Functional Decomposition Using Principal Subfields

Luiz E. Allem
Univ. Federal do Rio Grande do Sul
Av. Bento Gonçalves, 9500
Porto Alegre, RS 91509-900
emilio.allem@ufrgs.br

Juliane G. Capaverde
Univ. Federal do Rio Grande do Sul
Av. Bento Gonçalves, 9500
Porto Alegre, RS 91509-900
juliane.capaverde@ufrgs.br

Mark van Hoeij
Florida State University
211 Love Building
Tallahassee, FL 32306
hoeij@math.fsu.edu

Jonas Szutkoski
Univ. Federal do Rio Grande do Sul
Av. Bento Gonçalves, 9500
Porto Alegre, RS 91509-900
jonas.szutkoski@ufrgs.br

ABSTRACT

Let $f \in K(t)$ be a univariate rational function. It is well known that any non-trivial decomposition $g \circ h$, with $g, h \in K(t)$, corresponds to a non-trivial subfield $K(f(t)) \subsetneq L \subsetneq K(t)$ and vice-versa. In this paper we use the idea of *principal subfields* and fast subfield-intersection techniques to compute the subfield lattice of $K(t)/K(f(t))$. This yields a Las Vegas type algorithm with improved complexity and better run times for finding *all* non-equivalent complete decompositions of f.

KEYWORDS

Rational Function Decomposition; Subfield Lattice; Partitions;

ACKNOWLEDGMENTS

This work is part of the doctoral studies of Jonas Szutkoski, who acknowledges the support of CAPES. Mark van Hoeij was supported by the National Science Foundation under Grant No.:1618657.

1 INTRODUCTION

The problem of finding a decomposition of a rational function $f \in K(t)$ has been studied by several authors. We highlight the work of [23], who gave the first polynomial time algorithm that finds (if it exists) a single decomposition of f. In [2], an exponential time algorithm was given that computes all decompositions of f by generalizing the ideas of [4] for the polynomial case. More recently, [3] have presented improvements on the work of [2], though the complexity is still exponential on the degree of f.

The particular case of polynomial decomposition has long been studied. As far as the authors' knowledge goes, the first work on polynomial decomposition is from [15], which presented a strong structural property of polynomial decompositions over complex numbers. In [4], two (exponential time) algorithms are presented for finding the decompositions of a polynomial over a field of characteristic zero. Some simplifications are suggested in [1, 2]. In [13], the first polynomial time algorithm is given, which works over any commutative ring containing an inverse of $\deg(g)$. Further improvements are presented in [20, 21]. More recently, [7] presented a polynomial time algorithm that finds all *minimal decompositions* of f, with no restrictions on $\deg(g)$ or the characteristic of the field.

Univariate Functional Decomposition (either rational function or polynomial) is closely related to the subfield lattice of the field extension $K(t)/K(f(t))$ (see Theorem 2.3 below). However, in general, the number of subfields is not polynomially bounded and algorithms for finding *all* complete decompositions can suffer a combinatorial explosion. In this work, we try to improve the non-polynomial part of the complexity. In order to achieve this, we make use of the so-called *principal subfields*, as defined in [19].

Let $f(t) = p(t)/q(t) \in K(t)$, $n = \max\{\deg(p), \deg(q)\}$ and $\nabla_f := p(x)q(t) - p(t)q(x) \in K[x, t]$. Assuming we are given the factorization of ∇_f, using fast arithmetic and fast subfield intersection techniques (see [17]), we can compute the subfield lattice of $K(t)/K(f(t))$ with an expected number of

$$\tilde{O}(rn^2) \text{ field operations plus } \tilde{O}(mr^2) \text{ CPU operations,}$$

where m is the number of subfields of $K(t)/K(f(t))$ and $r \leq n$ is the number of irreducible factors of ∇_f (see Corollary 4.16). This approach has the following improvements:

- Better complexity: our algorithm does not depend exponentially on r as previous methods (e.g., [3]), only on the number m (usually $m \ll 2^r$). Furthermore, the non-polynomial part of the complexity is reduced to CPU operations.
- Better run times: an implementation in Magma shows the efficiency of our algorithm when compared to [3].
- Better complexity for polynomial decomposition (especially in the wild case): given $f(t) \in \mathbb{F}_q[t]$, we can find all *minimal decompositions* of f with an expected number of $\tilde{O}(rn^2)$ operations in $\mathbb{F}_q$ plus the cost of factoring $\nabla_f = f(x) - f(t) \in \mathbb{F}_q[x, t]$, where r is the number of irreducible factors of ∇_f. See Remark 7.

As previous methods, our algorithm requires the factorization of a bivariate polynomial over K of total degree at most $2n$, where n is the degree of f.

ISSAC '17, July 25-28, 2017, Kaiserslautern, Germany
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5064-8/17/07…$15.00
https://doi.org/http://dx.doi.org/10.1145/3087604.3087608

1.1 Roadmap

In Section 2, we recall some basic definitions and results about rational function decomposition. Let K be a field and let $f \in K(t)$ be a rational function. In Section 3, we give a description of the principal subfields of the extension $K(t)/K(f)$. Every subfield of a finite separable field extension corresponds to a unique partition on the set of irreducible factors of the minimal polynomial of this extension. In Section 4, we show how one can compute this partition for every principal subfield. This allows us to compute the subfield lattice of $K(t)/K(f(t))$ efficiently. Finally, in Section 5, we show how one can use these partitions to compute all decompositions of f. Some timings comparing our algorithm with [3] are also given.

1.2 Complexity model

Throughout this paper, field operations $(+, -, \times, \div)$ and the equality test are assumed to have a constant cost. Given polynomials $f, g \in K[x]$ of degree at most n, we can compute their product (and the remainder of f divided by g) with $O(M(n))$ field operations. We recall that M is *super-additive*: $M(n_1) + M(n_2) \leq M(n_1 + n_2)$ (see [22], Chapter 8.3). If $f \in K[x]$ is irreducible with degree n, then arithmetic in $K[x]/(f)$ costs $O(M(n))$ operations in K (see [22], Chapter 9). Furthermore, the greatest common divisor of two polynomials f, g of degree $\leq n$ costs $O(M(n) \log n)$ field operations (see [22], Chapter 11). Finally, given a linear system $\mathcal{S}$, with m equations in r variables, we can compute a basis of solutions of $\mathcal{S}$ with $O(mr^{\omega-1})$ field operations (see [5], Chapter 2), where $2 < \omega \leq 3$ is a *feasible matrix multiplication exponent* (see [22], Chapter 12) .

2 BASIC DEFINITIONS

Let K be an arbitrary field and let $K(t)$ be the function field over K. Let $\mathbb{S} = K(t) \backslash K$ be the set of non-constant rational functions and let $f = f_n/f_d \in \mathbb{S}$ be a rational function with $f_n, f_d \in K[t]$ coprime. The *degree* of f is defined as $\max\{\deg(f_n), \deg(f_d)\}$ and denoted by $\deg(f)$. The set $\mathbb{S}$ is equipped with a structure of a monoid under composition. The K-automorphisms of $K(t)$ are the fractional transformations $u = (ax+b)/(cx+d)$ such that $ad-bc \neq 0$. The group of automorphisms is isomorphic to $PGL_2(K)$ and also to the group of units of $\mathbb{S}$ under composition.

An element $f \in K(t)$ is *indecomposable* if f is not a unit and $f = g \circ h$ implies g or h is a unit. Otherwise, f is called *decomposable*. If f is decomposable with $f = g \circ h$, then h (resp. g) is called the *right* (resp. *left*) *component* of the decomposition $g \circ h$. Furthermore, a decomposition $f = g \circ h$ is *minimal* if h is indecomposable and a decomposition $f = g_m \circ \cdots \circ g_1$ is *complete* if all g_i are indecomposable.

