Algorithms for Function Fields

Jürgen Klüners

CONTENTS

1. Introduction
2. Notation
3. Newton Lifting and Reconstruction
4. Automorphisms
5. Embedding of Subfields
6. Zeros of Polynomials in $\mathbb{Z}$[t][x]
7. Subfields
8. Rational Decompositions
9. The Computation of Subfields of a Splitting Field
10. Examples
Acknowledgements
References

2000 AMS Subject Classification: Primary 11Y40; Secondary 11-04, 12E05, 12F10

Keywords: Galois groups, subfields, decompositions, algorithms

Let $K/\mathbb{Q}(t)$ be a finite extension. We describe algorithms for computing subfields and automorphisms of $K/\mathbb{Q}(t)$. As an application we give an algorithm for finding decompositions of rational functions in $\mathbb{Q}(\alpha)$. We also present an algorithm which decides if an extension $L/\mathbb{Q}(t)$ is a subfield of K. In case $[K : \mathbb{Q}(t)] = [L : \mathbb{Q}(t)]$ we obtain a $\mathbb{Q}(t)$-isomorphism test. Furthermore, we describe an algorithm which computes subfields of the normal closure of $K/\mathbb{Q}(t)$.

1. INTRODUCTION

Let $K/\mathbb{Q}(t)$ be a finite extension of a function field. In this paper, we develop algorithms for deciding if $K/\mathbb{Q}(t)$ is a normal or even an abelian extension. In this case, we give a method for computing all automorphisms of $K/\mathbb{Q}(t)$. Another problem we consider is the determination of all intermediate fields of $K/\mathbb{Q}(t)$. Here it is not necessary to assume that $K/\mathbb{Q}(t)$ is a normal extension.

As an application, we show how to obtain decompositions of rational functions using the fact that rational functions correspond to rational function fields. Furthermore, we give an explicit description of the main algorithm in [Klüners and Malle 00] in the function field case. This yields a method for computing subfields of the splitting field of a finite extension of $\mathbb{Q}(t)$.

All algorithms presented in this paper are based on the following idea: Let $f \in \mathbb{Z}[t][x]$ be the minimal polynomial of a primitive element of $K/\mathbb{Q}(t)$. Then by Hilbert's irreducibility theorem, there are infinitely many specializations $t_0 \in \mathbb{Z}$ such that $\bar{f}(x) := f(t_0, x) \in \mathbb{Z}[x]$ is irreducible as well. After finding such a t_0, we solve the corresponding problem in the residue class field and then use lifting procedures to get the solution of our initial problem. In contrast to the case of global fields, we have the advantage that in the generic case the Galois group of the residue class field is the same as the Galois group of the given field.

In this paper, we assume that the corresponding problems can be solved in the number field case. Algorithms

for the computation of subfields of algebraic number fields are described in [Klüners and Pohst 97, Klüners 98]. In [Acciaro and Klüners 99, Klüners 97] algorithms for the computation of automorphisms of algebraic number fields are explained.

All algorithms are implemented in the computer algebra system KANT [Daberkow et al. 97]. We give several examples to demonstrate the efficiency of the algorithms.

2. NOTATION

We consider finite extensions of $\mathbb{Q}(t)$. We assume that these extensions are given by a primitive element α with minimal polynomial f of degree n. By applying suitable transformations, we can assume that f is a monic polynomial in $\mathbb{Z}[t][x]$. The stem field $\mathbb{Q}(t)(\alpha)$ of f is denoted by K and the splitting field of f is denoted by N. The zeros of f in N are denoted by $\alpha = \alpha_1, \alpha_2, \ldots, \alpha_n$. Throughout, $G = \mathrm{Gal}(f)$ is the Galois group of f acting on the roots $\alpha_1, \ldots, \alpha_n$.

In our algorithmic approach, we need to consider residue class fields. Therefore, let $t_0 \in \mathbb{Z}$ be chosen in such a way that $\bar{f}(x) := f(t_0, x) \in \mathbb{Z}[x]$ is irreducible. We denote by $\bar{\cdot}$ the corresponding structures in the residue class field, i.e., $\bar{K}$ denotes a stem field of $\bar{f}$, $\bar{N}$ the splitting field of $\bar{f}$. $\bar{G}$ is the Galois group of $\bar{f}$ acting on the roots $\bar{\alpha} = \bar{\alpha}_1, \bar{\alpha}_2, \ldots, \bar{\alpha}_n$.

3. NEWTON LIFTING AND RECONSTRUCTION

Let R be a commutative ring with 1 and $\mathfrak{a}$ an ideal of R. Furthermore, let $g \in R[x]$ be a polynomial and $\beta_0 \in R$ such that $g(\beta_0) \equiv 0 \bmod \mathfrak{a}$ and $g'(\beta_0)$ is invertible modulo $\mathfrak{a}$. Then for every $k \in \mathbb{N}$, we can compute β_k such that $\beta_k \equiv \beta_0 \bmod \mathfrak{a}$ and $g(\beta_k) \equiv 0 \bmod \mathfrak{a}^{2^k}$ using the extended Newton lifting which avoids divisions. Here we only give the algorithm. Details can be found in [von zur Gathen and Gerhard 99, Algorithm 9.22]. Let ω_0 be the inverse of $g'(\beta_0)$ modulo $\mathfrak{a}$. Then we can use the following double iteration for $i = 0, \ldots, k-1$:

$$\beta_{i+1} \equiv \beta_i - \omega_i g(\beta_i) \bmod \mathfrak{a}^{2^{i+1}}; \quad (3\text{–}1)$$

$$\omega_{i+1} \equiv \omega_i[2 - \omega_i g'(\beta_{i+1})] \bmod \mathfrak{a}^{2^{i+1}}. \quad (3\text{–}2)$$

Let f, α, K, and n be defined as in Section 2. In the following, we look at the special situation where R is the equation order $\mathbb{Q}[t][\alpha] := \mathbb{Q}[t] + \mathbb{Q}[t]\alpha + \cdots + \mathbb{Q}[t]\alpha^{n-1}$ and $\mathfrak{a} := (t - t_0) \subseteq R$ is the principal ideal generated by $t - t_0 \in \mathbb{Z}[t]$.

Lemma 3.1. (Newton lifting.) *Let $g \in \mathbb{Z}[t][x]$ be a polynomial, $t_0 \in \mathbb{Z}$, and $\beta_0 \in \mathbb{Q}[t][\alpha]$ such that $g(\beta_0) \equiv 0 \bmod (t - t_0)$ and $\mathfrak{a} = (t - t_0) \nmid \mathrm{disc}(f)\,\mathrm{disc}(g)$. Then for every $k \in \mathbb{N}$ we can compute an element $\beta_k \in \mathbb{Q}[t][\alpha]$ with $g(\beta_k) \equiv 0 \bmod \mathfrak{a}^{2^k}$ and $\beta_k \equiv \beta_0 \bmod \mathfrak{a}$.*

Proof: From $(t - t_0) \nmid \mathrm{disc}(f)\,\mathrm{disc}(g)$ we get that $g'(\beta_0)$ is invertible in $R/\mathfrak{a}$. Its inverse ω_0 can be computed using the extended Euclidean algorithm. The elements β_k are now obtained using the above double iteration. □

