\beta_i) \bmod p^{2^{i+1}}$.

Step 4: *Print* β_k *and terminate.*

In the following we give a variant of this algorithm. In our application of the Newton lifting we want to find an element $\beta \in E$ with $g(\beta) = 0$. We know estimates for the numerators and denominators of the coefficients of β. In this case the following lemma is very useful.

Lemma 2.12. *Let* $U, M \in \mathbb{N}$ *such that* $(U, M) = 1$ *and suppose that there exists a pair of integers* (C, D) *such that* $C \equiv DU \pmod M$ *with* $D > 0$ *and* $|C|, D < \sqrt{M/2}$. *Then the pair is uniquely determined and there exists an efficient algorithm to compute it.*

The proof and the algorithm can be found in [5]. We remark that algebraic numbers are reconstructed by applying the lemma to all coefficients.

If we know estimations for the numerators and denominators we are able to compute β from β_k. Since the a priori estimates we use are usually not sharp, we want to use a smaller k in the Newton lifting process to compute β. One idea is to compute an element $\tilde\beta$ from β_k using Lemma 2.12. Now it can be checked if $g(\tilde\beta) = 0$. Unfortunately it turns out that an evaluation with a "wrong" β is very expensive. Therefore we need a good test which is likely to detect $\tilde\beta = \beta$ at an early stage. This is used in the following algorithm.

Algorithm 2.13. (Newton lifting)

Input: $p \in \mathbb{P}, k \in \mathbb{N},\ f, g \in R[t],\ \beta_0 \in E$ *with* $g(\beta_0) \equiv 0 \bmod p$.

Output: $\beta \in E$ *with* $g(\beta) = 0$ *or "false".*

Step 1: *Compute* $\omega_0 \in E$ *with* $\omega_0 g'(\beta_0) \equiv 1 \bmod p$.

Step 2: $\beta_1 \equiv \beta_0 - \omega_0 g(\beta_0) \bmod p^2$.

Step 3: *Set* $\beta_{old} := 0$.

Step 4: *For* $i = 1, \dots, k-1$ *do:*

1. *Compute* $1, \beta_i, \dots, \beta_i^m \bmod p^{2^{i+1}}$.
2. $\omega_i \equiv \omega_{i-1}[2 - \omega_{i-1} g'(\beta_i)] \bmod p^{2^i}$.
3. $\beta_{i+1} \equiv \beta_i - \omega_i g(\beta_i) \bmod p^{2^{i+1}}$.
4. *Define* β_{new} *to the result of the reconstruction of* β_{i+1} *and* $p^{2^{i+1}}$ *using Lemma 2.12.*
5. *If* $\beta_{old} = \beta_{new}$, *then compute* $g(\beta_{new})$. *If this evaluation equals 0, terminate and return* β_{new} .
6. *Set* $\beta_{old} := \beta_{new}$.

Step 5: *Compute* $g(\beta_{new})$. *If the result equals 0, then return* β_{new} *and terminate.*

Step 6: *Terminate with the message "false".*

3. Blocks of imprimitivity

In this section we develop some properties about blocks of imprimitivity. We recall a correspondence between blocks and subfields, which is very useful for the computation. In the following let $f \in \mathbb{Z}[t]$ be an irreducible monic polynomial with roots $\{\alpha = \alpha_1, \dots, \alpha_n\}$ in a suitable extension. The Galois group $G = \mathrm{Gal}(f)$ operates transitively on $\Omega := \{\alpha_1, \dots, \alpha_n\}$.

3.1. Introduction.

Definition 3.1. (Blocks of imprimitivity)

1. $\emptyset \neq \Delta \subseteq \Omega$ *is called block (of imprimitivity), if* $\Delta^\tau \cap \Delta \in \{\emptyset, \Delta\}$ *for all* $\tau \in G$.
2. $\Delta = \{\alpha_i\}$ $(1 \leq i \leq n)$ *and* $\Delta = \Omega$ *are called trivial blocks.* G *is called imprimitive if there exists a non-trivial block. Otherwise* G *is called primitive.*
3. *Blocks* $\Delta_1, \dots, \Delta_m$ *with* $\Delta_i \neq \Delta_j$ $(1 \leq i < j \leq m)$ *are called a (complete) block system, if the set* $\{\Delta_1, \dots, \Delta_m\}$ *remains invariant under* G.

If Δ is a block it is easy to see that Δ^τ is a block, too. It follows that each block is contained in exactly one block system. The number of elements in a block or the number of elements of a block of a block system is called the size of a block or a block system.

The proof of the following theorem can be found in [20, Theorem 2.3]. Combined with the main theorem of Galois theory we get a correspondence between block systems and subfields.

Theorem 3.2. *The correspondence $\Delta \mapsto G_\Delta := \{\tau \in G \mid \Delta^\tau = \Delta\}$ is a bijection between the set of blocks which contain α and the set of subgroups of G containing the isotropy subgroup G_α of α.*

The following diagram illustrates our situation:

$$\begin{tikzcd}
\mathbb{Q}(\alpha_1, \dots, \alpha_n) \arrow[d, dash] & \{id\} \arrow[d, dash] & \\
\mathbb{Q}(\alpha_1) \arrow[d, dash, "d"] & G_{\alpha_1} \arrow[d, dash] & \{\alpha_1\} \arrow[d, dash] \\
\mathbb{Q}(\beta) \arrow[d, dash, "m = \frac{n}{d}"] & H \arrow[d, dash] & H\alpha_1 = \{\alpha_{i_1}, \dots, \alpha_{i_d}\} = \Delta_1 \arrow[d, dash] \\
\mathbb{Q} & G & \{\alpha_1, \dots, \alpha_n\}
\end{tikzcd}$$

We have a bijection between subfields L of E and blocks Δ which contain α. In this case we say that L corresponds to Δ. The proof of the following lemma can be found in [20].

Lemma 3.3. *Let B_1 and B_2 two blocks which contain α with corresponding subfields L_1 and L_2 of E. Then $B := B_1 \cap B_2$ is a block which contains α. It corresponds to a subfield $L = L_1 L_2$ of E. Furthermore L_1 is a subfield of L_2 if and only if $B_2 \subseteq B_1$.*

This lemma is very useful if some subfields are already known. This will be discussed later.

Suppose that we know a complete block system $\Delta_1, \dots, \Delta_m$ which corresponds to a subfield L. From $H := G_{\Delta_1}$ we get $L = \mathrm{Fix}(H)$. Define

(9) $$\delta_i := \prod_{\gamma \in \Delta_i} \gamma \ (1 \le i \le m).$$

Therefore we get $\delta_1 \in \mathrm{Fix}(H) = L$. Furthermore the δ_i $(1 \le i \le m)$ are all conjugates of δ_1. This means that

(10) $$g(t) = \prod_{i=1}^{m} (t - \delta_i) \in \mathbb{Z}[t]$$

is the characteristic polynomial of $\delta_1 \in L$ over $\mathbb{Q}$. This polynomial is of the form $g(t) = \hat{g}^j$ with $j \in \mathbb{N}$ and $\hat{g}$ irreducible. In the case that g is irreducible we have found a primitive element of L. Otherwise the polynomial g has multiple roots which can be easily checked. In this case we make a linear transformation $f(t) \leftarrow f(t-a)$ with $a \in \mathbb{Z}$ and compute a new g. Later we will prove that at most n substitutions leads to multiple roots for g.

3.2. **The Dedekind criterion.** We have reduced the problem of computing subfields to the problem of computing block systems of the Galois group of G. This reduction is only theoretical since the Galois group computation is a very difficult problem for higher degrees. We want to use the knowledge of cyclic subgroups of the Galois group which we get from the following theorem.