It is well known by Lüroth's Theorem that if $K \subsetneq L \subseteq K(t)$, then there exists $h \in \mathbb{S}$ such that $L = K(h)$ (a proof can be found in [18]). The rational function h is not unique however, $K(h) = K(h')$, if and only if, there exists a unit $u \in \mathbb{S}$ such that $h' = u \circ h$. As in [3], we define the *normal form* of a rational function $f \in \mathbb{S}$.

Definition 2.1. A rational function $f = p/q \in \mathbb{S}$ is in *normal form* or *normalized* if $p, q \in K[t]$ are monic, coprime, $p(0) = 0$ and either $deg(p) > \deg(q)$ or $m := \deg(p) < \deg(q) =: n$ and $q = t^n + q_{n-1}t^{n-1} + \cdots + q_0$, with $q_m = 0$.

Given $f \in \mathbb{S}$, there exists a unique normalized $\hat{f} \in \mathbb{S}$ such that $K(f) = K(\hat{f})$ ([3], Proposition 2.1). Hence, if $\mathcal{N}_K$ is the set of all normalized rational functions over K, then there exists a bijection between $\mathcal{N}_K$ and the set of fields L such that $K \subsetneq L \subseteq K(t)$. In particular, there is a bijection between normalized rational functions $h \in \mathbb{S}$ such that $f = g \circ h$, for some $g \in \mathbb{S}$, and the fields $L = K(h)$ such that $K(f) \subseteq L \subseteq K(t)$.

Definition 2.2. For a rational function $g = g_n/g_d \in \mathbb{S}$, with $g_n, g_d \in K[t]$ coprime, define $\nabla_g(x,t) := g_n(x)g_d(t) - g_n(t)g_d(x) \in K[x,t]$ and $\Phi_g(x) := g_n(x) - g(t)g_d(x) \in K(g)[x]$. A bivariate polynomial $a(x,t) \in K[x,t]$ is called *near-separate* if $a(x,t) = \nabla_g(x,t)$, for some $g \in K(t)$.

In this work, we assume that f is such that Φ_f is monic. If this is not the case, we can find a unit $u \in K(t)$ such that $\tilde{f} := u \circ f$ and $\Phi_{\tilde{f}}$ is monic. Decomposing f is equivalent to decomposing $\tilde{f}$.

Remark 1. *Let $f \in K(t)$ of degree n and let $G_1, \ldots, G_r$ be the irreducible factors of $\nabla_f \in K[x,t]$. Let $m_1, \ldots, m_r \in K[t]$ be the leading coefficients of $G_1, \ldots, G_r$ w.r.t. x. Then $m_1 \cdots m_r = f_d(t)$ and $F_i := G_i/m_i \in K(t)[x]$ are monic, irreducible and $\nabla_f/f_d(t) = \Phi_f(x) = F_1 \cdots F_r$. In particular, if the exponents of t in G_i are bounded by d_i, then $\sum d_i = n$.*

The following theorem is the key result behind all *near-separate based* rational function decomposition algorithms, such as [2] and [3] (see also [4] for the polynomial case).

Theorem 2.3 ([2], Proposition 3.1). *Let $f, h \in \mathbb{S}$ be rational functions. The following are equivalent:*

a) $K(f) \subseteq K(h) \subseteq K(t)$.
b) $f = g \circ h$, for some $g \in \mathbb{S}$.
c) $\nabla_h(x,t)$ divides $\nabla_f(x,t)$ in $K[x,t]$.
d) $\Phi_h(x)$ divides $\Phi_f(x)$ in $K(t)[x]$.

If $G_1, \ldots, G_r$ are the irreducible factors of ∇_f over $K[x,t]$, then the product of any subset of $\{G_1, \ldots, G_r\}$, which is a near-separate multiple of $x - t$, yields a right component h and hence, a decomposition $f = g \circ h$. Many authors use this approach to compute all decompositions of f: factor ∇_f and search for near-separate factors (see [2–4]). However, this approach leads to exponential time algorithms due to the number of factors we have to consider.

3 PRINCIPAL SUBFIELDS

In this section we use the idea of *principal subfields* to compute the subfield lattice of $K(t)/K(f)$. By Theorem 2.3, this gives us all complete decompositions of f. Principal subfields and fast field intersection techniques (see [17]) allow us to improve the non-polynomial part of the complexity.

3.1 Main Theorem

Let K/k be a separable field extension of finite degree n. A field L is said to be *a subfield of K/k* if $k \subseteq L \subseteq K$. It is well known that the number of subfields of K/k is not polynomially bounded in general. However, we have the following remarkable result from [19]:

Theorem 3.1. *Given a separable field extension K/k of finite degree n, there exists a set $\{L_1, \ldots, L_r\}$, with $r \leq n$, of subfields* *of K/k such that, for any subfield L of K/k, there exists a subset $I_L \subseteq \{1, \ldots, r\}$ with*

$$L = \bigcap_{i \in I_L} L_i.$$

The subfields $L_1, \ldots, L_r$ are called *principal subfields* of the extension K/k and can be obtained as the kernel of some application (see [19]). Instead of directly searching for all subfields of a field extension, which leads to an exponential time complexity, principal subfields allow us to search for a specific set of $r \leq n$ subfields, a polynomial time task.

By Theorem 3.1, the non-polynomial part of the complexity of computing the subfield lattice is then transfered to computing all intersections of the principal subfields. However, according to [17], each subfield of K/k can be uniquely represented by a partition of $\{1, \ldots, r\}$. Computing intersections of principal subfields can now be done by simply joining the corresponding partitions of $\{1, \ldots, r\}$, which in practice can be done very quickly and hence, corresponds to a very small percentage of the total CPU time.

In the remaining of this section we give a description of the principal subfields of $K(t)/K(f(t))$ and in the next section we show how one can compute the partitions associated to every principal subfield of $K(t)/K(f(t))$.

3.2 Principal Subfields of $K(t)/K(f)$

In this section we describe the principal subfields of the field extension $K(t)/K(f)$. We follow [19], making the necessary changes to our specific case.

Remark 2. *If char(K) $= 0$, then Φ_f is separable. If char(K) $= p > 0$ and Φ_f is not separable, then $f = \tilde{f} \circ t^{p^s}$, for some $s \geq 1$ and $\tilde{f} \in K(t)$ with $\Phi_{\tilde{f}}$ separable. For this reason, we assume that Φ_f is separable.*

Definition 3.2. Let $F_1, \ldots, F_r$ be the monic irreducible factors of Φ_f over $K(t)$. For $j = 1, \ldots, r$, define the set

$$L_j := \left\{ g(t) \in K(t) \ : \ F_j \mid \Phi_g \right\}. \tag{1}$$

If we assume that $F_1 = x - t$, then $L_1 = K(t)$. Furthermore

Theorem 3.3. *Let $F_1, \ldots, F_r$ be the irreducible factors of Φ_f over $K(t)$. Then $L_1, \ldots, L_r$ are subfields of $K(t)/K(f)$.*

Proof. We show that L_j is closed under multiplication and taking inverse. The remaining properties can be shown in the same fashion. Let $g(t) = g_n(t)/g_d(t)$ and $h(t) = h_n(t)/h_d(t)$ be elements of L_j. By definition,

$$F_j \mid \Phi_g \text{ and } F_j \mid \Phi_h. \tag{2}$$

Now $g(t)h(t) \in L_j$ if and only if, $F_j \mid \Phi_{gh}$. By a simple manipulation, one can show that

$$\Phi_{gh} = g_n(x)\Phi_h + h(t)h_d(x)\Phi_g. \tag{3}$$

Therefore, by Equation (2), it follows that $F_j \mid \Phi_{gh}$ and hence, $g(t)h(t) \in L_j$. To show that the inverse of $g(t)$ is in L_j, notice that

$$F_j \mid \Phi_g \text{ if and only if } F_j \mid \Phi_{1/g}, \tag{4}$$

since $\Phi_g = -g(t)\Phi_{1/g}$ in $K(t)[x]$. Therefore, $1/g(t) \in L_j$. □

Finally, we show that the subfields $L_1, \ldots, L_r$ are the principal subfields of $K(t)/K(f)$.