In our algorithm, we want to compute an element of the form

$$\beta = \sum_{i=0}^{n-1} b_i \alpha^i \ (b_i \in \mathbb{Q}(t)),$$

where we make the additional assumption that all denominators of the b_i divide a given polynomial $d \in \mathbb{Q}[t]$. Now let $M := t - t_0 \in \mathbb{Z}[t]$ be a polynomial which is prime to d. For $a, b, c \in \mathbb{Q}[t]$ with $\gcd(M, b) = 1$, we say that $\frac{a}{b} \equiv c \bmod (M^k)$ if and only if $a \equiv bc \bmod M^k$. We further say that

$$\sum_{i=0}^{n-1} b_i \alpha^i \equiv \sum_{i=0}^{n-1} c_i \alpha^i \bmod \mathfrak{a}^k \text{ if and only if}$$
$$b_i \equiv c_i \bmod (M^k)\ (0 \le i \le n-1).$$

In our applications, we are able to compute $\gamma = \sum_{i=0}^{n-1} c_i \alpha^i$ with $\beta \equiv \gamma \bmod \mathfrak{a}^k$. Knowing that all denominators of the b_i divide d, the reconstruction of β from γ can be done coefficientwise using the following lemma:

Lemma 3.2. (Padé approximation.) *Let $c, M = (t - t_0)^k \in \mathbb{Q}[t]$ and $k_1, k_2 \in \mathbb{N}$ with $k_1 + k_2 < k$. If there exist $a, b \in \mathbb{Q}[t]$ with $\deg(a) \le k_1$ and $\deg(b) \le k_2$ such that $\frac{a}{b} \equiv c \bmod (M)$, then a, b can be computed efficiently based on an extended gcd-algorithm. Furthermore, $\frac{a}{b}$ is unique in this case.*

The proof can be found in [von zur Gathen and Gerhard 99, Section 5.9]. If we want to use the above lemma, it is important to have estimates for the degrees of a and b in order to choose the needed precision k.

We denote by $|\cdot|_\infty$ the negated degree valuation on $\mathbb{Q}(t)$, i.e., $|\frac{a}{b}|_\infty = \deg(a) - \deg(b)$. Let $N/\mathbb{Q}(t)$ be a finite extension. We know that there exists a valuation of N extending $|\cdot|_\infty$. We denote this valuation by $|\cdot|_\infty$, too. Let $f \in \mathbb{Q}[t][x]$ be an irreducible polynomial. It is well known how to compute the valuations of the zeros of f in a splitting field N of f.

Theorem 3.3. *Let $f = x^n + a_1 x^{n-1} + \cdots + a_n \in \mathbb{Q}(t)[x]$ be a monic polynomial and denote by $\alpha_1, \ldots, \alpha_n$ the zeros in a splitting field. Then we can recursively define $1 \leq k_1 < k_2 < \ldots < k_s = n$ such that the following holds:*

(i) Let $k_1 \in \{1, \ldots, n\}$ be the largest number such that

$$\frac{|a_{k_1}|_\infty}{k_1} = \max_{1 \leq i \leq n} \frac{|a_i|_\infty}{i}.$$

Then $v_1 := \frac{|a_{k_1}|_\infty}{k_1}$ is the maximal negated degree valuation of a zero of f and there are exactly k_1 zeros with this valuation.

(ii) Supposing $k_i < n$, we define $k_{i+1} \in \{k_i + 1, \ldots, n\}$ to be the largest number such that

$$\frac{|a_{k_{i+1}}|_\infty - \sum_{\nu=1}^{i} k_\nu v_\nu}{k_{i+1} - k_i} = \max_{k_i < j \leq n} \frac{|a_j|_\infty - \sum_{\nu=1}^{i} k_\nu v_\nu}{j - k_i}.$$

Then $v_{i+1} := \frac{|a_{k_{i+1}}|_\infty - \sum_{\nu=1}^{i} k_\nu v_\nu}{k_{i+1} - k_i}$ is the maximal negated degree valuation of $k_{i+1} - k_i$ zeros of f.

Proof: Without loss of generality, we can assume that $|\alpha_1|_\infty \geq \ldots \geq |\alpha_n|_\infty$. The coefficients of f are the elementary symmetric functions in $\alpha_1, \ldots, \alpha_n$. Since $|\cdot|_\infty$ is non-archimedean, it follows that $|a_i|_\infty \leq i|\alpha_1|_\infty$ for $1 \leq i \leq k_1$. Furthermore, $|a_i|_\infty < i|\alpha_1|_\infty$ for $i > k_1$. Since there is no cancellation, we get $|a_{k_1}|_\infty = k_1|\alpha_1|_\infty$ which proves (i). The second part can be proved in an analogous way. □

Using the preceding theorem, the valuations of the zeros of a polynomial $f \in \mathbb{Q}(t)[x]$ can be computed easily.

Lemma 3.4. *Let $K = \mathbb{Q}(t)(\alpha)$ be an extension of degree n of $\mathbb{Q}(t)$ and $\beta \in K$. Furthermore, let $f \in \mathbb{Q}[t][x]$ be the minimal polynomial of α and denote by $v_j := \max(0, |\alpha_j|_\infty)$, where w.l.o.g. $\alpha_1, \ldots, \alpha_n$ are ordered in a way such that $v_1 \geq \ldots \geq v_n$. Denote by w the maximal valuation of a zero of the minimal polynomial of β over $\mathbb{Q}(t)$. Then*

$$\beta = \frac{1}{d} \sum_{i=0}^{n-1} \hat{b}_i \alpha^i, \text{ with } \hat{b}_i, d \in \mathbb{Q}[t];$$

$$|\hat{b}_i|_\infty \leq |d|_\infty - \frac{1}{2}|\operatorname{disc}(f)|_\infty + \sum_{j=1}^{n-1}(n-j)v_j + w.$$

Proof: Clearly, $\beta = \frac{1}{d}\sum_{i=0}^{n-1} \hat{b}_i \alpha^i$ for some $\hat{b}_i, d \in \mathbb{Q}[t]$. Denote by $\alpha_1, \ldots, \alpha_n$ the conjugates of α. Then the conjugates of β are given by

$$\beta_j = \frac{1}{d} \sum_{i=0}^{n-1} \hat{b}_i \alpha_j^i \ (1 \leq j \leq n).$$

This defines a linear system of equations:

$$\frac{1}{d} \begin{pmatrix} 1 & \alpha_1 & \cdots & \alpha_1^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & \alpha_n & \cdots & \alpha_n^{n-1} \end{pmatrix} \begin{pmatrix} \hat{b}_0 \\ \vdots \\ \hat{b}_{n-1} \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix}.$$

Denote by A the above Vandermonde matrix, by $A_0, \ldots, A_{n-1}$ the columns of A, and define $B := (\beta_1, \ldots, \beta_n)^{\mathrm{tr}}$. Using Cramer's rule, we obtain

$$b_i = \frac{d \det(A_0, \ldots, A_{i-1}, B, A_{i+1}, \ldots, A_{n-1})}{\det(A)}.$$

We want to estimate $\det(A_0, \ldots, A_{i-1}, B, A_{i+1}, \ldots, A_{n-1})$ using the fact that the determinant is the sum of products of n factors, where we have exactly one factor in each row and in each column. The worst case is when we place B in the first column. Using $\det(A)^2 = \operatorname{disc}(f)$ and that $|\cdot|_\infty$ is non-archimedean we get

$$|\hat{b}_i|_\infty \leq |d|_\infty - \frac{1}{2}|\operatorname{disc}(f)|_\infty + \sum_{j=1}^{n-1}(n-j)v_j + w \quad \text{for } 0 \leq i \leq n-1.$$

□

This estimate can be sharpened when f has zeros α_i with negative valuation. Now we are able to give the following algorithm:

Algorithm 3.5. (Root finding.)