Theorem 3.4. (Dedekind Criterion)
Let R be a UFD, $\mathfrak{p}$ a prime ideal in R, $\bar{R} := R/\mathfrak{p}$ its residue class ring, $f \in R[t]$ and $\bar{f} \in \bar{R}[t]$ with $f \equiv \bar{f} \bmod \mathfrak{p}$. If $\bar{f}$ is square–free, it follows that $\bar{G} = \mathrm{Gal}(\bar{f})$ is isomorphic to a subgroup of $G = \mathrm{Gal}(f)$.

The Dedekind criterion allows us to determine cyclic subgroups of G which are generated by a permutation $\pi \in G$. Let $\pi = \pi_1 \cdots \pi_u$ be the decomposition of π into disjoint cycles and $n_i = |\pi_i|$ the number of zeros permuted by π_i $(1 \le i \le u)$. We say that π is of cycle type $[n_1, \dots, n_u]$ and w.l.o.g. we can assume $n_1 \le \ldots \le n_u$. In our situation we choose a prime $p \nmid \mathrm{disc}(f)$ to obtain a congruence factorization $f \equiv f_1 \cdot \ldots \cdot f_u \bmod p\mathbb{Z}[t]$. It follows that n_i $(i = 1, \dots, u)$ coincides with the degree of the polynomial f_i. The cycles π_i permute the roots of f_i.

Example 3.5. *Let $f(t) = t^4 + 2$ be a generating polynomial of K and $G = \mathrm{Gal}(f)$.*

1. $f(t) \equiv t^4 \bmod 2$.
2. $f(t) \equiv (t+2)(t+1)(t^2+1) \bmod 3$.
3. $f(t) \equiv t^4 + 2 \bmod 5$.
4. $f(t) \equiv (t^2 + 6t + 4)(t^2 + t + 4) \bmod 7$.

Let p denote the modulus. In the first case p divides the discriminant and the Dedekind criterion is of no use. In the other cases we get cycles of cycle type $[1,1,2]$, $[4]$ and $[2,2]$. In all of these cases the roots can only be identified modulo p in a suitable finite field.

3.3. **Potential block systems.** In the algorithm we are trying to enumerate all block systems without knowing the Galois group G. So we enumerate a larger set of potential block systems that can be defined with the knowledge of a cyclic subgroup of G. This subgroup can be obtained with theorem 3.4.

Fix an arbitrary $\pi \in G$. Let $\pi = \pi_1 \cdots \pi_u$ be the decomposition of π into disjoint cycles of length $|\pi_i| = n_i$ $(1 \le i \le u)$.

Definition 3.6. *A subset $A \subseteq \Omega$ with d elements is called potential block of size d, if $A^{\pi^j} \cap A \in \{\emptyset, A\}$ for $1 \le j \le |\langle\pi\rangle|$. A system $A_1, \ldots, A_m$ of potential blocks of size d is called potential block system of size d, if*

1. $\Omega = \bigcup_{1 \le i \le m} A_i$,
2. $A_i \cap A_j = \emptyset$ $(i \neq j)$,
3. $A_i^{\pi^j} \in \{A_1, \ldots, A_m\}$ $(1 \le i \le m, 1 \le j \le |\langle\pi\rangle|)$.

Remark 3.7. *The definitions potential block and potential block system depend on π. A block is always a potential block and a block system is always a potential block system.*

Our goal is to determine all potential block systems (for one π). In the following we give some useful properties of potential block systems. We say that a cycle π_i contains an element α if this element is not fixed under this cycle or $\pi_i = (\alpha)$.

Theorem 3.8. *Let A be a potential block corresponding to π and k be the smallest positive integer such that $A^{\pi^k} = A$. If a cycle π_l of length n_l contains an element of A, then k divides n_l and π_l contains exactly $\frac{n_l}{k}$ elements of A.*

Proof. Since A is a potential block it follows that there exists a k with

$$A^{\pi^j} \cap A = \emptyset \text{ for } 1 \le j < k \text{ and } A^{\pi^k} = A.$$

Suppose that α is contained in A and π_l. It follows that all elements of the form $\alpha^{\pi^{ck}}$ $(c \in \mathbb{N})$ are contained in A and π_l. From $\alpha^{\pi^{n_l}} = \alpha$ we see that k divides n_l. Furthermore π_l contains exactly $\frac{n_l}{k}$ elements of A. □

Definition 3.9. *The number k from theorem 3.8 is called inertia degree of the potential block.*

Theorem 3.10. *Let $A_1, \ldots, A_m$ be a potential block system corresponding to π of inertia degrees $k_1, \ldots, k_m$. If A_i and A_j contain an element of the same cycle, it follows that $k_i = k_j$. In this case A_i contains an element of the cycle π_l if and only if A_j contains an element of the same cycle.*

Proof. There exists a minimal number $c \in \mathbb{N}$ such that $A_i^{\pi^c} \cap A_j \neq \emptyset$. From the definition of a potential block system it follows that $A_i^{\pi^c} = A_j$. The assertion follows immediately. □

Definition 3.11. *Let $A_1, \ldots, A_m$ be a (potential) block system of inertia degrees $k_1, \ldots, k_m$. We call $A_i, A_i^{\pi}, \ldots, A_i^{\pi^{k_i - 1}}$ a (potential) block cluster $(1 \le i \le m)$.*

¿From theorem 3.10 we get that all blocks of a (potential) block cluster have the same inertia degree.

The preceding two theorems are very important for the construction of potential block systems. We will construct systems of subsets $A_1, \ldots, A_m \subseteq \Omega$ of size d and corresponding inertia degrees $k_1, \ldots, k_m$ with the following properties:

1. $|A_i| = d$ for $1 \leq i \leq m$.
2. If A_i contains elements of a cycle π_l, then A_i contains exactly $\frac{n_l}{k_i}$ elements of this cycle.
3. $\bigcup_{1 \leq i \leq m} A_i = \Omega$.
4. $A_i \cap A_j = \emptyset \ (i \neq j)$.
5. All potential blocks of a potential block cluster are contained in $A_1, \ldots, A_m$, that means $A_i^{\pi^j} \in \{A_1, \ldots, A_m\} \ (0 \leq j < k_i)$.

A system of subsets $A_1, \ldots, A_m$ is a potential block system if and only if it has the above properties. These properties are sufficient to give an efficient algorithm to compute all potential block systems and therefore all block systems.

To compute the minimal polynomial g of a primitive element of a subfield L we need a method to compute the zeros which are contained in a potential block. Let $p \in \mathbb{P}$ with $p \nmid \operatorname{disc}(f)$ and $\bar{f} \in \mathbb{F}_p[t]$ be the image of f under the canonical mapping from $\mathbb{Z}$ to $\mathbb{F}_p$. We denote the zeros of f in a suitable extension $\mathbb{F}_q$ of $\mathbb{F}_p$ with $\bar{\alpha}_1, \ldots, \bar{\alpha}_n$. Furthermore let $\bar{f} = \bar{f}_1 \cdots \bar{f}_u \in \mathbb{F}_p[t]$ be a complete factorization. Suppose that $\pi = \pi_1 \cdots \pi_u$ is computed using Dedekind's criterion 3.4. We know that π_i permutes the zeros of $\bar{f}_i$.