Theorem 3.4. *The subfields $L_1, \ldots, L_r$ of $K(t)/K(f(t))$, where L_j is defined as in (1), for $j = 1, \ldots, r$, are the principal subfields of the extension $K(t)/K(f(t))$.*

Proof. Given a subfield L of $K(t)/K(f(t))$, by Lüroth's Theorem, there exists a rational function $h(t) \in K(t)$ such that $L = K(h(t))$ and therefore, $f = g \circ h$, for some $g \in K(t)$. By Theorem 2.3 it follows that $\Phi_h \mid \Phi_f$. Therefore, there exists a set $I_L \subseteq \{1, \ldots, r\}$ such that $\Phi_h = \prod_{i \in I_L} F_i$. We shall prove that

$$L = \{g(t) \in K(t) \ : \ \Phi_h \mid \Phi_g\} = \bigcap_{i \in I_L} L_i. \tag{5}$$

Let $g(t) \in K(t)$. Then $g(t) \in L = K(h)$ if and only if $g(t) = \tilde{g} \circ h(t)$, for some $\tilde{g}(t) \in K(t)$, if and only if $\Phi_h \mid \Phi_g$, by Theorem 2.3. For the second equality, suppose that $g(t) \in \cap_{i \in I_L} L_i$. Then $F_i \mid \Phi_g$, for every $i \in I_L$. Since we are assuming Φ_f to be separable (see Remark 2), it follows that $\Phi_h = \prod_{i \in I_L} F_i \mid \Phi_g$. Conversely, if $\Phi_h \mid \Phi_g$, then $F_i \mid \Phi_g$, for every $i \in I_L$, that is, $g(t) \in L_i$, for every $i \in I_L$ and hence, $g(t) \in \cap_{i \in I_L} L_i$. □

4 PARTITION OF PRINCIPAL SUBFIELDS

Let $K(t)/K(f)$ be a separable field extension of finite degree n and let $\Phi_f(x)$ be the minimal polynomial of t over $K(f)$. Let $F_1, \ldots, F_r$ be the irreducible factors of Φ_f over $K(t)$ and let $L_1, \ldots, L_r$ be the corresponding principal subfields of $K(t)/K(f)$.

Definition 4.1. A *partition* of $S = \{1, \ldots, r\}$ is a set $\{P^{(1)}, \ldots, P^{(s)}\}$ such that $P^{(i)} \subseteq S$, $P^{(i)} \cap P^{(j)} = \emptyset$, for every $i \neq j$ and $\cup P^{(i)} = S$.

Definition 4.2. Let P and Q be partitions of $\{1, \ldots, r\}$. We say that P *refines* Q if every part of P is contained in some part of Q.

Recall that $F_1 = x - t$. We number the parts of a partition $P = \{P^{(1)}, \ldots, P^{(s)}\}$ in such a way that $1 \in P^{(1)}$. Let P be a partition of $\{1, \ldots, r\}$. We say that P is the *finest partition* satisfying some property X if P satisfies X and if Q also satisfies X then P refines Q. Moreover, the *join* of two partition P and Q is denoted by $P \vee Q$ and is the finest partition that is refined by both P and Q.

Definition 4.3. Let $F_1, \ldots, F_r$ be the irreducible factors of Φ_f over $K(t)$. Given a partition $P = \{P^{(1)}, \ldots, P^{(s)}\}$ of $\{1, \ldots, r\}$, define the polynomials (so called *P-products*)

$$g_i := \prod_{j \in P^{(i)}} F_j \in K(t)[x], \ i = 1, \ldots, s.$$

Theorem 4.4 ([17], Section 2). *Let $f \in K(t)$ and let $F_1, \ldots, F_r$ be the irreducible factors of Φ_f over $K(t)$. Given a subfield L of $K(t)/K(f)$, there exists a unique partition $P_L = \{P^{(1)}, \ldots, P^{(s)}\}$ of $\{1, \ldots, r\}$, called the partition of L, such that s is maximal with the property that the P_L-products are polynomials in $L[x]$. Furthermore, $P_{L \cap L'} = P_L \vee P_{L'}$, that is, the partition of $L \cap L'$ is the join of the partitions P_L and $P_{L'}$ of L and L', respectively.*

Since $F_1, \ldots, F_r$ are the irreducible factors of Φ_f over $K(t)$, P_L represents the factorization of Φ_f over L. Algorithms for computing the join of two partitions can be found in [10, 17] (see also [11]).

Since $1 \in P_L^{(1)}$, the first P_L-product is the minimal polynomial of t over L. As in [17], we give two algorithms for computing the partition of the principal subfield L_i: one deterministic and one probabilistic, with better performance.

4.1 A Deterministic Algorithm

In this section we present a deterministic algorithm that computes, by solving a linear system, the partitions $P_1, \dots, P_r$ of the principal subfields $L_1, \dots, L_r$. We recall (see [17], Section 3) that to find the partition of L_i it is enough to find a basis of the vectors $(e_1, \dots, e_r) \in \{0,1\}^r$ such that $\prod_{j=1}^r F_j^{e_j} \in L_i[x]$.

Theorem 4.5 ([17], Lemmas 31 and 32). *Let $c_1, \dots, c_{2n} \in K(f)$ be distinct elements and let $h_{j,k}(t) := F_j'(c_k)/F_j(c_k) \in K(t)$. If $(e_1, \dots, e_r) \in \{0,1\}^r$ is such that $\sum_{j=1}^r e_j h_{j,k}(t) \in L_i$, for $k = 1, \dots, 2n$, then $\prod_{j=1}^r F_j^{e_j} \in L_i[x]$.*

Let us consider $e_1, \dots, e_r$ as variables. To show that $\sum e_j h_{j,k}(t) \in L_i$ we need an expression of the form $a(t)/b(t)$, where $a, b \in K[t]$. Assume $h_{j,k}(t) = n_{j,k}(t)/d_{j,k}(t)$, where $n_{j,k}(t), d_{j,k}(t) \in K[t]$ are coprime. Hence

$$\sum_{j=1}^r e_j \frac{F_j'(c_k)}{F_j(c_k)} = \sum_{j=1}^r e_j h_{j,k}(t) = \sum_{j=1}^r e_j \frac{n_{j,k}(t)}{d_{j,k}(t)}.$$

Furthermore, let $l_k(t) \in K[t]$ be the least common multiple of $d_{1,k}(t), \dots, d_{r,k}(t) \in K[t]$. Hence

$$\sum_{j=1}^r e_j h_{j,k}(t) = \sum_{j=1}^r e_j \frac{n_{j,k}(t)}{d_{j,k}(t)} = \frac{\sum_{j=1}^r e_j p_{j,k}(t)}{l_k(t)}, \tag{6}$$

where $p_{j,k}(t) := l_k(t)\frac{n_{j,k}(t)}{d_{j,k}(t)} \in K[t]$. Hence, $\sum_{j=1}^r e_j h_{j,k}(t) \in L_i$ if, and only if (see Definition 3.2)

$$\left[\sum_{j=1}^r e_j p_{j,k}(x) - \frac{\sum_{j=1}^r e_j p_{j,k}(t)}{l_k(t)} l_k(x)\right] \bmod F_i = 0, \tag{7}$$

where $a \bmod b$ is the remainder of division of a by b. By manipulating Equation (7) we have

$$\sum_{j=1}^r e_j \left[\left(p_{j,k}(x) - h_{j,k}(t) l_k(x)\right) \bmod F_i\right] = 0. \tag{8}$$

Hence, if $(e_1, \dots, e_r) \in \{0,1\}^r$ is a solution of (8), for $k = 1, \dots, 2n$, then Theorem 4.5 tells us that $\prod_{j=1}^r F_j^{e_j} \in L_i[x]$.