Input: Minimal polynomial $f \in \mathbb{Z}[t][x]$ of a primitive element α of an extension $K/\mathbb{Q}(t)$, a polynomial $g \in \mathbb{Z}[t][x]$, $t_0 \in \mathbb{Z}$ such that $f(t_0, x)$ and $g(t_0, x)$ are irreducible, and $\bar{\beta}$ with $g(\bar{\beta}) \equiv 0 \bmod (t - t_0)$.

Output: $\beta = \sum_{i=0}^{n-1} b_i \alpha^i$ ($b_i \in \mathbb{Q}(t)$) with $g(\beta) = 0$ and $\beta \equiv \bar{\beta} \bmod (t - t_0)$, or indication that such a β does not exist.

Step 1: Compute the valuations $v_1 \geq \ldots \geq v_n$ of the zeros of f using Theorem 3.3 ($n = [K : \mathbb{Q}(t)]$) and set $v_i := \max(v_i, 0)$.

Step 2: Compute the maximal valuation w of the zeros of g using Theorem 3.3.

Step 3: Compute the discriminant of f and its factorization $\operatorname{disc}(f) = \prod_{i=1}^{r} d_i^{e_i}$ in $\mathbb{Q}[t]$. Set $d := \prod_{i=1}^{r} d_i^{\lfloor \frac{e_i}{2} \rfloor}$.

Step 4: Compute $\tilde{k} := |d|_\infty - \frac{1}{2}|\operatorname{disc}(f)|_\infty + \sum_{j=1}^{n-1}(n-j)v_j + w$. We get $|\hat{b}_i|_\infty \leq \tilde{k}$ using Lemma 3.4.

Step 5: Set $k := \tilde{k} + |d|_\infty + 1$.

Step 6: Using Newton lifting (Lemma 3.1), compute $\tilde{b}_i \in \mathbb{Q}[t]$ such that

$$g(\sum_{i=0}^{n-1} \tilde{b}_i \alpha^i) \equiv 0 \bmod (t-t_0)^k.$$

Step 7: Using Lemma 3.2, retrieve the rational coefficients $b_i \equiv \tilde{b}_i \bmod (t-t_0)^k$.

Step 8: If $\beta := \sum_{i=0}^{n-1} b_i \alpha^i$ is a zero of g, return β, otherwise return that $\beta \notin K$.

The polynomial d computed in Step 3 is a multiple of all denominators of the b_is. In case a smaller polynomial with this property is known, this can be used to improve the algorithm. We remark that Step 3 can be improved by using square-free factorization. The correctness of this algorithm follows from the considerations in this section.

4. AUTOMORPHISMS

We use the notations of Section 2 and assume that $K/\mathbb{Q}(t)$ is a normal extension of degree n. Our aim is to compute the automorphism group of $K/\mathbb{Q}(t)$. An automorphism σ of $K/\mathbb{Q}(t)$ is uniquely determined by its image:

$$\beta := \sigma(\alpha) = \sum_{i=0}^{n-1} b_i \alpha^i \text{ with } b_i \in \mathbb{Q}(t).$$

Once we know this image, it is easy to apply σ to an element $\gamma = \sum_{i=0}^{n-1} c_i \alpha^i$ with $c_i \in \mathbb{Q}(t)$, since

$$\sigma(\gamma) = \sum_{i=0}^{n-1} c_i \sigma(\alpha)^i.$$

In case we want to apply σ more than once, it is desirable to store the normal form of $\sigma(\alpha), \sigma(\alpha)^2, \ldots, \sigma(\alpha)^{n-1}$ in order to save computing time.

Later in this section, we describe how to compute one single automorphism. If we want to get the whole automorphism group A, we have to compute generators of A. Afterwards, we can apply Dimino's algorithm [Butler 91, pp. 14–23] to compute all elements of A.

By Hilbert's irreducibility theorem there exists $t_0 \in \mathbb{Z}$ such that $\bar{f}(x) := f(t_0, x) \in \mathbb{Q}[x]$ is irreducible. Then $\operatorname{Gal}(f) = \operatorname{Gal}(\bar{f})$. Denote as before by $\bar{}$ the corresponding structures in the residue class field of the prime ideal $(t-t_0)$. We obtain

$$\sigma(\alpha) = \beta = \sum_{i=0}^{n-1} b_i \alpha^i \equiv \bar{\sigma}(\bar{\alpha}) = \sum_{i=0}^{n-1} \bar{b}_i \bar{\alpha}^i \bmod (t-t_0).$$

Therefore, if we are able to compute an automorphism in a residue class field, we can apply the Newton lifting and reconstruction techniques of Section 3 to determine the corresponding automorphism of $K/\mathbb{Q}(t)$. Acciaro and Klüners [Acciaro and Klüners 99] describe how to compute automorphisms of an abelian number field. The author extended this algorithm to the nonabelian case [Klüners 97].

Now we are able to give the algorithm for computing automorphisms of finite extensions of $\mathbb{Q}(t)$.

Algorithm 4.1. (Computation of automorphisms.)

Input: Minimal polynomial $f \in \mathbb{Z}[t][x]$ of a primitive element α of a normal extension $K/\mathbb{Q}(t)$, $t_0 \in \mathbb{Z}$ such that $f(t_0, x)$ is irreducible, and an automorphism $\bar{\sigma}$ of the corresponding residue class field extension.

Output: An automorphism σ of $K/\mathbb{Q}(t)$ such that $\sigma(\alpha) \equiv \bar{\sigma}(\bar{\alpha}) \bmod (t-t_0)$.

Step 1: Call Algorithm 3.5 with f, f, t_0, and $\bar{\beta} = \bar{\sigma}(\bar{\alpha})$ and store the result in β.

Step 2: Return the corresponding automorphism σ with $\sigma(\alpha) = \beta$.

The correctness of this algorithm follows from the considerations in this section. We remark that the above algorithm can also be used to check if the extension $K/\mathbb{Q}(t)$ is normal. In the negative case, $\sum_{i=0}^{n-1} b_i \alpha^i$ fails to be a zero of f.

5. EMBEDDING OF SUBFIELDS

This situation is very similar to the one in the preceding section. Let $K = \mathbb{Q}(t)(\alpha)$ be a finite extension of degree n of $\mathbb{Q}(t)$. Furthermore, we have a field $L = \mathbb{Q}(t)(\beta)$ of degree m over $\mathbb{Q}(t)$. We denote by f and g the minimal polynomials of α and β, respectively. W.l.o.g. we assume that $f, g \in \mathbb{Z}[t][x]$. We want to decide if $L/\mathbb{Q}(t)$ is a subfield of $K/\mathbb{Q}(t)$. In the latter case, we want to determine the embedding of L in K which can be done by expressing β in terms of α:

$$\beta = \sum_{i=0}^{n-1} b_i \alpha^i.$$

Note that in the case $[K : \mathbb{Q}(t)] = [L : \mathbb{Q}(t)]$, this gives an $\mathbb{Q}(t)$-isomorphism test.