Let $A_1, \ldots, A_k$ be a potential block cluster of inertia degree k. W.l.o.g. we assume that it contains the zeros of $\pi_1, \ldots, \pi_v$ that means the potential blocks contain the zeros of $\bar{f}_1, \ldots, \bar{f}_v$. Let

$$\bar{f}_i = \bar{f}_{i,1} \cdots \bar{f}_{i,k} \text{ in } \mathbb{F}_{p^k}[t] \text{ and } \pi_i^k = \pi_{i,1} \cdots \pi_{i,k} \ (1 \leq i \leq v).$$

Then $\pi_{i,j}$ permutes the zeros of $\bar{f}_{i,j}$ $(1 \leq j \leq k,\ 1 \leq i \leq v)$. Therefore all these zeros are contained in one potential block. We want to compute equation (9). Therefore we are only interested in the product of the zeros of $\bar{f}_{i,j}$ which is equal to $(-1)^{\deg(\bar{f}_{i,j})} \bar{f}_{i,j}(0)$. That means that there is no reason to factor $\bar{f}$ over a larger finite field.

Definition 3.12. (Polynomial representation of potential blocks and block systems)

Let A be a set of polynomials. We say that A is a potential block in polynomial representation if the set of zeros of the polynomials in A is a potential block. We say that a potential block system is given in polynomial representation if all potential blocks are given in polynomial representation.

A potential block cluster is given in polynomial representation if all its blocks are given in polynomial representation.

The polynomials of a polynomial representation are not necessarily linear. Now we can formulate our algorithm to compute potential block systems.

Algorithm 3.13. (ComputePotentialBlockSystems)

Input: *Generating polynomial f of E, the block size d and a prime $p \nmid \operatorname{disc}(f)$.*

Output: *A list of all potential block systems of size d in polynomial representation.*

Step 1: *Compute $f(t) \equiv \bar{f}_1(t) \cdots \bar{f}_u(t) \bmod p\mathbb{Z}[t]$.*

Step 2: *Set $Z := \{\bar{f}_1, \ldots, \bar{f}_u\}$ and call ComputeBlockCluster($Z, d, \emptyset$).*

Algorithm 3.14. (ComputeBlockCluster)

Input: *A set Z consisting of r irreducible polynomials $\bar{f}_i$ in $\mathbb{F}_p[t]$, a block size $d \in \mathbb{N}$ and a set Y consisting of already computed block clusters in polynomial representation.*

Output: *A list of potential block systems of size d in polynomial representation.*

Step 1: *Set $k := 1$ and $n_i := \deg(\bar{f}_i)$ $(1 \le i \le r)$.*

Step 2: *Determine all $B \subseteq \{2, \ldots, r\}$ (including $\emptyset$) with $dk - n_1 = \sum_{b \in B} n_b$ and $k \mid n_b$ for all $b \in B$.*

Step 3: *For all computed B do:*
1. *Set $Z' := \{\bar{f}_b \mid b \in B \cup \{1\}\}$.*
2. *Set $Y := Y \cup \{Z'\}$.*
3. *If $Z = Z'$, call PrintBlockSystem(Y', d);*
   otherwise call ComputeBlockCluster($Z \setminus Z', d, Y$).
4. *Set $Y := Y \setminus \{Z'\}$.*

Step 4: *Terminate, if $k = n_1$. Otherwise set $k := \min\{l \in \mathbb{N} \mid l > k \text{ and } l \mid n_1\}$ and go to Step 2.*

Algorithm 3.15. (PrintBlockSystem)

Input: *A set Y consisting of r sets Y_i of block clusters in polynomial representation and a block size d.*

Output: *A list of all potential block systems in polynomial representation corresponding to Y.*

Step 1: *Set* $A := \emptyset$.

Step 2: *For* $i = 1, \dots, r$ *do*

1. *Set* $s_i := |Y_i|$. *Denote the elements of* Y_i *with* $f_{i,1}, \dots, f_{i,s_i}$.
2. *Set* $k_i := \frac{1}{d} \sum_{j=1}^{s_i} \deg(f_{i,j}) \in \mathbb{N}$.
3. *Factorize* $f_{i,j} = f_{i,j,1} \cdots f_{i,j,k_i}$ *in* $\mathbb{F}_{p^{k_i}}[t]$ $(1 \le j \le s_i)$.
4. *Let* σ *be the Frobenius automorphism of* $\mathbb{F}_{p^{k_i}}/\mathbb{F}_p$. *Sort the* $f_{i,j,l}$, *such that* $f_{i,j,l} = \sigma(f_{i,j,l-1})$ $(1 \le j \le s_i, 2 \le l \le k_i)$.
5. *Set* $A_l := \{f_{i,1,l}, \dots, f_{i,s_i,l}\}$ $(1 \le l \le k_i)$.
6. *Add* $A_1, \dots, A_{k_i}$ *to* A.

Step 3: *Set* $\pi_{i,j}(f_{i,j,l}) := f_{i,j,l+1}$ $(f_{i,j,k_i+1} := f_{i,j,1})$ *and* $\pi_{i,j}(f_{i',j',l}) := f_{i',j',l}$ *for* $(i,j) \neq (i',j')$ $(1 \le i \le r, 1 \le j \le s_i, 1 \le l \le k_i)$.

Step 4: *Set* $M := \{\prod_{i=1}^{r} \prod_{j=2}^{s_i} \pi_{i,j}^{e_{i,j}} \mid 1 \le i \le r,\ 2 \le j \le s_i,\ 0 \le e_{i,j} < k_i\}$.

Step 5: *For all* $\tau \in M$ *print the potential block system* $A^\tau := \{A_1^\tau, \dots, A_m^\tau\}$.

The above algorithm computes all potential block systems $A_1, \dots, A_m$. Each A_i contains irreducible polynomials $f_{i,j,l}$ which are given over an extension of $\mathbb{F}_p$. The block consists exactly of the zeros of these polynomials. We have remarked that we are only interested in the product of the zeros. It is possible that polynomials in different blocks are given over different extension fields, but in a block cluster all polynomials are given over the same extension field. Let $A_1, \dots, A_k$ be a block cluster. Then we have (compare (10)):

$$\prod_{i=1}^{k} (t - \delta_i) \in \mathbb{F}_p[t] \text{ with } \delta_i = \prod_{\gamma \in A_i} \gamma \ (1 \le i \le k).$$

3.4. **The intersection of block systems.** For the computation of potential block systems we have used the knowledge of a $\pi \in G$. If we do not find a "good" π, we have to consider a lot of potential block systems which are not block systems.

We have seen in Lemma 3.3 that the intersection of two blocks is a block. We want to use this in two ways. Firstly we are able to compute new block systems from existing ones. Secondly we want to reduce the number of potential block systems to consider. That means, we need one (or more) criteria to distinguish "wrong" potential block systems from block systems.

Definition 3.16. *The intersection of two (potential) block systems* $\Delta_1, \dots, \Delta_m$ *and* $\hat{\Delta}_1, \dots, \hat{\Delta}_{\hat{m}}$ *are the (potential) blocks which are contained in the set* $\{\Delta_i \cap \hat{\Delta}_j \mid 1 \le i \le m,\ 1 \le j \le \hat{m}\} \setminus \{\emptyset\}$.

Lemma 3.17. *The intersection of two block systems* $\Delta_1, \dots, \Delta_m$ *and* $\hat{\Delta}_1, \dots, \hat{\Delta}_{\hat{m}}$ *is a block system of size* $c \in \mathbb{N}$. *The intersection of two blocks* Δ_i *and* $\hat{\Delta}_j$ *is the empty set or contains c elements* $(1 \leq i \leq m, 1 \leq j \leq \hat{m})$.

Proof. The assertion follows from the fact that a block is contained in exactly one block system. □

In the following let $\Delta_1, \dots, \Delta_m$ be a block system and $A_1, \dots, A_r$ a potential block system. W.l.o.g. we assume that $\alpha \in \Delta_1 \cap A_1$ and $c = |\Delta_1 \cap A_1|$. In the sequel we will give some more necessary conditions for potential block systems to be block systems. We will use this to reduce the number of wrongly computed generating polynomials and embeddings. The following lemma is an immediate consequence of the last lemma.