We will now explicitly present the system given by Equation (8). Let

$$q_{j,k}(x) := p_{j,k}(x) - h_{j,k}(t) l_k(x) \in K(t)[x].$$

Notice that $\deg_x(q_{j,k}) \le dn$, where $d = \deg_t(c_k)$. Furthermore, let

$$r_{i,j,k}(x) := q_{j,k}(x) \bmod F_i \in K(t)[x]. \tag{9}$$

Let $m_j(t) \in K[t]$ be the monic lowest degree polynomial such that $m_j(t) r_{i,j,k} \in K[t][x]$ and let $l \in K[t]$ be the least common multiple of $m_1(t), \dots, m_r(t)$. Hence

$$l \sum_{j=1}^r e_j r_{i,j,k} = \sum_{j=1}^r e_j \hat{r}_{i,j,k} \in K[t][x],$$

where $\hat{r}_{i,j,k} = l \cdot r_{i,j,k} \in K[t][x]$. Notice that Equation (8) holds if and only if, $\sum_{j=1}^r e_j \hat{r}_{i,j,k} = 0$. Next, let us write

$$\hat{r}_{i,j,k} = \sum_{d=0}^{d_i - 1} \sum_{s=0}^{S} c_j(s,d,k) t^s x^d, \quad \text{where } c_j(s,d,k) \in K,$$

where d_i is the degree of F_i and $S \ge 0$ is a bound for the t-exponents. Therefore,

$$\sum_{j=1}^r e_j \hat{r}_{i,j,k} = \sum_{d=0}^{d_i - 1} \sum_{s=0}^{S} \left(\sum_{j=1}^r e_j c_j(s,d,k) \right) t^s x^d$$

and hence, the system in $e_1, \dots, e_r$ from Equation (8) is given by

$$\mathcal{S}_i := \left\{ \sum_{j=1}^r e_j c_j(s,d,k) = 0, \quad \begin{array}{l} d = 0, \dots, d_i - 1, \\ s = 0, \dots, S, \\ k = 1, \dots, 2n. \end{array} \right. \tag{10}$$

Definition 4.6. A basis of solutions $s_1, \dots, s_d$ of a linear system with r variables is called a $\{0,1\}$*-echelon basis* if

(1) $s_i = (s_{i,1}, \dots, s_{i,r}) \in \{0,1\}^r$, $1 \le i \le d$, and
(2) For each $j = 1, \dots, r$, there is a unique i, $1 \le i \le d$ such that $s_{i,j} = 1$.

For instance, $S = \{(1,1,0,0), (0,0,1,0), (0,0,0,1)\}$ is a basis of solutions in $\{0,1\}$-echelon form. If a linear system admits a $\{0,1\}$-echelon basis then this basis coincides with the (unique) reduced echelon basis of this system.

Definition 4.7. Let $\mathcal{S}$ be a linear system with $\{0,1\}$-echelon basis $\{s_1, \dots, s_d\}$. The *partition defined by* this basis is the partition $P = \{P^{(1)}, \dots, P^{(d)}\}$ where $P^{(j)} = \{i \ : \ s_{j,i} = 1\}$, for $j = 1, \dots, d$.

For instance, $P_S = \{\{1,2\}, \{3\}, \{4\}\}$ is the partition defined by S given above. Therefore, by computing the $\{0,1\}$-echelon basis of the system $\mathcal{S}_i$ given in (10) (notice that $\mathcal{S}_i$ admits such basis), the partition defined by this basis is the partition of L_i. This is summarized in the next algorithm.

Algorithm 1 `Partition-D` (Deterministic)

```
Input: The irreducible factors $F_1, \dots, F_r$ of $\Phi_f(x)$ over $K(t)$ and an index $1 \le i \le r$.
Output: The partition $P_i$ of $L_i$.
1. Compute the system $\mathcal{S}_i$ as in (10).
2. Compute the $\{0,1\}$-echelon basis of $\mathcal{S}_i$.
3. Let $P_i$ be the partition defined by this basis.
4. return $P_i$.
```

However, algorithm `Partition-D` is not efficient in practice due to the (costly) $2nr$ polynomial divisions in $K(t)[x]$. We shall present a probabilistic version of this algorithm in Section 4.3, which allows us to compute P_i much faster.

4.2 Valuation rings of $K(t)/K$

In this section we briefly recall the definition and some properties of valuation rings of a rational function field. We will use valuation rings to simplify and speed up the computation of the partition P_i of L_i. The results presented in this subsection can be found in [16].

Definition 4.8. A *valuation ring* of $K(t)/K$ is a ring $\mathcal{O} \subseteq K(t)$ with the following properties:

(1) $K \subsetneq \mathcal{O} \subsetneq K(t)$, and
(2) for every $g \in K(t)$ we have $g \in \mathcal{O}$ or $1/g \in \mathcal{O}$.

Valuation rings are *local rings*, that is, if $\mathcal{O}$ is a valuation ring, then there exists a unique maximal ideal $\mathcal{P} \subseteq \mathcal{O}$.

Lemma 4.9. *Let $p \in K[x]$ be an irreducible polynomial. Let*

$$O_p := \left\{ \frac{g_n(t)}{g_d(t)} \in K(t) \ : \ p(x) \nmid g_d(x) \right\} \text{ and}$$

$$\mathcal{P}_p := \left\{ \frac{g_n(t)}{g_n(t)} \in K(t) \ : \ p(x) \nmid g_d(x) \text{ and } p(x) \mid g_n(x) \right\}.$$

Then O_p is a valuation ring with maximal ideal $\mathcal{P}_p$.

Furthermore, every valuation ring O of $K(t)/K$ is of the form O_p, for some irreducible polynomial $p(x) \in K[x]$, or is the place at infinity of $K(t)/K$, that is, $O = \{\frac{g_n(t)}{g_d(t)} \in K(t) \ : \ \deg(g_n(x)) \le \deg(g_d(x))\}$.

Lemma 4.10. *Let O_p be a valuation ring of $K(t)/K$, where $p \in K[x]$ is an irreducible polynomial, and let $\mathcal{P}_p$ be its maximal ideal. Let $\mathbf{F}_p$ be the residue class field $O_p/\mathcal{P}_p$. Then $\mathbf{F}_p \cong K[x]/\langle p(x) \rangle$.*

4.3 A Las Vegas Type Algorithm

In this section we present a probabilistic version of Algorithm `Partition-D`. We begin by noticing, as in [17], that fewer points are enough to find the partition P_i (usually much less than $2n$). Furthermore, the equations of the system $\mathcal{S}_i$ come from the computation of $r_{i,j,k} \in K(t)[x]$ in (9), which involves a polynomial division over $K(t)$. Let us define a *good ideal* $\mathcal{P}_p$:

Definition 4.11. Let $f \in K(t)$ and let $F_1, \dots, F_r$ be the monic irreducible factors of Φ_f over $K(t)$. Let $O_p \subset K(t)$ be a valuation ring with maximal ideal $\mathcal{P}_p$, where $p = p(x) \in K[x]$ is irreducible. Let $\mathbf{F}_p$ be its residue field. We say that $\mathcal{P}_p$ is a *good $K(t)$-ideal* (with respect to f) if

1) $F_i \in O_p[x]$, $i = 1, \dots, r$.
2) The image of f in $\mathbf{F}_p$ is not zero.
3) The image of $\Phi_f(x)$ in $\mathbf{F}_p[x]$ is separable.