Let $t_0 \in \mathbb{Z}$ such that $\bar{f}(x) := f(t_0, x) \in \mathbb{Q}[x]$ and $\bar{g}(x) := g(t_0, x) \in \mathbb{Q}[x]$ are irreducible. Denote by $\bar{}$ the corresponding structures in the residue class field of the prime ideal $(t - t_0)$. If L is a subfield of K, it follows that $\bar{L}$ is a subfield of $\bar{K}$. We assume now that $\bar{L}$ is a subfield of $\bar{K}$ and that we are able to determine the embedding

$$\bar{\beta} = \sum_{i=0}^{n-1} \bar{b}_i \bar{\alpha}^i.$$

If L is a subfield of K, we know that there exist $b_i \in \mathbb{Q}(t)$ with

$$\beta = \sum_{i=0}^{n-1} b_i \alpha^i \equiv \bar{\beta} \bmod (t - t_0).$$

Again, we can apply the Newton lifting and reconstruction techniques of Section 3 to compute the embedding. There are algorithms to solve the subfield problem in the number field case. One possibility is to use factorization of polynomials over number fields to decide the problem. Another possibility is described in [Pohst 87]. In our context, we get this information as a part of the subfield algorithm described in Section 7. Now we state the algorithm.

Algorithm 5.1. (Subfield test.)

Input: Minimal polynomial $f \in \mathbb{Z}[t][x]$ of a primitive element α of an extension $K/\mathbb{Q}(t)$, minimal polynomial $g \in \mathbb{Z}[t][x]$ of a primitive element β of an extension $L/\mathbb{Q}(t)$.

Output: Embedding $\beta = \sum_{i=0}^{n-1} b_i \alpha^i$, or indication that L is not a subfield of K.

Step 1: Find $t_0 \in \mathbb{Z}$ such that $f(t_0, x)$ and $g(t_0, x)$ are irreducible.

Step 2: Test, if $\bar{L}$ is a subfield of $\bar{K}$. If this is the case, compute the embedding of $\bar{\beta}$. Otherwise, return that L is not a subfield of K.

Step 3: Call Algorithm 3.5 with f, g, t_0, and $\bar{\beta}$.

Step 4: In case the computation of β was successful, return the corresponding embedding. Otherwise, return that L is not a subfield of K.

The correctness of this algorithm follows from the considerations in this section.

6. ZEROS OF POLYOMIALS IN $\mathbb{Z}[t][x]$

We use the notations of Section 2. In this section, we develop a method to compute approximations to the zeros of f. It is well known that all zeros of f can be expressed as power series in $\bar{N}[[t]]$. In our applications, it is sufficient to know these series modulo t^l for a suitable $l \in \mathbb{N}$. We have the problem that computations in the splitting field $\bar{N}$ of $\bar{f}$ are not very convenient. Therefore, we embed $\bar{N}$ into some unramified p-adic extension. Let $\mathfrak{p}$ be the prime ideal of the valuation ring of this p-adic field. We approximate p-adic numbers by truncated series modulo $\mathfrak{p}^k$. The necessary p-adic arithmetic is described in [Klüners 98].

Using Newton lifting, we can express α as a power series:

$$\alpha = \bar{\alpha} + \sum_{i=1}^{\infty} \frac{a_i}{d_i} t^i, \text{ where } a_i \in \mathbb{Z}[\bar{\alpha}],\ d_i \in \mathbb{N}. \qquad (6\text{–}1)$$

Note that even if $\mathbb{Z}[\bar{\alpha}]$ is the maximal order of $\bar{K}$ the denominators d_i are not necessarily equal to 1. In the p-adic approach, it is important to find a prime p which does not divide any denominator d_i as the following lemma shows.

Lemma 6.1. *Let p be a prime which divides d_i for some $i \in \mathbb{N}$ in the above power series. Then p divides* $\operatorname{disc}(\bar{f})$.

Proof: Define $a_0 := \bar{\alpha}$, $d_0 := 1$, and $c_i := \sum_{j=0}^{i} \frac{a_j}{d_j} t^j$. Using linear Newton lifting we find that

$$c_{i+1} \equiv c_i - \frac{f(c_i)}{f'(c_0)} \bmod t^{i+2}$$

$$\text{which implies} \quad \frac{a_{i+1}}{d_{i+1}} = \frac{c_{i+1} - c_i}{t^{i+1}} \equiv \frac{\frac{-f(c_i)}{t^i}}{f'(c_0)} \bmod t.$$

We see that all d_i must divide $f'(c_0)$. Denote by N the norm function of the number field $\bar{K}$. Using $f'(c_0) \equiv \bar{f}'(\bar{\alpha}) \bmod t$ and the fact that $\operatorname{disc}(\bar{f}) = \pm \mathrm{N}(\bar{f}'(\bar{\alpha}))$, we find that all primes dividing $\bar{f}'(\bar{\alpha})$ also divide $\operatorname{disc}(\bar{f})$. □

From Equation (6–1) we know that one root α of f can be expressed as a power series in $\bar{K}[[t]]$. We use the double iteration described in Section 3 to find an approximation modulo t^l for some $l \in \mathbb{N}$. Now we describe how to get all zeros of f in a suitable completion. We start to express the zeros as power series in $\mathbb{C}[[t]]$. The following lemma is an immediate consequence of the above considerations.

Lemma 6.2. *Let $\hat{\alpha}_1, \ldots, \hat{\alpha}_n \in \mathbb{C}$ be the zeros of $\bar{f}$. For $1 \le i \le n$, define $\phi_i : \bar{K}[[t]] \to \mathbb{C}[[t]]$, $\bar{\alpha} \mapsto \hat{\alpha}_i$, $t \mapsto t$. Furthermore, let α be defined as in equation* (6–1)*. Then $\check{\alpha}_i := \phi_i(\alpha)$ $(1 \le i \le n)$ are the zeros of f in $\mathbb{C}[[t]]$.*

Using complex approximations it is very difficult to get proven results. Therefore, we only use complex approximations to get bounds for the coefficients $\frac{a_i}{d_i}$. We need to find a representation for elements in the splitting field $\bar{N}$. As suggested in [Klüners 98], we want to use p-adic approximations in unramified p-adic extensions. Let p be a prime not dividing $\operatorname{disc}(\bar{f})$. From Lemma 6.1 we know that p does not divide any denominator d_i of a coefficient of α in Equation (6–1). Now let $\mathfrak{p}$ be a prime ideal of $\mathcal{O}_{\bar{N}}$ lying above p. Therefore, $\bar{f}$ splits into linear factors over $\bar{N}_{\mathfrak{p}}$. Denote the zeros of $\bar{f}$ in $\bar{N}_{\mathfrak{p}}$ by $\bar{\alpha}_1, \ldots, \bar{\alpha}_n$. For $1 \le i \le n$ define

$$\psi_i : \bar{K}[[t]] \to \bar{N}_{\mathfrak{p}}[[t]],\ \bar{\alpha} \mapsto \bar{\alpha}_i,\ t \mapsto t \text{ and } \alpha_i := \psi_i(\alpha).$$

Then it is immediate that $\alpha_1, \ldots, \alpha_n$ are the roots of f in $\bar{N}_{\mathfrak{p}}[[t]]$ and we get the following lemma.

Lemma 6.3. *For $k, l \in \mathbb{N}$ and for $1 \le i \le n$, let*

$$\alpha_i = \sum_{j=0}^{\infty} a_{i,j} t^j \in \bar{N}_{\mathfrak{p}}[[t]] \text{ and } \tilde{\alpha}_i = \sum_{j=0}^{l-1} (a_{i,j} \bmod \mathfrak{p}^k) t^j \in \bar{N}_{\mathfrak{p}}[t].$$

Then $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_n$ are the zeros of f modulo $(t^l, \mathfrak{p}^k)$ in $\bar{N}_{\mathfrak{p}}[t]$, i.e., $f(\tilde{\alpha}_i) \equiv 0 \bmod (t^l, \mathfrak{p}^k)$.