Lemma 3.18. *Let* $M = \{\Delta_i \cap A_j \mid 1 \leq i \leq m, 1 \leq j \leq r\} \setminus \{\emptyset\}$. *If* M *contains an element of size not equal c, it follows that* $A_1, \dots, A_r$ *is not a block system.*

Definition 3.19. *The number c of the last lemma is called intersection number. If there is an element of size not equal to c in* M, *the intersection number is defined to be* 0. *The intersection number of a potential block cluster is defined in an analogue way.*

Let us consider the intersection $\check{\Delta}_1, \dots, \check{\Delta}_{\check{m}}$ of two block systems $\Delta_1, \dots, \Delta_m$ and $\hat{\Delta}_1, \dots, \hat{\Delta}_{\hat{m}}$. We know that the intersection is a block system, too. Let $A_1, \dots, A_r$ be a potential block system. We want to test if $A_1, \dots, A_m$ can be a block system. A natural question to ask if it is necessary to intersect $A_1, \dots, A_m$ with all known block systems to get maximal information.

Example 3.20. *To simplify we consider only the indices of the zeros. Let* $\Omega = \{1, \dots, 12\}$. *Suppose we know two block systems* $\{1, 2, 7, 8\}$, $\{3, 4, 9, 10\}$, $\{5, 6, 11, 12\}$ *and* $\{1, 2, 3, 4, 5, 6\}$, $\{7, 8, 9, 10, 11, 12\}$. *The intersection of these block systems is* $\{1, 2, \}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \{9, 10\}, \{11, 12\}$. *We consider the potential block system* $\{1, 2, 3, 10, 11, 12\}, \{4, 5, 6, 7, 8, 9\}$. *Looking at the intersection with the first two block systems we get no contradiction. But we have* $\{1, 2, 3, 10, 11, 12\} \cap \{1, 2\} = \{1, 2\}$ *and* $\{1, 2, 3, 10, 11, 12\} \cap \{3, 4\} = \{3\}$. *This proves that* $A_1, \dots, A_r$ *is not a potential block system.*

This example shows that it is useful to consider all known block systems. With this method we can decide for most potential block systems that they are not block systems. We summarize what we have done up to now. Let $L_1, \dots, L_w$ be the known subfields and B be a set of potential block systems.

1. Compute the set S containing the block systems corresponding to $L_1, \dots, L_w$.

2. Compute the intersection of all block systems in S and add the non-trivial ones to S.
3. Set $T := \emptyset$ and for all potential block systems $A_1, \dots, A_m$ contained in B do:
   (a) Intersect $A_1, \dots, A_m$ with each block system from S and apply Lemma 3.18.
   (b) If $A_1, \dots, A_m$ passes all tests, then add it to T.
4. Print T.

The block systems which are computed in steps 1 and 2 are known in most cases. Now we give a method how to compute a block system if we know a subfield and the zeros of f in some representation. This algorithm is useful if some subfields are known or if we want to change the prime p. The following lemma can be easily proved.

Lemma 3.21. *Let $\alpha_1, \dots, \alpha_n$ be the zeros of f and $\beta_1, \dots, \beta_m$ be the zeros of g given in the same completion. If the β_i are pairwise distinct, then $\Delta_1, \dots, \Delta_m$ with*

$$\Delta_i := \{\alpha_j \mid h(\alpha_j) = \beta_i,\ 1 \le j \le n\} \ \ (1 \le i \le m)$$

is the corresponding block system.

The intersection method allows us easily to detect many potential block systems which are not block systems. In the following we give conditions to exclude a lot of block systems with one intersection. If we look at algorithm 3.15 we see that potential block systems consist of r potential block clusters. We want to give conditions that a potential block cluster cannot be a part of a block system. We denote the inertia degrees of the block clusters with $k_1, \dots, k_r$. If we analyze algorithm 3.15 we see that each block cluster consists of s_i modulo p factors of f. Suppose that V_i $(1 \le i \le r)$ is a set of all constructed block clusters. In the last step of the algorithm all potential block systems are constructed in the following way:

$$\{v_1, \dots, v_r \mid v_i \in V_i,\ 1 \le i \le r\}.$$

We have used the notation v_i for $A_{i,1}, \dots, A_{i,k_i}$ $(1 \le i \le r)$. The number of elements of V_i only depend on k_i and s_i. We get:

$$|V_i| = k_i^{s_i - 1}.$$

The algorithm generates $|V_1| \cdots |V_r|$ potential block systems. Suppose we are able to show that a potential block cluster $v_1 \in V_1$ cannot be part of a block system. In this case we have decreased the number of possibilities by $|V_2| \cdots |V_r|$. Furthermore we only combine block clusters with the same intersection number (Definition 3.19). We want to use all known block systems to get maximal information. We denote with $(c_1, \dots, c_{\hat{w}})^t$ the intersection numbers of a potential block cluster with $\hat{w}$ block systems, where c_i is the intersection number with the ith block system.

Algorithm 3.22. (Intersection algorithm)

Input: *$V_i = \{v_{i,1}, \dots, v_{i,|V_i|}\}$ $(1 \le i \le r)$, $k_1, \dots, k_r$, $s_1, \dots, s_r$ as defined in the above text. $\hat{w}$ known block systems.*

Output: *Set of potential block systems, such that there is no contradiction with the known block systems.*

Step 1: *For $i = 1, \dots, r$ do:*
1. *For $j = 1, \dots, |V_i|$ do:*
   (a) *Set $W_{i,j}$ to the intersection number of $v_{i,j}$ with the known block systems.*
   (b) *If one of the components of $W_{i,j}$ equals 0, set $V_i := V_i \setminus \{v_{i,j}\}$.*

Step 2: *Compute all potential block systems $v_{1,j_1}, \dots, v_{r,j_r}$ with $W_{1,j_1} = \dots = W_{r,j_r}$ and $v_{i,j_i} \in V_i$ $(1 \le i \le r)$ and print the computed ones.*

Step 3: *Terminate the algorithm.*

4. The computation of generating polynomials

We call a minimal polynomial of a primitive element of an extension a generating polynomial. As in the last sections let $E = \mathbb{Q}(\alpha)$, f be the minimal polynomial of α, and $\{\alpha = \alpha_1, \dots, \alpha_n\}$ be the roots of f. The Galois group G operates transitively on the roots of f. In the last section we have seen how to compute potential block systems corresponding to a permutation π. In this section we will explain how to get generating polynomials from a block system. As a byproduct, we get more necessary conditions for potential block systems to be block systems. Nevertheless, we will not get sufficient conditions. Wrong systems remaining after this step will finally be removed in the concluding step, the computation of the embedding.

Let $\Delta_1, \dots, \Delta_m$ be a block system consisting of zeros of f, where the zeros of Δ_i are in the splitting field N of E. Furthermore let $\mathfrak{P}$ be an arbitrary prime ideal of $\mathfrak{o}_N$ lying over p. We denote with $\mathcal{E} = N_{\mathfrak{P}}$ the p-adic completion. Let Φ be the canonical embedding from N to $\mathcal{E}$.