To avoid the expensive computations of $r_{i,j,k} \in K(t)[x]$, we only compute their image modulo a good $K(t)$-ideal $\mathcal{P}_p$ (i.e., by mapping $t \longrightarrow \alpha$, where α is a root of $p(x)$). These reductions will simplify our computations and we will still be able to construct a system $\tilde{S}_i$ which is likely to give us the partition P_i.

Remark 3. *Condition* 1) *in Definition 4.11 is equivalent to $p(x) \nmid f_d(x)$ (recall Remark 1) and condition* 2) *is equivalent to $p(x) \nmid f_n(x)$. The image of Φ_f in $\mathbf{F}_p[x]$ is separable if $p(t)$ does not divide $R := resultant(\nabla_f, \nabla'_f, x) \in K[t]$. The degree of R is bounded by $(2n-1)n$. Instead of mapping $t \longrightarrow \alpha$, we could map t to any element in $\mathbf{F}_p = K[x]/\langle p(x) \rangle$. Hence, if size$(K)^{d_p} > (2n-1)n$, where $d_p = \deg(p(x))$, then we are guaranteed to find a good evaluation point in $\mathbf{F}_p$ which satisfies the conditions in Definition 4.11. Hence, $d_p \in O(\log n)$. For best performance, we look for $p(x)$ of smallest degree possible and use the mapping $t \longrightarrow \alpha$. Notice that if char$(K) = 0$, we can always choose $p(x)$ linear.*

4.3.1 Simplified System.

4.3.1 Simplified System. Let $\mathcal{P}_p$ be a good $K(t)$-ideal, where $p = p(x) \in K[x]$ is irreducible. Let O_p be its valuation ring and $\mathbf{F}_p$ be its residue class field. Let $c \in K(f)$ be such that

$$h_{j,c}(t) := F'_j(c)/F_j(c) \in O_p \subseteq K(t),$$

for $j = 1, \dots, r$, and let $p_{j,c}(t), l_c(t) \in K[t]$ be as in Equation (6). Let $\tilde{F}_i$ be the image of F_i in $\mathbf{F}_p[x]$ and let $\tilde{h}_{j,c}$ be the image of $h_{j,c}$ in $\mathbf{F}_p$. Let

$$\tilde{q}_{j,c}(x) := p_{j,c}(x) - \tilde{h}_{j,c}\ l_c(x) \in \mathbf{F}_p[x] \tag{11}$$

and let $\tilde{r}_{i,j,c} := \tilde{q}_{j,c}(x) \bmod \tilde{F}_i \in \mathbf{F}_p[x]$. Let d_p be the degree of $p(x) \in K[x]$ and let α be one of its roots. By Lemma 4.10 we have $\mathbf{F}_p \cong K[\alpha]$ and hence

$$\tilde{r}_{i,j,c} = \sum_{d=0}^{d_i-1} \sum_{s=0}^{d_p-1} C_j(s,d)\alpha^s x^d, \text{ where } C_j(s,d) \in K. \tag{12}$$

Consider the system $\tilde{\mathcal{S}}_{i,c}$ given by

$$\tilde{\mathcal{S}}_{i,c} := \left\{ \sum_{j=1}^{r} e_j C_j(s,d) = 0, \qquad \begin{array}{l} d = 0, \dots, d_i - 1, \\ s = 0, \dots, d_p - 1. \end{array} \right. \tag{13}$$

where $C_j(s,d) \in K$ is as in Equation 12. If $(e_1, \dots, e_r) \in \{0,1\}^r$ is a solution of $\mathcal{S}_i$, then $(e_1, \dots, e_r)$ must also satisfy the system $\tilde{\mathcal{S}}_{i,c}$. The converse, however, need not be true. A basis of solutions of $\tilde{\mathcal{S}}_{i,c}$ is not necessarily a basis of solutions of $\mathcal{S}_i$. In fact, a basis of solutions of $\tilde{\mathcal{S}}_{i,c}$ might not even be a $\{0,1\}$-echelon basis. If this happens we need to consider more equations by taking $c' \in K(f)$ such that $h_{j,c'}(t) \in O_p$, for $j = 1, \dots, r$, and solving $\tilde{S}_i := \tilde{\mathcal{S}}_{i,c} \cup \tilde{\mathcal{S}}_{i,c'}$, and so on. In subsection 4.3.2 we give a halting condition that tells us when to stop adding more equations to the system $\tilde{S}_i$.

Remark 4. *Advantages of considering $\tilde{\mathcal{S}}_i$ over $\mathcal{S}_i$:*

(1) *Smaller number of polynomial divisions to define $\tilde{\mathcal{S}}_i$.*
(2) *The polynomial divisions are over $K[x]/\langle p(x) \rangle$, where $p(x) \in K[x]$ is the polynomial defining the ideal $\mathcal{P}$.*
(3) *Smaller system: $\tilde{\mathcal{S}}_i$ has at most dd_id_p equations, where d is the number of c's used to construct $\tilde{S}_i$, while $\mathcal{S}_i$ has at most $2nd_iS$ equations in $r \le n$ variables.*

Although in practice we need very few elements $c \in K(f)$ to find P_i (see Table 1), we were not able to show that $2n$ elements are sufficient to compute P_i.

4.3.2 Halting Condition.

4.3.2 Halting Condition. Let $\tilde{S}_i = \cup \tilde{S}_{i,c}$ be a system constructed from several $c \in K(f)$, where $\tilde{S}_{i,c}$ is as in (13). We will give a halting condition that tells us when to stop adding more equations. If $\tilde{S}_i$ does not have a $\{0,1\}$-echelon basis then we clearly need more equations. Now let us suppose that $\tilde{S}_i$ has a $\{0,1\}$-echelon basis. Then the partition $\tilde{P}_i$ corresponding to this basis (see Definition 4.7) might still be a proper refinement of P_i (the correct partition). To show that $\tilde{P}_i = P_i$ it suffices to show that the $\tilde{P}_i$-products are polynomials in $L_i[x]$. To do so, we use the following lemma.

Lemma 4.12 ([17], Lemma 37). *Let K be a field and $F \in K[x]$ monic and separable. Let $O \subseteq K$ be a ring such that $F = g_1 \cdots g_s = h_1 \cdots h_s$, where $g_j, h_j \in O[x]$ are monic (not necessarily irreducible). Let $\mathcal{P} \subseteq O$ be a maximal ideal such that the image of F over the residue class field is separable. If $g_j \equiv h_j \bmod \mathcal{P}$, $1 \le j \le s$, then $g_j = h_j$, $1 \le j \le s$.*

In order to apply this lemma, consider the following map

$$\begin{array}{rcl} \Psi_i : K(t) & \longrightarrow & K(t,x) \\ g(t) & \longmapsto & \frac{g_n(x) \bmod F_i}{g_d(x) \bmod F_i}. \end{array}$$

Hence, $g(t) \in L_i$ if, and only if, $\Psi_i(g) = g$ (see Definition 3.2) and therefore, we can rewrite $L_i = \{g(t) \in K(t) \ :\ \Psi_i(g(t)) = g(t)\}$.