Using the above lemma, approximations to the zeros of f can easily be computed:

(i) Compute p-adic approximations modulo $\mathfrak{p}^k$ of the zeros of $\bar{f}$.

(ii) Using Newton lifting, compute $\alpha \in \bar{K}[[t]]$ modulo t^l.

(iii) Using ψ_i and Lemma 6.3 to compute approximations modulo $(t^l, \mathfrak{p}^k)$ of the zeros of f.

The approximations to the zeros of f are used in the subfield algorithm. In the next section, we give an algorithm to compute sufficiently large k and l.

7. SUBFIELDS

The algorithm for computing subfields is more complicated than the ones presented in the preceding sections. Similar to the other algorithms, we want to use the fact that we are able to compute subfields in the residue class field which is a number field. But from this computation, we do not have enough information to lift the subfields. Therefore, we have to recall some properties of subfields. For more details see [Klüners and Pohst 97, Klüners 98].

Let G be a transitive permutation group acting on $\Omega := \{\alpha_1, \ldots, \alpha_n\}$. Recall that $\Delta \subseteq \Omega$ is called a block of size $|\Delta|$, if $\Delta^\tau \cap \Delta \in \{\emptyset, \Delta\}$ for all $\tau \in G$. The orbit of a block Δ under G is called a block system. The full set and all sets of size 1 are blocks, the so called trivial blocks. Suppose that $\alpha_1, \ldots, \alpha_n$ are the roots of an irreducible polynomial $f \in \mathbb{Z}[t][x]$ and G is the Galois group of f. Then the subfields of a stem field of f are in bijection with the groups $G_{\alpha_1} \subseteq H \subseteq G$, where G_{α_1} denotes the point stabilizer of α_1. Therefore, the following theorem establishes a bijection between subfields and block systems.

Theorem 7.1. *The correspondence $\Delta \mapsto G_\Delta := \{\tau \in G \mid \Delta^\tau = \Delta\}$ is a bijection between the set of blocks of size d which contain α and the set of subgroups of G of index $m = n/d$ containing the subgroup G_α of α.*

Proof: The proof of the theorem can be found in [Wielandt 64, Theorem 2.3]. □

We use the notation of Section 2. We want to determine the intermediate fields $\mathbb{Q}(t) < L < K$ using the correspondence to block systems. The diagram in Figure 1 illustrates the situation:

Suppose we are able to determine a block system consisting of blocks $\Delta_1, \ldots, \Delta_m$ of size d. Then we can define

$$g(t, x) := \prod_{i=1}^{m} (x - \prod_{\alpha \in \Delta_i} (\alpha + a)) \in \mathbb{Z}[t][x] \ \ (a \in \mathbb{Z}). \quad (7\text{–}1)$$

It is an immediate consequence of the definition of a block system that g has coefficients in $\mathbb{Z}[t]$. Instead of just taking products, it is possible to consider an arbitrary symmetric function of the zeros in a block. The product has

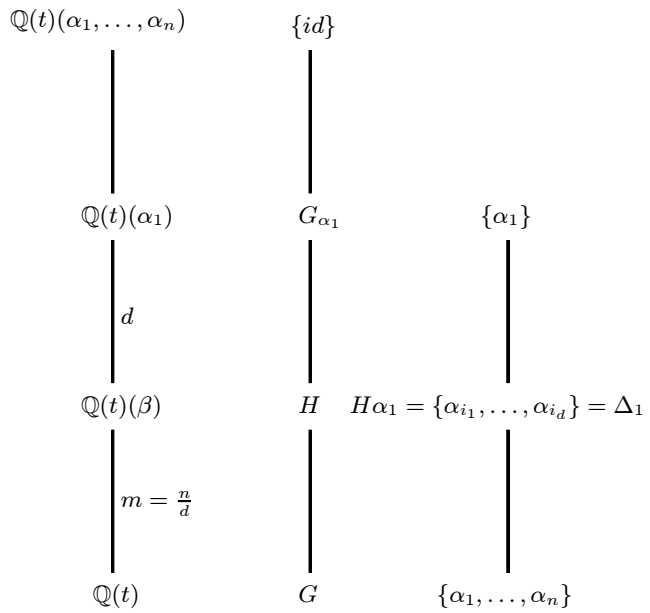

FIGURE 1.

the advantage that we can prove that at most n choices of a lead to a polynomial g which has multiple zeros, e.g., [Klüners 98, Lemma 4.5]. If the polynomial has no multiple zeros, it is irreducible and, therefore, we have found a minimal polynomial of a primitive element of the corresponding subfield L. Let $t_0 \in \mathbb{Z}$ be chosen such that $\bar{f}(x) := f(t_0, x) \in \mathbb{Z}[x]$ is irreducible. We assume w.l.o.g. that $t_0 = 0$. We denote by $\bar{G}$ the Galois group of $\bar{f}$ and by $\bar{\alpha}_1, \ldots, \bar{\alpha}_n$ the zeros of $\bar{f}$. Using the subfield algorithm for number fields, we are able to compute a block system $\bar{\Delta}_1, \ldots, \bar{\Delta}_m$. We know that the zeros of f can be expressed as power series in $\bar{N}[[t]]$, where $\bar{N}$ denotes the splitting field of $\bar{f}$. We obtain

$$\alpha_i = \bar{\alpha}_i + \sum_{j=1}^{\infty} a_{i,j} t^j, \text{ where } a_{i,j} \in \bar{N}.$$

If we are able to compute the power series (see Section 6), we can establish the correspondence between the α_i and the $\bar{\alpha}_i$. For the computation of the zeros, we have to find integers k and l such that it is sufficient to compute the zeros modulo $(t^l, \mathfrak{p}^k)$. In a first step, we give an estimate for l. As in Section 3, we denote by $|\cdot|_\infty$ the negated degree valuation on $\mathbb{Q}(t)$. For a polynomial $f(t,x) = \sum_{i=0}^{n} f_i(t)x^i \in \mathbb{Q}(t)[x]$, we denote by $|f|_\infty := \max_{0\le i\le n}(|f_i|_\infty)$ the valuation of the polynomial.

Theorem 7.2. *Let g be defined as in equation* (7–1). *Then* $|g|_\infty \le |f|_\infty$.

Proof: Assume that $a = 0$ in equation (7–1). Then

$$\begin{aligned} |g|_\infty &= |\prod_{i=1}^{m}(x - \prod_{\alpha\in\Delta_i}\alpha)|_\infty \\ &= \sum_{i=1}^{m}\max(0, \sum_{\alpha\in\Delta_i}|\alpha|_\infty) \le \sum_{i=1}^{m}\sum_{\alpha\in\Delta_i}\max(0,|\alpha|_\infty) \\ &= \sum_{i=1}^{n}\max(0,|\alpha_i|_\infty) = |\prod_{i=1}^{n}(x-\alpha_i)|_\infty = |f|_\infty. \end{aligned}$$

In case $a \neq 0$, $|\alpha_i + a|_\infty = \max(|\alpha_i|_\infty, 0)$. Therefore, the same argument shows the assertion for arbitrary a. □

Theorem 7.2 shows that we are allowed to do all computations modulo t^l, where $l = |f|_\infty + 1$. The next step is to derive a bound for the real size of the coefficients. Let

$$f(t,x) = \sum_{i=0}^{n} f_i(t)x^i \in \mathbb{Z}[t][x], \text{ where } f_i = \sum_{j=0}^{r_i} f_{i,j}t^j \in \mathbb{Z}[t].$$

We denote by $||f_i||_\infty := \max_{1\le j\le r_i}(|f_{i,j}|)$ the maximum norm of f_i and by $||f||_\infty := \max_{0\le i\le n}(||f_i||_\infty)$ the maximum norm of f. We are interested in computing a bound for $||g||_\infty$.