Now let $\tilde{f} = \Phi(f)$ and $\{\tilde{\alpha}_1, \dots, \tilde{\alpha}_n\}$ be the zeros of $\tilde{f}$ in $\mathcal{E}$, where $\Phi(\alpha_i) = \tilde{\alpha}_i$. Letting $\tilde{\Delta}_i = \Phi(\Delta_i)$ $(1 \le i \le m)$ we define:

$$\tilde{g}(t) := \prod_{i=1}^{m}(t - \tilde{\delta}_i) \in \mathbb{Z}_p[t] \text{ with } \tilde{\delta}_i := \prod_{\tilde{\gamma} \in \tilde{\Delta}_i} \tilde{\gamma} \ (1 \le i \le m). \tag{11}$$

$$g(t) := \prod_{i=1}^{m}(t - \delta_i) \in \mathbb{Z}[t] \text{ with } \delta_i := \prod_{\gamma \in \Delta_i} \gamma \ (1 \le i \le m). \tag{12}$$

Therefore we get:

Theorem 4.1. *Let $\Delta_1, \dots, \Delta_m$ be a block system and g and $\tilde{g}$ as defined in* (11) *and* (12), *then $\Phi(g) = \tilde{g}$.*

Supposing that $\Delta_1, \dots, \Delta_m$ is only a potential block system corresponding to π we still get $\tilde{g} \in \mathbb{Z}_p[t]$, where p corresponds to π. We remark that we have no method to compute Φ explicitly. We know that for each extension $\mathbb{F}_q/\mathbb{F}_p$ there exists a unique unramified p-adic extension $\mathcal{E}/\mathbb{Q}_p$ such that the residue class field equals $\mathbb{F}_q$. In the last section we have developed an algorithm to compute potential block systems $A_1, \dots, A_m$. We have identified the zeros resp. the δ_i in a suitable finite field. Using the p-adic methods presented in section 2 it is possible to compute these values modulo $\mathfrak{p}^k$. The following lemma is an immediate consequence.

Lemma 4.2. *Let $g, \tilde{g}$ and $\tilde{\delta}_i \in \mathcal{E}$ $(1 \le i \le m)$ be as defined in* (11) *and* (12). *Furthermore let $k \in \mathbb{N}$ and $\mathfrak{p}$ be the maximal ideal of $\mathfrak{o}_{\mathcal{E}}$. Supposing $\hat{\delta}_i \equiv \tilde{\delta} \bmod \mathfrak{p}^k$ $(1 \le i \le m)$ and $\hat{g}(t) = \prod_{i=1}^m (t - \hat{\delta}_i)$ we get $\hat{g} \equiv \tilde{g} \bmod p^k$. Thus we have $\hat{g} \equiv g \bmod p^k$.*

Let M be a bound for the size of the coefficients of g and suppose $p^k > 2M$. Then it follows that $\hat{g} = g$ if we choose the symmetrical residue system $\{\frac{-(p^k-1)}{2}, \dots, \frac{p^k-1}{2}\}$ for the coefficients of $\hat{g}$. The following lemma gives us an estimation for M. It is an immediate consequence of [4, Lemma 3.5.2].

Lemma 4.3. *Let $g(t) = \sum_{i=0}^m b_i t^i$ be defined as in* (12). *We get:*

$$|b_i| \le \binom{m-1}{i-1} B + \binom{m-1}{i} \ (1 \le i < m) \text{ with } B = \prod_{j=1}^n \max(1, |\alpha_j|).$$

¿From the construction of g we know that $b_m = 1$ and $b_0 = \pm f(0)$. Supposing the knowledge of an upper bound for B it is easy to compute an upper bound for the absolute size of the coefficients of g. One way is to compute approximations of the roots of f in $\mathbb{C}$ to derive a bound B. If we do not want to compute the zeros of f in $\mathbb{C}$ we can use an estimation of Mignotte [16, Theorem 1].

Lemma 4.4. *Let $f(t) = \sum_{i=0}^n a_i t^i \in \mathbb{C}[t]$ with zeros $\alpha_1, \dots, \alpha_n \in \mathbb{C}$. Then we have:*

$$\prod_{i=1}^n \max(1, |\alpha_i|) \le \sqrt{\sum_{i=0}^n |a_i|^2}.$$

It remains to discuss the case when g is not irreducible, i.e. g has multiple roots. As remarked above, we use linear transforms on f: $f(t) \leftarrow f(t+a)$. The next lemma shows that this procedure will yield irreducible polynomials g.

Lemma 4.5. *There are at most n linear substitutions to f such that the constructed polynomial g (12) has multiple roots.*

Proof. For $1 \leq i \leq m$ we define:

$$\Phi_i(x) := \prod_{\gamma \in \Delta_i} (x + \gamma).$$

These polynomials are pairwise distinct since they have different zeros. All polynomials have degree d. This means that at most d evaluations of two polynomials can coincide. If the δ_i in (12) are not pairwise distinct, then each δ_i is a multiple root since g is a characteristic polynomial. Therefore there are at most $d(m-1) = n - d$ evaluations values $a \in \mathbb{Z}$ such that $\Phi_1(a) = \Phi_i(a)$ for $2 \leq i \leq m$. If we choose another $a \in \mathbb{Z}$ for the transformation we get that all δ_i are pairwise distinct. $\square$

This lemma remains valid if the ground field is a finite field. We need the additional assumption that the finite field contains enough elements. The following lemma is an immediate consequence.

Lemma 4.6. *Let $p > n$ and suppose that $p \nmid \mathrm{disc}(f)$. Then there are at most n linear substitutions for f such that $p \mid \mathrm{disc}(g)$.*

For our embedding algorithm it is important to have $p \nmid \mathrm{disc}(g)$. Therefore we choose primes $p > n$ in our algorithm.

Now we give an algorithm to compute generating polynomials for the subfields corresponding to a block system.

Algorithm 4.7. (ComputeGeneratingPolynomial)

Input: *A generating polynomial f of a number field E. A prime $p > n$ and a potential block system $\Delta_1, \dots, \Delta_m$ in polynomial representation.*

Output: *A generating polynomial g of a potential subfield L, or the message, that $\Delta_1, \dots, \Delta_m$ is not a block system.*

Step 1: *Compute the inertia degrees k_i $(1 \leq i \leq m)$ of the blocks $\Delta_1, \dots, \Delta_m$.*

Step 2: *Set $l := \mathrm{lcm}(k_1, \dots, k_m)$.*

Step 3: *Compute with Lemma 4.3 a bound M for the absolute size of the coefficients of g.*

Step 4: *Factorize $f \equiv f_1 \cdots f_r \bmod \mathfrak{p}^k$ over an unramified p-adic extension of degree l of $\mathbb{Q}_p$, where $p^k > 2M$.*

Step 5: *Set $\tilde{\Delta}_j := \{f_i \mid 1 \leq i \leq r$, it exists a $\bar{f} \in \Delta_j$ with $(f_i \bmod \mathfrak{p}) | \bar{f}\}(1 \leq j \leq m)$.*

Step 6: *For $i = 1, \dots, m$ compute the product δ_i of the zeros, which are contained in Δ_i.*

Step 7: *Compute $\sum_{i=1}^{m} \delta_i$ (modulo p^k). If the absolute value of this sum is larger than M, go to step 12.*

Step 8: *Compute $g(t) := \prod_{i=1}^{m}(t - \delta_i)$ (modulo p^k).*

Step 9: *If the absolute value of one of the coefficients of g is larger than M, go to step 12.*

Step 10: *If g modulo p has multiple factors, set $f(t) := f(t+1)$ and go to step 3.*

Step 11: *Compute $\hat{f}(t) := f(t+1)$, $\hat{\delta}_i := \prod_{\gamma \in \Delta_i}(\gamma - 1)$, $\hat{g}(t) := \prod_{i=1}^{m}(t - \hat{\delta}_i)$ and a bound $\hat{M}$ for the coefficients of $\hat{g}$. Test, if the absolute size of coefficients of $\hat{g}$ are smaller than $\hat{M}$. In this case print potential generating polynomial g and terminate.*

Step 12: *Print, that $\Delta_1, \dots, \Delta_m$ is not a block system and terminate.*

The correctness of the algorithm follows from the above considerations. We remark that it is advisable to store a lot of values. The inertia degrees of the potential block systems are already known. The bound M in step 3 only depends on f and the degree of the subfield.