Theorem 4.13. *Let P_i be the partition of L_i and let $\tilde{P}_i$ be a refinement of P_i. Let $\mathcal{P}_p$ be a good $K(t)$-ideal. If $\tilde{g}_1, \ldots, \tilde{g}_s \in K(t)[x]$ are the $\tilde{P}_i$-products and if $\Psi_i(\tilde{g}_j) \equiv \tilde{g}_j \bmod \mathcal{P}_p$, $j = 1, \ldots, s$, where Ψ_i acts on $\tilde{g}_j$ coefficient-wise, then $\tilde{P}_i = P_i$.*

Proof. Since $\tilde{P}_i$ is a refinement of P_i, it suffices to show that the $\tilde{P}_i$-products $\tilde{g}_1 \ldots, \tilde{g}_s$ are polynomials in $L_i[x]$. That is, we have to show that $\Psi_i(\tilde{g}_j) = \tilde{g}_j$, for $j = 1, \ldots, s$. Since

$$\tilde{g}_1 \cdots \tilde{g}_s = \Phi_f(x) = \Psi_i(\Phi_f(x)) = \Psi_i(\tilde{g}_1) \cdots \Psi_i(\tilde{g}_s)$$

and $\Psi_i(\tilde{g}_j) = \tilde{g}_j \bmod \mathcal{P}_p$, for $1 \leq j \leq s$, then Lemma 4.12 implies that $\Psi_i(\tilde{g}_j) = \tilde{g}_j$. Thus $\tilde{g}_j \in L_i[x]$, for $j = 1, \ldots, s$, and $\tilde{P}_i = P_i$. □

This gives us a procedure to determine if the solutions of a system give the partition P_i of the principal subfield L_i.

```
Algorithm 2 Check
  Input:  A linear system $\mathcal{S}$ in $e_1, \ldots, e_r$ and an index $i$.
  Output: The partition $P_i$ of $L_i$ or false.
  1. Compute a basis of solutions of $\mathcal{S}$.
  2. if this basis is not a $\{0,1\}$-echelon basis then
  3.     return false  *Need more equations.
  4. Let $\tilde{P}_i$ be the partition defined by this basis.
  5. Let $\tilde{F}_i$ be the image of $F_i$ in $\mathbf{F}_p[x]$.
  6. Let $\tilde{g}_1, \ldots, \tilde{g}_d$ be the $\tilde{P}_i$-products.
  7. for every coefficient $c = \frac{c_n(t)}{c_d(t)} \in K(t)$ of $\tilde{g}_1, \ldots, \tilde{g}_d$ do
  8.     Let $\tilde{c}$ be the image of $c$ in $\mathbf{F}_p$.
  9.     if $c_n(x) \bmod \tilde{F}_i \neq \tilde{c} \cdot (c_d(x) \bmod \tilde{F}_i)$ then
  10.        return false  *Need more equations.
  11. return $\tilde{P}_i$
```

The correctness of the algorithm follows from Theorem 4.13. We end this section by computing the complexity of Algorithm Check.

Theorem 4.14. *One call of Algorithm* Check *can be performed with $O(n_e r^{\omega-1} + M(n^2) + nM(n)M(d_p))$ field operations, where d_p is the degree of the polynomial defining $\mathcal{P}_p$, n_e is the number of equations in $\mathcal{S}$ and ω is a feasible matrix multiplication exponent.*

Proof. A basis of solutions of $\mathcal{S}$ is computed with $O(n_e r^{\omega-1})$ field operations. If this basis is not a $\{0,1\}$-echelon basis, then the algorithm returns *false*. The computation of the polynomials $\tilde{g}_1, \ldots, \tilde{g}_d$ in Step 6 can be done with $r - d$ bivariate polynomial multiplications. By Remark 1, $\sum \deg_t G_i = \sum \deg_x G_i = n$ and hence, we can compute $\tilde{g}_1, \ldots, \tilde{g}_d$ with $O(M(n^2))$ field operations (recall that $M(\cdot)$ is super-additive). For each coefficient of $\tilde{g}_1, \ldots, \tilde{g}_d$, we have to verify the condition in Step 9, which can be performed with a reduction modulo $\mathcal{P}_p$ (to compute $\tilde{c}$) and two polynomial divisions over $\mathbf{F}_p$. Therefore, for each c, we can perform Steps 8 and 9 with $O(M(n)M(d_p))$ field operations. Since $\sum \deg \tilde{g}_i = n$, we have a total cost of $O(nM(n)M(d_p))$ field operations for Steps 7-10. □

4.3.3 Algorithm Partitions. The following is a Las Vegas type algorithm that computes the partitions $P_1, \ldots, P_r$ of $L_1, \ldots, L_r$.

```
Algorithm 3 Partitions
  Input:  The irreducible factors $F_1, \ldots, F_r$ of $\Phi_f$ and a good $K(t)$-
          ideal $\mathcal{P}_p$ (see Definition 4.11)
  Output: The partitions $P_1, \ldots, P_r$ of $L_1, \ldots, L_r$.
  1. Let $\tilde{\mathcal{S}}_i = \{\ \}$, $i = 1, \ldots, r$.
  2. $I := \{1, \ldots, r\}$.
  3. while $I \neq \emptyset$ do
  4.     Let $c \in K(f)$ such that $h_{j,c}(t) \in O_p$, $j = 1, \ldots, r$.
  5.     Compute $\tilde{q}_{j,c}(x) \in \mathbf{F}_p[x]$ as in Equation 11.
  6.     for $i \in I$ do
  7.         Compute the system $\tilde{\mathcal{S}}_{i,c}$ (see Equation (13)).
  8.         Let $\tilde{\mathcal{S}}_i := \tilde{\mathcal{S}}_i \cup \tilde{\mathcal{S}}_{i,c}$.
  9.         if Check($\tilde{\mathcal{S}}_i, i$) $\neq$ false then
  10.            Remove($I, i$).
  11.            Let $P_i$ be the output of Check($\tilde{\mathcal{S}}_i, i$).
  12. return $P_1, \ldots, P_r$.
```

Remark 5. *In general, the elements in Step 4 can be taken inside K. This will work except, possibly, when K has very few elements, which might not be enough to find P_i. If this happens we have two choices:*

1) *Choose $c \in K(f) \backslash K$ or*
2) *Extend the base field K and compute/solve the system $\tilde{\mathcal{S}}_i$ over this extension.*

We choose the latter. Recall that the solutions we are looking for are composed of 0's and 1's and hence can be computed over any extension of K. Furthermore, extending the base field K does not create new solutions since the partitions are determined by the factorization of $\Phi_f(x)$ computed over $K(t)$, where K is the original field.

In what follows we determine the complexity of computing $P_1, \ldots, P_r$. We assume, based on our experiments (see Table 1), that the algorithm finishes using $O(1)$ elements $c \in K$ (or in a finite extension of K) to generate a system $\tilde{\mathcal{S}}_i$ whose solution gives P_i.

Theorem 4.15. *Assuming that* Algorithm Partitions *finishes using $O(1)$ elements inside K in Step 4, the partitions $P_1, \ldots, P_r$, corresponding to the principal subfields $L_1, \ldots, L_r$ of the field extension $K(t)/K(f(t))$, can be computed with an expected number of $O(r(rM(n)M(d_p) + M(n^2)))$ field operations, where d_p is the degree of the polynomial defining $\mathcal{P}_p$.*

Proof. Given $g = \frac{g_n(t)}{g_d(t)} \in O_p$, we can compute its image in $\mathbf{F}_p$ with $O(M(deg(g)) + M(d_p))$ field operations. Hence, we can compute the images of the polynomials $F_1, \ldots, F_r$ in $\mathbf{F}_p$ with $O(n(M(n) + M(d_p)))$ field operations.