Theorem 7.3. *Let $f \in \mathbb{Z}[t][x]$ be a monic irreducible polynomial and denote by*

$$\alpha_i = \sum_{j=0}^{\infty} a_{i,j}t^j \in \mathbb{C}[[t]] \ (1 \le i \le n)$$

the zeros of f. Let g be defined as in Equation (7–1) *where $a = 0$ and set $l := ||f||_\infty + 1$. For $0 \le j \le l-1$, define $c_j := \max_{1\le i\le n}(\lceil |a_{i,j}| \rceil, 1)$. Define*

$$h(t) := c_0 + c_1 t + \cdots + c_{l-1}t^{l-1} \in \mathbb{Z}[t]$$

and

$$H(t,x) := (x + h(t)^{\frac{n}{m}})^m \bmod t^l.$$

Then we have $||g||_\infty \le ||H||_\infty$.

Proof: From Theorem 7.2, we know $|g|_\infty \le |f|_\infty = l - 1$. Since $|a_{i,j}| \le c_j$ for $0 \le j \le l-1$, it is immediate that $||g||_\infty \le ||H||_\infty$. □

Bounds for the c_i can be computed easily using Equation (6–1) and a bound for a maximal root of $\bar{f}$. Experience shows that c_{l-1} tends to be larger than c_0. We are now able to give the complete algorithm for computing subfields.

Algorithm 7.4. (Computation of subfields.)

Input: Minimal polynomial $f \in \mathbb{Z}[t][x]$ of a primitive element α of an extension $K/\mathbb{Q}(t)$.

Output: All subfields $\mathbb{Q}(t) < L < K$ of K described by a pair (g, β), where $g \in \mathbb{Z}[t][x]$ is the minimal polynomial of $\beta = \sum_{i=0}^{n-1} b_i \alpha^i$.

Step 1: Compute $t_0 \in \mathbb{Z}$ such that $f(t_0, x)$ is irreducible. By applying a linear transformation to f, we assume that $t_0 = 0$.

Step 2: Compute all subfields $\mathbb{Q} < \bar{L} < \bar{K}$ of $\bar{K}$ and the corresponding block systems $\bar{\Delta}_1, \ldots, \bar{\Delta}_m$. Each $\bar{L}$ is described by a pair $(\bar{g}, \bar{\beta})$, where $\bar{g} \in \mathbb{Z}[x]$ is the minimal polynomial of $\bar{\beta} = \sum_{i=0}^{n-1} \bar{b}_i \bar{\alpha}^i$.

Step 3: If there are no such $\bar{L}$, return the empty list.

Step 4: For each $\bar{L}$, do

(i) Choose a prime p such that $p \nmid \operatorname{disc}(\bar{f}) \operatorname{disc}(\bar{g})$.

(ii) Compute $l := |f|_\infty + 1$ and a bound M such that $||g||_\infty \leq M$ using Theorem 7.3.

(iii) Compute the smallest $k \in \mathbb{N}$ such that $p^k \geq 2M$.

(iv) Compute $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_n$ modulo $(t^l, \mathfrak{p}^k)$ using Lemma 6.3.

(v) Identify the $\tilde{\alpha}_i$ with the $\bar{\alpha}_i$ to compute the corresponding block system $\tilde{\Delta}_1, \ldots, \tilde{\Delta}_m$ consisting of the zeros $\tilde{\alpha}_i$.

(vi) Use Equation (7–1) to compute $g \in \mathbb{Z}[t][x]$ modulo $(t^l, p^k\mathbb{Z})$ taking the symmetric residue system modulo p^k.

(vii) Call Algorithm 5.1 with f, g to test if L is a subfield of K. If this is the case, return g and the computed embedding β.

Proof: The correctness of the algorithm follows from the above considerations. In Theorem 7.2, we proved that $|g|_\infty < l$. Therefore we can perform all computations modulo t^l. In Theorem 7.3, we showed that $||g||_\infty \leq M$. Since $p^k \geq 2M$, we can take the symmetric residue system to retrieve the true coefficients of $g \in \mathbb{Z}[t][x]$ from the computed approximations. If L is a subfield of K, $\bar{L}$ is a subfield of $\bar{K}$. The converse is not necessarily true. Therefore, in Step 4 (vi), we compute g modulo (t^l, p^k) since $\mathfrak{p}^k \cap \mathbb{Z} = p^k\mathbb{Z}$. In Step 4 (vii), we test if L is indeed a subfield of K. □

We have given a simplified version of the subfield algorithm. One improvement could be to try several $t_0 \in \mathbb{Z}$ which lead to irreducible polynomials $\bar{f}$. Afterwards, we can take the t_0 which corresponds to the field $\bar{K}$ with minimal number of subfields to avoid unnecessary callings of Algorithm 5.1.

In practice, it is important to store the zeros $\tilde{\alpha}_i$ computed in Step 4 (iv). To use the stored results, it is important to choose the same prime p for all subfields $\bar{L}$. For large examples, it is a good idea to choose the prime p in such a way that the corresponding p-adic extension $\bar{N}_\mathfrak{p}$ has small degree. In the case that the subfield algorithm over $\mathbb{Q}$ has chosen a different prime, the block systems in Step 2 can be computed using the following lemma:

Lemma 7.5. *Let $\bar{L} = \mathbb{Q}(\bar{\beta})$ be a subfield of $\bar{K} = \mathbb{Q}(\bar{\alpha})$ with corresponding minimal polynomials $\bar{g}$ and $\bar{f}$. Let $\bar{\beta} = \sum_{i=0}^{n-1} \bar{b}_i \bar{\alpha}^i$ and define $\bar{h}(x) := \sum_{i=0}^{n-1} \bar{b}_i x^i \in \mathbb{Q}[x]$. Denote by $\bar{\alpha}_1, \ldots, \bar{\alpha}_n, \bar{\beta}_1, \ldots, \bar{\beta}_m$ the zeros of $\bar{f}$ and $\bar{g}$ in a suitable closure, respectively. Define*

$$\bar{\Delta}_i := \{\bar{\alpha}_j \mid \bar{h}(\bar{\alpha}_j) = \bar{\beta}_i\}.$$

Then $\bar{\Delta}_1, \ldots, \bar{\Delta}_m$ form a block system of $\operatorname{Gal}(\bar{f})$ acting on the roots $\bar{\alpha}_1, \ldots, \bar{\alpha}_n$ corresponding to the subfield $\bar{L}$.