The most critical part of the algorithm is the factorization of f over an unramified p-adic extension of degree l. It is important to compute this factorization only once and store the result for further use. An other question is how to choose k in step 4. Since we use quadratic lifting it is useful to choose k of the form $2^{\tilde{k}}$. It is necessary to choose k in a way that $p^{2^k} > 2M$. But practical experience shows that it is better to choose k such that $p^k \approx M^4$ holds. The reason is that we have a better chance to detect in step 7 or 9 that $\Delta_1, \dots, \Delta_m$ is not a block system. We already remarked that it is possible to detect a "wrong" block system during the embedding algorithm, but it turns out that this is very expensive. To avoid this we have inserted step 11 in the algorithm. This is another necessary condition which must hold if $\Delta_1, \dots, \Delta_m$ is a block system. We know no example that passes all these tests but it is not a block system. We use these tests only to get better running times. The results will be proved if we compute the embedding.

5. Computation of the embedding of the subfields

In this section we give an algorithm to compute an embedding of the computed potential subfields L in the given field E. As in the preceding sections let $E = \mathbb{Q}(\alpha)$, f be the minimal polynomial of α, and $\{\alpha = \alpha_1, \dots, \alpha_n\}$ be the roots of f. Furthermore let $L = \mathbb{Q}(\beta)$ and g be the minimal polynomial of β. This is not a general algorithm to test if a number field L is contained in a number field E. We use the known potential block system $\Delta_1, \ldots, \Delta_m$ to compute the embedding. If we are able to compute an embedding we have a proof that L is indeed a subfield of E. Otherwise we get a proof that the potential subfield L is no subfield. We want to compute a polynomial $h \in \mathbb{Q}[t]$ such that $h(\alpha) = \beta$. The coefficients of h are not necessarily in $\mathbb{Z}$ since in general a equation order is not integrally closed.

To simplify the notation we suppose that g has been computed without substitution of f. Then we know the following equations for the zeros $\beta_1, \ldots, \beta_m$ of g:

$$\beta_j = \prod_{\gamma \in \Delta_j} \gamma \ (1 \leq j \leq m).$$

Therefore the polynomial h has the following property:

$$h(\alpha_i) = \beta_j \text{ for } \alpha_i \in \Delta_j.$$

We know the value of h at n distinct points. Since h is of degree at most $n-1$, it is uniquely defined this way. We have computed the zeros of the blocks in an unramified p-adic extension. In a first step we want to compute a modulo p approximation which can be done in the residue class field. Let $\{\bar{\alpha}_1, \ldots, \bar{\alpha}_n\}$ be the zeros of $\bar{f}$ in a suitable finite field $\tilde{\mathbb{F}_{\tilde{q}}}$. Now we can compute a modulo p approximation of h by solving a linear system of equations or by using the formula of Lagrange. Both methods have the disadvantage that it is necessary to compute all roots of $\bar{f}$ in $\tilde{\mathbb{F}_{\tilde{q}}}$. In the above algorithms we have worked in extensions $\mathbb{F}_q/\mathbb{F}_p$ of degree $l = \mathrm{lcm}(k_i)$ which is in general less than the degree of $\tilde{\mathbb{F}_{\tilde{q}}}/\mathbb{F}_p$. Now we give a method to compute a modulo p approximation for h which only needs a factorization of $\bar{f}$ in $\mathbb{F}_q[t]$. Let $\Delta_1, \ldots, \Delta_m$ be the potential block system in polynomial representation. That means that all zeros of one polynomial in Δ_i lie in the same block. Thus we are able to compute the following block polynomials:

$$a_j(t) := \prod_{\bar{\alpha} \in \Delta_j} (t - \bar{\alpha}) \in \mathbb{F}_q[t] \text{ and } b_j(t) := \prod_{1 \leq i \leq m, i \neq j} a_i(t) \in \mathbb{F}_q[t] \ (1 \leq j \leq m).$$

We denote with $\bar{\beta}_j$ the zeros of $\bar{g} \equiv g \bmod p$. Now we compute with the extended Euclidean algorithm for polynomials over finite fields polynomials $c_j,\ d_j \in \mathbb{F}_q[t]$ with

$$a_j c_j + b_j d_j = 1 \ (1 \leq j \leq m).$$

Now we define:

$$h_0(t) := \sum_{j=1}^{m} b_j(t) d_j(t) \bar{\beta}_j. \tag{13}$$

For $\bar{\alpha}_i \in \Delta_j$ and each $\tilde{j} \neq j$ we have: $b_{\tilde{j}}(\bar{\alpha}_i)d_{\tilde{j}}(\bar{\alpha}_i)\bar{\beta}_j = 0$. Thus we get: $h_0(\bar{\alpha}_i) = b_j(\bar{\alpha}_i)d_j(\bar{\alpha}_i)\bar{\beta}_j = (1 - a_j(\bar{\alpha}_i)c_j(\bar{\alpha}_i))\bar{\beta}_j = \bar{\beta}_j$, since $a_j(\bar{\alpha}_i) = 0$.

The last thing to do is to give a bound for the coefficients of h. Since the coefficients are in $\mathbb{Q}$ we need a bound for the absolute values of the denominator and numerator of the coefficients. A proof of the following lemma can be found in [8, 13, 19].

Lemma 5.1. *The absolute values of the numerators of h are less than M with*

$$M := |\beta|_\infty n(n-1)^{(n-1)/2}|\alpha|_\infty^{n(n-1)/2},$$

where $|\beta|_\infty$ and $|\alpha|_\infty$ denote the biggest absolute value of a zero of g resp. f. The absolute value of the denominators of h is bounded by $\sqrt{|\operatorname{disc}(f)|}$.

Now we are able to give the algorithm.

Algorithm 5.2. (Compute Embedding)

Input: *Generating polynomial f of a field E. Polynomial g of a potential subfield L computed with algorithm 4.7. Corresponding potential block system $\Delta_1, \dots, \Delta_m$ in polynomial representation and $p \in \mathbb{P}$ with $p \nmid \operatorname{disc}(f)\operatorname{disc}(g)$.*

Output: *Embedding polynomial $h \in \mathbb{Q}[t]$, if L is a subfield of E, otherwise the message that $\Delta_1, \dots, \Delta_m$ is not a block system.*

Step 1: *Compute h_0 with formula* (13).

Step 2: *Set $\beta_0 \equiv h_0(\alpha) \bmod p$.*

Step 3: *Compute M with Lemma 5.1 and a $k \in \mathbb{N}$, such that $p^{2^k} > 2M$.*

Step 4: *Compute using Newton lifting 2.13 an element β with $g(\beta) = 0$. If β is not computable, return that $\Delta_1, \dots, \Delta_m$ is not a block system.*

Step 5: *Compute $h \in \mathbb{Q}[t]$ with $h(\alpha) = \beta$ and print h.*

6. The whole algorithm

Now we are able to give the whole algorithm to compute subfields of degree m.

Algorithm 6.1. (Computation of subfields of degree m.)

Input: *A generating polynomial f of a number field E and a degree m.*

Output: *The list of all subfields L of E of degree m given by (g, h)*

Step 1: *Set $n := \deg(f)$ and choose a prime $p > n$ not dividing the discriminant of f.*

Step 2: *Set* $L :=$ *ComputePotentialBlockSystems*$(f, \frac{n}{m}, p)$.

Step 3: *If some block systems are known, call the Intersection algorithm 3.22 to reduce* L.

Step 4: *Set result*$:= \emptyset$.

Step 5: *For each* B *in* L *do*
1. *Set* $g :=$ *ComputeGeneratingPolynomial*(f, p, B).
2. *If* g *is a (potential) generating polynomial then set* $h :=$ *ComputeEmbedding*(f, g, B, p).
3. *If* (g, h) *defines a subfield, add it to result.*
4. *Call the Intersection algorithm to reduce* L.