Let $c \in K$. We first compute $h_{j,c} := F_j'(c)/F_j(c) = G_j'(c)/G_j(c) \in O_p$, $j = 1, \ldots, r$ (see Remark 1). Evaluating $G_j \in K[x,t]$ at $x = c$ costs $O(nd_j^x)$, where $d_j^x = \deg_x(G_j)$. If $d_j^t = \deg_t(G_j)$, simplifying the rational function $G_j'(c)/G_j(c)$ to its minimal form costs $O(M(d_j^t) \log d_j^t)$. Keeping in mind that $\sum d_j^t = \sum d_j^x = n$, one can compute $h_{j,c}$, $j = 1, \ldots, r$, with $O(n^2 + M(n) \log n)$ field operations.

Since $c \in K$, $\deg_t(h_{j,c}) \leq d_j^t$ and we can compute the image $\tilde{h}_{j,c}$ of $h_{j,c}, j = 1, \dots, r$, in $\mathbf{F}_p$ with $O(M(n) + rM(d_p))$ field operations. Let us write $h_{j,c} = n_{j,c}/d_{j,c}$, where $n_{j,c}, d_{j,c} \in K[t]$ are coprime. We can compute $l_c = \mathrm{lcm}(d_{1,c}, \dots, d_{r,c})$ with r lcm computations, with a total cost of $O(rM(n)\log n)$ field operations. Next, we define $\tilde{q}_{j,c} = p_{j,c}(x) - \tilde{h}_{j,c}(t)l_c(x)$, $j = 1, \dots, r$, where $p_{j,c}(x) := l_c(x)\frac{n_{j,c}(x)}{d_{j,c}(x)} \in K[x]$. The cost of this step is negligible.

For each $i = 1, \dots, r$, to compute the partition P_i we have to compute the system $\tilde{S}_{i,c}$, which involves the division of $\tilde{q}_{j,c}$ by $\tilde{F}_i$, for $j = 1, \dots, r$. Since $\deg(\tilde{q}_{j,c}(x)) \leq n$, each of these divisions cost $O(M(n)M(d_p))$ field operations and hence, we can compute the system $\tilde{S}_{i,c}$ with $O(rM(n)M(d_p))$ field operations. This system has at most $d_i d_p$ equations and hence, one call of algorithm Check costs $O(d_i d_p r^{\omega-1} + M(n^2) + M(n)M(d_p))$. The result follows by adding the complexities and simplifying. □

Remark 6. *If Algorithm* Partitions *needs s elements $c \in K$ to compute all partitions $P_1, \dots, P_r$, then the total cost is bounded by s times the cost given in Theorem 4.15.*

Corollary 4.16. *Let $f \in K(t)$ of degree n and let $F_1, \dots, F_r$ be the irreducible factors of $\Phi_f(x) \in K(t)[x]$. Let m be the number of subfields of $K(t)/K(f(t))$. One can compute, using fast arithmetic, the subfield lattice of $K(t)/K(f(t))$ with $\tilde{O}(rn^2)$ field operations plus $\tilde{O}(mr^2)$ CPU operations.*

Proof. Using fast arithmetic, we can compute the partitions of the principal subfields with $\tilde{O}(rn^2 d_p)$ field operations, by Theorem 4.15. By Remark 3, $d_p \in O(\log n)$. The complete subfield lattice can be computed with $\tilde{O}(mr^2)$ CPU operations (see [17]). □

5 GENERAL ALGORITHM AND TIMINGS

In this section we outline an algorithm for computing all complete decompositions of f and give an example. Some timings, comparing our algorithm with [3], are also given.

5.1 General Algorithm

Let $f \in K(t)$ and let $F_1, \dots, F_r$ be the monic irreducible factors of Φ_f. By Theorem 2.3, each complete decomposition corresponds to a maximal chain of subfields of $K(t)/K(f(t))$ and vice-versa. Using the algorithms in Section 4 and fast subfield intersection techniques from [17], we can (quickly) compute the subfield lattice of $K(t)/K(f(t))$, where each subfield is represented by a partition. To actually compute the decompositions of f, we need to find a *Lüroth generator* for each subfield. That is, given a partition P_L of $\{1, \dots, r\}$ representing a subfield L, we want to find a rational function $h \in K(t)$ such that $L = K(h)$.

Theorem 5.1. *Let $f \in K(t)$ and let $F_1, \dots, F_r$ be the monic irreducible factors of $\Phi_f \in K(t)[x]$. Let L be a subfield of $K(t)/K(f)$ and $P = \{P^{(1)}, \dots, P^{(s)}\}$ be the partition of L. Let $g := \prod_{i \in P^{(1)}} F_i \in L[x]$. If $c \in K(t)$ is any coefficient of g not in K, then $L = K(c)$.*

Proof. By Luröth's Theorem, there exists a rational function $h(t) \in K(t)$ such that $L = K(h(t))$. Let $\Phi_h \in L[x]$. We may suppose that $\Phi_h \in L[x]$ is the minimal polynomial of t over L. Let $g = \prod_{i \in P^{(1)}} F_i \in L[x]$. Since $1 \in P^{(1)}$ (recall that $F_1 = x - t$), it follows that $g(t) = 0$ and hence, $\Phi_h \mid g$. However, Φ_h and g are monic irreducible polynomials (over L) and hence, $g = \Phi_h$. Therefore, $g = h_n(x) - h(t)h_d(x)$. Let c_i be the coefficient of x^i of g, then

$$c_i = h_{ni} - h(t)h_{d,i} = (-h_{d,i}t + h_{n,i}) \circ h(t),$$

where $h_{n,i}$ and $h_{d,i}$ are the coefficients of x^i in $h_n(x)$ and $h_d(x)$, respectively. If $h_{d,i} \neq 0$, then $-h_{d,i}t + h_{n,i}$ is a unit and hence, $L = K(h(t)) = K(c_i)$. □

Finally, given $f, h \in K(t)$, we want to find $g \in K(t)$ such that $f = g \circ h$. It is known that g is unique (see [2]) and several methods exist for finding g. The most straightforward method is to solve a linear system in the coefficients of g (see [9] for details). Another approach can be found in [12] and uses $O(nM(n)\log n)$ field operations.

Remark 7. *Our algorithm also works when $f \in K[t]$ is a polynomial if we normalize the generator of each subfield. This follows from Corollary 2.3 of [3]. If $f = g \circ h$ is a minimal decomposition, then $K(h)$ is a principal subfield and its partition is not refined by any other except P_1. Thus, given $P_1, \dots, P_r$, it is very easy to verify which of these partitions represents a minimal decomposition. For a principal subfield, a Lüroth generator can be obtained as a byproduct of Algorithm* Check*. Hence, given $P_1, \dots, P_r$, to compute all minimal decompositions of f we only need to compute at most $r - 1$ left components. When char(K) > 0, the factorization of $f(x) - f(t)$ can be computed with $\tilde{O}(n^{\omega+1})$ field operations, where $2 < \omega \leq 3$ is a matrix multiplication exponent (see [8] and [14]). An algorithm in [7] also computes all minimal decompositions, and take $\tilde{O}(n^6)$ field operations (for finite fields). For more details, see [6, Theorem 3.23].*

5.2 An Example

Let $f := (t^{24} - 2t^{12} + 1)/(t^{16} + 2t^{12} + t^8)$ and consider the extension $\mathbb{Q}(t)/\mathbb{Q}(f)$. The irreducible factors of $\Phi_f(x)$ are $F_1 = x - t, F_2 = x + t, F_3 = x + 1/t, F_4 = x - 1/t, F_5 = x^2 + t^2, F_6 = x^2 + 1/t^2, F_7 = x^8 + (\alpha/t^4\beta)x^4 + 1/t^4$ and $F_8 = x^8 + (\alpha/\beta)x^4 + t^4$, where $\alpha = t^8 + 1$ and $\beta = t^4 + 1$.