Proof: Let $\sigma \in \operatorname{Gal}(\bar{f})$ with $\sigma(\bar{\beta}_i) = \bar{\beta}_k$. Then

$$\bar{\gamma} \in \bar{\Delta}_i \Leftrightarrow \bar{h}(\bar{\gamma}) = \bar{\beta}_i \Leftrightarrow \sigma(\bar{h}(\bar{\gamma})) = \bar{h}(\sigma(\bar{\gamma})) = \bar{\beta}_k \Leftrightarrow \sigma(\bar{\gamma}) \in \bar{\Delta}_k.$$

Consequently, $\bar{\Delta}_1, \ldots, \bar{\Delta}_m$ is a block system. Assuming $\bar{\alpha}_1 \in \bar{\Delta}_1$, we find that the subgroups fixing $\bar{\beta}_1$ and $\bar{\Delta}_1$ coincide. Therefore, the block system $\bar{\Delta}_1, \ldots, \bar{\Delta}_m$ corresponds to $\bar{L}$. □

8. RATIONAL DECOMPOSITIONS

Let $t = \frac{a(\alpha)}{b(\alpha)} \in \mathbb{Q}(\alpha)$ with $a, b \in \mathbb{Q}[\alpha]$ monic and $\gcd(a, b) = 1$ be a rational function. Recall that the degree of a rational function $\frac{a(\alpha)}{b(\alpha)}$ is defined to be the maximum of the degrees of $a(\alpha)$ and $b(\alpha)$. It is an interesting question to determine if there exist rational functions

$u, v \in \mathbb{Q}(\alpha)$ with $1 < \deg(u), \deg(v) < \deg(t)$ such that $t = u \circ v$. It is an immediate consequence of a theorem of Lüroth (see e.g., [Jacobson 80]) that such a decomposition corresponds to a proper subfield $\mathbb{Q}(t) < L < \mathbb{Q}(\alpha)$. Therefore, it is natural to apply the subfield algorithm of the last section to compute such decompositions.

Define $f(t,x) := a(x) - tb(x) \in \mathbb{Q}[t][x]$. Since a and b have no common divisor, f has to be irreducible. Furthermore, f is the minimal polynomial of α over $\mathbb{Q}(t)$. By applying suitable transformations, we assume that f is a monic polynomial in $\mathbb{Z}[t][x]$.

Now assume that we have computed a subfield $\mathbb{Q}(t) < L < \mathbb{Q}(t,\alpha) = \mathbb{Q}(\alpha)$ using Algorithm 7.4. The algorithm returns a polynomial $g \in \mathbb{Z}[t][x]$ which is a minimal polynomial of

$$\beta = \sum_{i=0}^{n-1} b_i(t)\alpha^i,$$

where α is a zero of f. Since we know that $|f|_\infty = 1$, Theorem 6–1 implies that $|g|_\infty = 1$ as well. We remark that from Lüroth's theorem, it is clear that such a polynomial g exists, but it is not a priori clear that a general subfield algorithm will produce such a g.

Since $|g|_\infty = 1$, we can write $g(t,x) = c(x) - td(x)$ with $c, d \in \mathbb{Z}[x]$. Then for a root β of g, $t = \frac{c(\beta)}{d(\beta)}$ and $\mathbb{Q}(\beta)$ is a subfield of $\mathbb{Q}(\alpha)$ containing $\mathbb{Q}(t)$. It remains to express β as a rational function in α. We have

$$\beta = \sum_{i=0}^{n-1} b_i(t)\alpha^i.$$

Replacing t by $\frac{a(\alpha)}{b(\alpha)}$, we can express β as a rational function in α, say $\beta = \frac{\mu(\alpha)}{\nu(\alpha)}$ with $\mu, \nu \in \mathbb{Q}[\alpha]$ and $\gcd(\mu, \nu) = 1$. Altogether, this shows $\frac{a(\alpha)}{b(\alpha)} = \frac{c(\alpha)}{d(\alpha)} \circ \frac{\mu(\alpha)}{\nu(\alpha)}$.

The algorithm for rational function fields can be improved compared to the general subfield algorithm. Experiments on a computer show that the embedding part, i.e., the computation of β, is the most time consuming part. This step can be improved as follows: At some point in the computations, we know the rational functions $t = \frac{a(\alpha)}{b(\alpha)}$ and $t = \frac{c(\beta)}{d(\beta)}$ and would like to know the rational function $\beta = \frac{\mu(\alpha)}{\nu(\alpha)}$. Since

$$\frac{a(\alpha)}{b(\alpha)} = \frac{c(\beta)}{d(\beta)},$$

we consider the polynomial $a(\alpha)d(\beta) - b(\alpha)c(\beta) \in \mathbb{Q}[\alpha, \beta]$. If $\mathbb{Q}(\beta)$ is a subfield of $\mathbb{Q}(\alpha)$, this polynomial has a linear factor $\nu(\alpha)\beta - \mu(\alpha)$, where $\deg(\frac{\mu(\alpha)}{\nu(\alpha)}) = [\mathbb{Q}(\alpha) : \mathbb{Q}(\beta)]$. Therefore, we have to find linear factors in β of $a(\alpha)d(\beta) - b(\alpha)c(\beta) \in \mathbb{Q}[\alpha, \beta]$, which can be done using well-known methods.

Note that there are specialized algorithms for the rational function field case, e.g., [Alonso et al. 95]. Experiments show that the perfomance of the algorithms depends on the examples (see Section 10).

9. THE COMPUTATION OF SUBFIELDS OF A SPLITTING FIELD

In [Klüners and Malle 00, Section 3.3], we explained how to compute a subfield L of a field extension of the rationals which was given by a minimal polynomial $f \in \mathbb{Z}[x]$ of a primitive element. In the same paper, we also explained how to compute a polynomial $R_{G,H,F}[x_1, \ldots, x_n][x]$, where G is the Galois group of f, H is the stabilizer of a subfield of the splitting field, and n is the degree of f. F is a so-called H-invariant G-relative polynomial [Klüners and Malle 00, Definition 3.1]. Let $\alpha_1, \ldots, \alpha_n$ be the roots of f. Then it is shown that $R_{G,H,F}(\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}[x]$ is the characteristic polynomial of an element of L over $\mathbb{Q}$. If this polynomial is not square-free, i.e., the element is not primitive, a suitable transformation on the α_i yields a primitive element. Back to our function-field setting, we aim at computing $R_{G,H,F}(\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}[t][x]$ using approximations to the α_i as before. We have explained in Section 6. how to represent the roots α_i of a polynomial $f \in \mathbb{Z}[t][x]$. The remaining problem is to determine sufficiently large k, l (see Lemma 6.3). We have to use Theorem 3.3 to get the (degree-)valuations of the roots of f. Unfortunately, we are not in the nice situation of Theorem 7.2. After determining the degree bound, we have to compute a bound for the p-adic approximations. Let us explain this procedure with an example.

Let $f(t,x) := x^7 - 3x^6 - x^5 + 3x^4 + (-t+1)x^3 + (t+1)x^2 - 5x + 4$ be the polynomial with Galois group $G = \mathrm{PSL}_2(7)$ given in [Malle and Matzat 99, p. 405]. We want to compute one of the (isomorphic) degree 8 subfields of the splitting field of f. First we compute the following $F(x_1, \ldots, x_7) := x_1x_2x_7 + x_1x_3x_6 + x_1x_4x_5 + x_2x_3x_4 + x_2x_5x_6 + x_3x_5x_7 + x_4x_6x_7$. Denote by H a subgroup of index 8 in G and let R be a full system of representatives of (left) cosets of G/H. Furthermore, we assume that G acts in the same way on the x_i as G acts on the roots of f. Then we get

$$R_{G,H,F} = \prod_{\sigma \in R} (x - F^\sigma).$$

The next step is to compute the necessary bounds. Using Theorem 3.3, we find the degree valuations of the

roots of f: $[\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4},0,-\frac{1}{2},-\frac{1}{2}]$. Unfortunately, we have no chance to determine which root has which valuation. Since each summand of F has valuation less than or equal to $\frac{3}{4}$ (after substituting the α_is), we see that the coefficients of $R_{G,H,F}$ have valuations which are less than or equal to $8\frac{3}{4}=6$. Now we compute the zeros of f as power series in $\mathbb{C}[[t]]$ (compare Theorem 7.3). It is sufficient to compute these series modulo t^7.