Step 6: *Print result.*

In general the above algorithm works for every prime $p > n$ not dividing the discriminant. The running time of the algorithm depends strongly on the choice of the prime. When choosing the prime we have to consider two points, the number of potential block systems and the degree of the p-adic fields. Unfortunately the number of potential block systems decreases if the degree of the p-adic fields increases. In our implementation we choose the prime p in a such way that the number of potential block systems is minimal. In most cases this seems to be the best choice.

To generate all potential block systems in Step 2 it is not a good idea. In order to avoid memory problems it is better to divide the computation of potential block systems in packages. First we apply Steps 3-5 to the potential block systems of the first package, then to the second package and so on. In our implementation we use the output of Algorithm 3.14 as a package. This has the advantage that the intersection algorithm can easily be applied to such a package.

7. Connections between block systems and prime ideal decomposition

In this section we give a connection between the prime ideal decomposition of a prime ideal in $\mathfrak{o}_L$ and the corresponding block system. This is not used in the presented subfield algorithm. It gives a deeper insight in the properties of subfields. Furthermore it explains the name inertia degree for the k_i corresponding to a block. The following connection is very useful if we want to compute special subfields. For instance if we only want to compute normal subfields the following shows that all inertia degrees of a block system must be the same.

Let $\Delta_1, \ldots, \Delta_m$ be a block system of $G = \mathrm{Gal}(f)$ and p a prime with $p \nmid \mathrm{disc}(f)$. Let $\pi = \pi_1 \cdots \pi_u$ be the corresponding permutation $(\mathrm{Gal}(\bar{f}) =$

$\langle\pi\rangle$). The block system does not depend on π, but the block clusters do. We proved that all blocks in a block cluster have the same inertia degree.

Theorem 7.1. *With the above notations it follows that $p\mathfrak{o}_L = \mathfrak{p}_1 \cdots \mathfrak{p}_r$, where r is the number of block clusters corresponding to π. The inertia degrees of the block clusters coincide with the inertia degrees of the prime ideals $\mathfrak{p}_i$ $(1 \leq i \leq r)$.*

Proof. Let $\tilde{g} = \prod_{i=1}^{m}(t - \tilde{\delta}_i))$ with $\tilde{\delta}_i = \prod_{\gamma \in \tilde{\Delta}_i} \gamma$ as defined in (11). The number and the degree of the factors of $\tilde{g} \in \mathbb{Z}_p[t]$ coincide with the number and the inertia degrees of the prime ideals of $\mathfrak{o}_L$ over p. Let $\tilde{\Delta}_1, \ldots, \tilde{\Delta}_s$ be an arbitrary block cluster of the block system of inertia degree k. We must show that $\tilde{g}_1 = \prod_{i=1}^{s}(t - \tilde{\delta}_i) \in \mathbb{Z}_p[t]$ is irreducible. From the supposition we know that the $\tilde{\delta}_i$ are pairwise distinct. Let σ be the Frobenius automorphism of an unramified extension of degree k over $\mathbb{Q}_p$. Then we get (if we sort the roots), that $\tilde{\delta}_i = \sigma^{i-1}(\tilde{\delta}_1)$ for $1 \leq i \leq s$ holds. This proves that $\tilde{g}_1 \in \mathbb{Z}_p[t]$ is irreducible and the corresponding prime ideal has inertia degree k. $\square$

8. Examples

In this section we give several examples demonstrating the efficiency of our algorithm. These algorithms were implemented in the computer algebra system KASH [6]. All computations were done on HP 9000/735 under HP-UX 9.05.

First we compare the running times with the algorithms presented in [10, 12]. This demonstrates the development of the subfield algorithm. Other methods [9, 1, 15] were compared in [12] resp. [9]. It turned out that the methods in [12] are much more efficient than the other ones.

First we compare this algorithm with the algorithm developed by the author in his master thesis [10]. We have computed the subfields of 1112 imprimitive fields of degree 9. These fields have been taken from a table of [7]. Explicit examples are given in [10]. We only give the running times. We denote with r_1 the number of real zeros.

r_1	Number fields	Number subfields	Running time old	Running time new	Average running time old	Average running time new
1	485	486	36:43 min	120 sec	4,5 sec	0,25 sec
3	423	446	31:25 min	88 sec	4,5 sec	0,21 sec
5	154	154	9:38 min	31 sec	3,8 sec	0,20 sec
7	23	23	1:30 min	5,7 sec	3,9 sec	0,25 sec
9	27	31	1:39 min	7,2 sec	3,7 sec	0,27 sec

The following table can be found in [9]. In this article a lot of subfield algorithms were compared. In [12] it has been shown that the other methods are limited to small examples. We only compare our algorithm (new) with the one presented in [12] (old).

No	Polynomial	old	new
1	$t^6 + 108$	1,1 sec	0,2 sec
2	$t^8 - 12t^6 + 23t^4 - 12t^2 + 1$	4,0 sec	0,6 sec
3	$t^8 - 10t^4 + 1$	1,5 sec	0,4 sec
4	$t^8 + 4t^6 + 10t^4 + 12t^2 + 7$	1,8 sec	0,4 sec
5	$t^9 - 18t^8 + 117t^7 - 348t^6 + 396t^5 + 288t^4 + 3012t^3 + 576t^2 + 576t - 512$	3,3 sec	0,7 sec
6	$t^{10} + 38t^9 - 99t^8 + 1334t^7 - 4272t^6 + 9244t^5 - 8297t^4 + 1222t^3 + 1023t^2 - 74t + 1$	3,4 sec	3,5 sec
7	$t^{10} - 20t^9 + 80t^8 + 200t^7 - 3770t^6 + 872t^5 + 29080t^4 + 36280t^3 - 456615t^2 + 541260t - 517448$	3,9 sec	1,9 sec
8	$t^{10} - 10t^8 + 20t^7 + 235t^6 + 606t^5 + 800t^4 + 600t^3 + 270t^2 + 70t + 16$	3,2 sec	0,7 sec
9	$t^{12} + 6t^9 + 4t^8 + 8t^6 - 4t^5 - 12t^4 + 8t^3 - 8t + 8$	7,4 sec	0,8 sec
10	$t^{12} + 9t^{11} + 3t^{10} - 73t^9 - 177t^8 - 267t^7 - 315t^6 - 267t^5 - 177t^4 - 73t^3 + 3t^2 + 9t + 1$	14 sec	9,7 sec
11	see below	98 sec	15 sec
12	$t^{15} + 20t^{12} + 125t^{11} + 503t^{10} + 1650t^9 + 3430t^8 + 4690t^7 + 4335t^6 + 2904t^5 + 1400t^4 + 485t^3 + 100t^2 + 15t + 1$	10 sec	8,6 sec

The eleventh polynomial in the table has the following form:

$$t^{12} - 34734t^{11} + 401000259t^{10} - 1456627492885t^9 - 2537142937228035t^8 + 18762072755679375516t^7 - 812368636358864062944t^6 - 7013286362975825751223193 1t^5 + 2583447251489310233282106208 5t^4 + 76623280610352450247247939584745t^3 - 450808850154226621325157634997584 50t^2 - 20704995522408122142883169810718189 00t - 5505057590977785454853648262467535 44$$

An other example which was computed in [12] is a field $E/\mathbb{Q}$ of degree 24 with Galois group $\mathfrak{S}_4$. The field is generated by a root of

$$f(t) = t^{24} + 8t^{23} - 32t^{22} - 298t^{21} + 624t^{20} + 4592t^{19} - 8845t^{18} - 31488t^{17} + 76813t^{16} + 65924t^{15} - 265616t^{14} + 48348t^{13} + 385639t^{12} - 394984t^{11} - 20946t^{10} + 369102t^9 - 362877t^8 + 183396t^7 + 434501t^6 - 194418t^5 + 450637t^4 + 125800t^3 - 16401t^2 - 45880t + 115151.$$

A list of generating polynomials can be found in [12]. The running time there was 3641 sec. Now we are able to compute all subfields within 105 sec.