The polynomial $h(t)$ in Theorem 7.3 can still be computed as before. Since F consists of seven monomials of degree 3, we define $\tilde{H}(t,x) := (x+7h(t)^3)^8 \bmod t^7$. The largest coefficient of H gives us a bound for the real norm. In our example, we get the bound 1491576722650942160 and compute everything modulo 41^{12}. The final result is the following (irreducible) polynomial:

$$x^8-18x^7+(14t+237)x^6+(-4t^2-168t-1563)x^5+(-2t^3+125t^2+2008t+9773)x^4+(-10t^3-966t^2-9231t-32724)x^3+(6t^4+383t^3+7002t^2+48745t+124283)x^2+(4t^5-38t^4-1757t^3-18994t^2-90189t-179511)x+t^6+24t^5+754t^4+8030t^3+60349t^2+226389t+576706.$$

The whole computation takes about three seconds (see next section).

10. EXAMPLES

In this section, we give the running times of some examples to demonstrate the efficiency of our algorithms. All computations were done on a 500MHz Intel Pentium III processor running under SuSE Linux 6.1.

We start with an example of degree 12. Let $K=\mathbb{Q}(t)(\alpha)$ be defined by the following minimal polynomial of α:

$$f(t,x)=x^{12}-36x^{11}+450x^{10}-2484x^9+3807x^8+25272x^7+(27t^2+299484)x^6+227448x^5+308367x^4-1810836x^3+2952450x^2-2125764x+531441.$$

This field has two proper subfields described by the following (g,β). The computations are done in 2.4 seconds.

(i) $g(t,x)=x^3+96x^2-3840x-27t^2-409600$,
$\beta=-452+936\alpha-690\alpha^2+160\alpha^3+(\frac{1}{243}t^2+\frac{33556}{243})\alpha^4+(\frac{8}{729}t^2+\frac{91544}{729})\alpha^5+\frac{293}{27}\alpha^6+\frac{284}{243}\alpha^7-\frac{686}{729}\alpha^8+\frac{388}{2187}\alpha^9-\frac{95}{6561}\alpha^{10}+\frac{8}{19683}\alpha^{11}$.

(ii) $g(t,x)=x^6-24x^5+96x^4+1024x^3-9984x^2+30720x+27t^2+409600$,
$\beta=-26+49\alpha-\frac{92}{3}\alpha^2+\frac{47}{9}\alpha^3+\frac{104}{27}\alpha^4+(\frac{1}{2187}t^2+\frac{11092}{2187})\alpha^5+\frac{104}{243}\alpha^6+\frac{47}{729}\alpha^7-\frac{92}{2187}\alpha^8+\frac{50}{6561}\alpha^9-\frac{4}{6561}\alpha^{10}+\frac{1}{59049}\alpha^{11}$.

Now let $f(t,x) := a(x)-tb(x)$ be a polynomial of degree 36, where $\frac{a(x)}{b(x)}$ is the following rational function:

$$\frac{a(x)}{b(x)} := \frac{(x^3+4)^3(x^3+6x^2+4)^3(x^6-6x^5+36x^4+8x^3-24x^2+16)^3}{(x-2)^6x^6(x+1)^3(x^2-x+1)^3(x^2+2x+4)^6}.$$

We use the methods of Section 8 to compute the rational decompositions corresponding to the subfields. There are 10 nontrivial ones and the computing time was 186 seconds. In order to save space, we only give one decomposition:

$$\frac{a(x)}{b(x)} = \frac{-(x^3-12x^2+24x-16)^3(x^3+24x-16)^3}{(x-4)^6(x-1)^3x^6} \circ \frac{-x(x-2)}{x+1}.$$

We have not used the improvements which are possible in the rational function field case as described in Section 8. Using these improvements, all decompositions can be computed within 60 seconds. The specialized package FRAC [Alonso et al. 95] needs 20 minutes for the computation of all rational decompositions.

Let $\frac{a(x)}{b(x)}$ be the rational function of degree 60 shown below. We only give one of its decompositions (which was not known before) to save space:

$$\frac{a(x)}{b(x)} := \frac{(x^4+228x^3+494x^2-228x+1)^3}{x(x^2-11x-1)^5} \circ \frac{x^4-2x^3+4x^2-3x+1}{-x(x^4+3x^3+4x^2+2x+1)}$$

We need 31 minutes to compute the three nontrivial rational decompositions. Without using the improvements of Section 8, the computation would take about 85 minutes. Here, the package FRAC needs 114 seconds to compute all rational decompositions.

ACKNOWLEDGMENT

I would like to thank John Cannon and the Magma group. I implemented most of the above algorithms during a two-month stay in Sydney.

REFERENCES

[Alonso et al. 95] C. Alonso, J. Gutierrez, and T. Recio. "A rational function decomposition algorithm by near-separated polynomials." *J. Symb. Comput.* **19**:6 (1995), 527–544.

[Acciaro and Klüners 99] V. Acciaro and J. Klüners. "Computing automorphisms of abelian number fields." *Math. Comput.* **68** (1999), 1179–1186.

[Butler 91] G. Butler. *Fundamental Algorithms for Permutation Groups*. LNCS 559. Springer, Berlin-Heidelberg, 1991.

[Daberkow et al. 97] Mario Daberkow, Claus Fieker, Jürgen Klüners, Michael Pohst, Katherine Roegner, and Klaus Wildanger. KANT V4. *J. Symb. Comput.* **24**:3 (1997), 267–283.

[Jacobson 80] N. Jacobson. *Basic Algebra II*. Freeman and Company, New York, 1980.

[Klüners 97] J. Klüners. *Über die Berechnung von Automorphismen und Teilkörpern algebraischer Zahlkörper*. Dissertation, Technische Universität Berlin, 1997.

[Klüners 98] J. Klüners. "On computing subfields—a detailed description of the algorithm." *Journal de Théorie des Nombres de Bordeaux* **10** (1998), 243–271.

[Klüners and Malle 00] Jürgen Klüners and Gunter Malle. "Explicit Galois realization of transitive groups of degree up to 15." *J. Symb. Comput.* **30** (2000), 675–716.

[Klüners and Pohst 97] J. Klüners and M. Pohst. "On computing subfields." *J. Symb. Comput.* **24**:3 (1997), 385–397.

[Malle and Matzat 99] Gunter Malle and Bernd H. Matzat. *Inverse Galois Theory*. Springer Verlag, Heidelberg, 1999.

[Pohst 87] Michael E. Pohst. "On computing isomorphisms of equation orders." *Math. Comput.* **48**:177 (1987).

[von zur Gathen and Gerhard 99] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, Cambridge, UK, 1999.

[Wielandt 64] H. Wielandt. *Finite Permutation Groups*. Academic Press, New York and London, 1964.

Jürgen Klüners, Universität Kassel, Heinrich-Plett-Str. 40, 34132 Kassel, Germany (klueners@mathematik.uni-kassel.de)

Received March 29, 2001; accepted in revised form October 17, 2001.