Now we look at an example with a huge number of potential block systems. The following field E of degree 60 was computed as splitting field of a field of degree 5 with Galois group $\mathfrak{A}_5$. The main problem is neither the degree nor the size of the coefficients. There are only cycle decompositions with small cycles. We have the following factorization shapes:

1. 60 factors of degree 1,
2. 30 factors of degree 2,
3. 20 factors of degree 3,
4. 12 factors of degree 5.

There are no subfields of degree 2,3, and 4, which can be figured out easily. If we choose a prime corresponding to 12 factors of degree 5, we have to consider 5^{11} potential block systems to compute subfields of degree 5. Without any additional information this would take about half a year computing time. We are able to complete this example if we know some subfields. With this information we can compute block systems and use the intersection algorithm 3.22.

To compute the splitting field of degree 60 we started with a field of degree 5 generated by a zero of $t^5 + t^4 - 2t^3 + t^2 + t + 1$. If we factor this polynomial over the number field generated by a root of it, we get a degree 4 factor. Now we computed a primitive element for the degree 20 extension. After this we used the OrderShort function of KASH [6] to compute a shorter representation. This function works in a similar way to the function polred in PARI. We have the following polynomial:

$$t^{20}+8t^{19}+13t^{18}-47t^{17}-136t^{16}-23t^{15}+451t^{14}+761t^{13}+640t^{12}-9t^{11}-390t^{10}-648t^{9}-396t^{8}-684t^{7}+36t^{6}+162t^{5}+270t^{4}-243t^{3}+405t^{2}-81t+81.$$

As a last step we computed the degree 60 polynomial. An important fact is that we are able to compute the embeddings of the degree 5 and 20 fields into E. The field E is generated by a zero of

$$t^{60}+36t^{59}+579t^{58}+5379t^{57}+30720t^{56}+100695t^{55}+98167t^{54}-611235t^{53}-2499942t^{52}-1083381t^{51}+15524106t^{50}+36302361t^{49}-22772747t^{48}-205016994t^{47}-194408478t^{46}+417482280t^{45}+954044226t^{44}+281620485t^{43}-366211766t^{42}-1033459767t^{41}-8746987110t^{40}-15534020046t^{39}+23906439759t^{38}+104232578583t^{37}+31342660390t^{36}-364771340802t^{35}-547716092637t^{34}+583582152900t^{33}+2306558029146t^{32}+998482693677t^{31}-3932078004617t^{30}-5195646620046t^{29}+2421428069304t^{28}+10559164336236t^{27}+3475972372302t^{26}-22874708335419t^{25}-33428241525914t^{24}+21431451023271t^{23}+90595197659892t^{22}+50882107959528t^{21}-67090205528313t^{20}-117796269461541t^{19}-74369954660792t^{18}+25377774560496t^{17}+126851217660123t^{16}+104232393296166t^{15}-29072256729168t^{14}-83163550972215t^{13}-24296640395870t^{12}+14633584964262t^{11}+8865283658688t^{10}+5364852154893t^{9}-1565702171883t^{8}-7601782249737t^{7}-2106132289551t^{6}+3369356619543t^{5}+3717661159674t^{4}+1754791133184t^{3}+573470363592t^{2}+74954438640t+3285118944$$

To save space we do not give the subfields here. In the following table we give a statistic about the number of subfields and the running times.

The running time for the subfields increases if the degree of the subfields becomes larger. The reason is that the embedding algorithm becomes more expensive. The exception is the degree 5 case. At this point only two subfields are known which means that many potential block systems must be tested. The computation of the field E including the embeddings of the two known subfields took about one hour.

degree	Number subfields	time whole	time embedding
2	0	21 sec	
3	0	61 sec	
4	0	142 sec	
5	5	2339 sec	610 sec
6	6	1415 sec	859 sec
10	10	2476 sec	2383 sec
12	6	4211 sec	1696 sec
15	5	2459 sec	1790 sec
20	10	6831 sec	4743 sec
30	15	12516 sec	10827 sec
All	57	≈ 9 h	≈ 6 h 22 min

References

[1] D. Casperson, D. Ford, J. McKay, *Ideal decompositions and subfields.* J. Symbolic Comput. **21** (1996), 133–137.

[2] J.W.S. Cassels, *Local Fields.* Cambridge University Press, 1986.

[3] H. Cohen, F. Diaz y Diaz, *A polynomial reduction algorithm.* Sém. Théor. Nombres Bordeaux (2) **3** (1991), no. 2, 351–360.

[4] Henri Cohen, *A Course in Computational Algebraic Number Theory.* Graduate Texts in Mathematics, **138**. Springer-Verlag, Berlin, 1993.

[5] G.E. Collins, M.E. Encarnación, *Efficient rational number reconstruction.* J. Symbolic Comput **20** (1995), 287–297.

[6] Mario Daberkow, Claus Fieker, Jürgen Klüners, Michael Pohst, Katherine Roegner, Klaus Wildanger, KANT V4. J. Symbolic Comput. **24** (1997), 267–283.

[7] F. Diaz y Diaz, M. Olivier, *Imprimitive ninth-degree number fields with small discriminants.* Math. Comput. **64** (1995), no. 209, 305–321.

[8] J. Dixon, *Computing subfields in algebraic number fields.* J. Austral. Math. Soc. Ser. A **49** (1990), 434–448.

[9] A. Hulpke, *Block systems of a Galois group.* Experiment. Math. **4** (1995), no. 1, 1–9.

[10] J. Klüners, *Über die Berechnung von Teilkörpern algebraischer Zahlkörper.* Diplomarbeit, Technische Universität Berlin, 1995.

[11] J. Klüners, *Über die Berechnung von Automorphismen und Teilkörpern algebraischer Zahlkörper.* Dissertation, Technische Universität Berlin, 1997.

[12] J. Klüners M. Pohst, *On computing subfields.* J. Symbolic Comput. **24** (1997), 385–397.

[13] S. Landau, *Factoring polynomials over algebraic number fields.* SIAM J. Comput. **14** (1985), no. 1, 184–195.
[14] S. Landau, G.L. Miller, *Solvability by radicals is in polynomial time.* J. Comput. System Sci. **30** (1985), no. 2, 179–208.
[15] D. Lazard, A. Valibouze, *Computing subfields: Reverse of the primitive element problem.* In A. Galligo F. Eyssete, editor, *MEGA-92, Computational algebraic geometry*, volume 109, pages 163–176. Birkhäuser, Boston, 1993.
[16] M. Mignotte, *An inequality about factors of polynomials.* Math. Comput. **28** (1974), no. 128, 1153–1157.
[17] W. Narkiewicz, *Elementary and Analytic Theory of Algebraic Numbers.* Springer-Verlag, 1990.
[18] M. E. Pohst, H. Zassenhaus, *Algorithmic Algebraic Number Theory.* Encyclopedia of Mathematics and its Applications, **30**. Cambridge University Press, Cambridge 1989
[19] P.J. Weinberger, L. Rothschild, *Factoring polynomials over algebraic number fields.* ACM Trans. Math. Software **2** (1976), no. 4, 335–350.
[20] H. Wielandt, *Finite Permutation Groups.* Academic Press, New York-London 1964.

Jürgen Klüners
IWR, Universität Heidelberg, Im Neuenheimer Feld 368
69 120 Heidelberg, Germany
E-mail : klueners@iwr.uni-heidelberg.de