Using Algorithm Partitions we get the following partitions of the principal subfields $L_1, \dots, L_8$:

$$\begin{aligned}
P_1 &= \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}\} \\
P_2 &= \{\{1, 2\}, \{3, 4\}, \{5\}, \{6\}, \{7\}, \{8\}\} \\
P_3 &= \{\{1, 3\}, \{2, 4\}, \{5, 6\}, \{7, 8\}\} \\
P_4 &= \{\{1, 4\}, \{2, 3\}, \{5, 6\}, \{7, 8\}\} \\
P_5 &= \{\{1, 2, 5\}, \{3, 4, 6\}, \{7\}, \{8\}\} \\
P_6 &= \{\{1, 2, 6\}, \{3, 4, 5\}, \{7, 8\}\} \\
P_7 &= \{\{1, 2, 5, 7\}, \{3, 4, 6, 8\}\} \\
P_8 &= \{\{1, 2, 3, 4, 5, 6, 7, 8\}\}.
\end{aligned}$$

By joining the partitions of all subsets of $\{P_1, \dots, P_8\}$, we get the following new partitions:

$$\begin{aligned}
P_9 &= P_2 \vee P_4 = \{\{1, 2, 3, 4\}, \{5, 6\}, \{7, 8\}\} \\
P_{10} &= P_3 \vee P_6 = \{\{1, 2, 3, 4, 5, 6\}, \{7, 8\}\}.
\end{aligned}$$

Hence, $P_1, \dots, P_{10}$ are the partitions of every subfield of $\mathbb{Q}(t)/\mathbb{Q}(f(t))$. Next we compute all maximal chains of subfields. Recall that the subfield relation translates as refinement of partitions, for instance, $L_5 \subseteq L_2$, since P_2 refines P_5. Therefore, by looking at the partitions $P_1, \dots, P_{10}$, we see that one maximal chain of subfields is

$$\mathbb{Q}(f) = L_8 \subseteq L_7 \subseteq L_5 \subseteq L_2 \subseteq L_1 = \mathbb{Q}(t).$$

Now, let us find generators for these fields. As an example, let us find a generator for L_7. Following Theorem 5.1, let

$$g = \prod_{i \in P_7^{(1)}} F_i = F_1 F_2 F_5 F_7 = x^{12} - cx^8 - cx^4 - 1,$$

where $c = (t^{12} - 1)/(t^8 + t^4)$. Since $c \in K(t) \backslash K$, it follows that $L_7 = \mathbb{Q}(c)$. This yields the maximal chain of subfields:

$$\mathbb{Q}(f) \subseteq \mathbb{Q}(c) \subseteq \mathbb{Q}(t^4) \subseteq \mathbb{Q}(t^2) \subseteq \mathbb{Q}(t).$$

Finally, we compute the corresponding complete decomposition of f by computing left components. For instance, $\mathbb{Q}(f) \subseteq \mathbb{Q}\left(\frac{t^{12}-1}{t^8+t^4}\right)$ implies that there exists $g \in K(t)$ such that $f = g \circ \frac{t^{12}-1}{t^8+t^4}$. In this case we have $g = t^2$ and hence

$$f = t^2 \circ \frac{t^{12} - 1}{t^8 + t^4}.$$

Now $\mathbb{Q}(\frac{t^{12}-1}{t^8+t^4}) \subseteq \mathbb{Q}(t^4)$ and we can write $\frac{t^{12}-1}{t^8+t^4} = \frac{t^3-1}{t^2+t} \circ t^4$, and so on. This yields the following complete decomposition:

$$f = t^2 \circ \frac{t^3 - 1}{t^2 + t} \circ t^2 \circ t^2.$$

Doing this for every maximal chain of subfields yields all non-equivalent complete decompositions of f.

5.3 Timings

Finally, we compare our algorithm Decompose, which returns all non-equivalent complete decompositions of f, with the algorithms full_decomp and all_decomps from [3], which returns a single complete decomposition and all complete decompositions, respectively. All timings presented below also include the factorization time for $\Phi_f \in K(t)[x]$.

In the table below, n is the degree of $f \in K(t)$ and r is the number of irreducible factors of Φ_f. We also list d_p, the degree of the polynomial defining the good $K(t)$-ideal and $\#c$, the number of elements in K (or an extension of K, see Remark 5) used to determine the partitions $P_1, \ldots, P_r$.

Our algorithm better compares to all_decomps, since both algorithms return all non-equivalent complete decompositions of f. According to our experiments, for *small* values of r, the time spent by algorithm Decompose to compute all non-equivalent complete decompositions is similar to the time spent by full_decomp to compute a single decomposition. However, as r increases, we see a noticeable improvement compared to full_decomp and more so to all_decomps. More examples and details about these timings can be found at www.math.fsu.edu/~jszutkos/timings and the implementation at www.math.fsu.edu/~jszutkos/Decompose.

Table 1: Timings (in seconds)

n	r	d_p, $\#c$	Decompose	Ayad & Fleischmann (2008) [3]	
				full_decomp	all_decomps
12	7	3,1	0.01	0.02	0.03
24	8	1,4	0.02	0.00	0.09
144	10	1,4	1.82	1.88	101.08
24	10	3,1	0.02	0.01	0.20
18	12	4,1	0.05	0.06	0.81
24	14	4,1	0.07	0.51	10.57
60	17	5,1	0.18	91.68	981.43
60	17	1,8	0.77	485.19	4,338.47
96	26	2,4	0.42	211.30	$> 12h$
60	60	3,5	1.91	$> 12h$	n.a.
120	61	3,5	2.36	n.a.	n.a.
169	91	3,7	3.41	n.a.	n.a.
120	120	5,4	18.59	n.a.	n.a.
168	168	4,9	50.53	n.a.	n.a.

n.a.: not attempted.

REFERENCES

[1] V. S. Alagar and Mai Thanh. 1985. *Fast polynomial decomposition algorithms.* Springer Berlin Heidelberg, Berlin, Heidelberg, 150–153.

[2] Cesar Alonso, Jaime Gutierrez, and Tomas Recio. 1995. A Rational Function Decomposition Algorithm by Near-separated Polynomials. *Journal of Symbolic Computation* 19, 6 (1995), 527 – 544.

[3] Mohamed Ayad and Peter Fleischmann. 2008. On the decomposition of rational functions. *Journal of Symbolic Computation* 43, 4 (2008), 259 – 274.

[4] David R. Barton and Richard Zippel. 1985. Polynomial decomposition algorithms. *Journal of Symbolic Computation* 1, 2 (1985), 159 – 168.

[5] Dario Bini and Victor Y. Pan. 1994. *Polynomial and Matrix Computations (Vol. 1): Fundamental Algorithms.* Birkhauser Verlag, Basel, Switzerland, Switzerland.

[6] Raoul Blankertz. 2011. *Decomposition of Polynomials.* Master's thesis. Bonn, Germany. arXiv:1107.0687

[7] Raoul Blankertz. 2014. A Polynomial Time Algorithm for Computing All Minimal Decompositions of a Polynomial. *ACM* 48, 1/2 (2014), 13–23.

[8] A. Bostan, G. Lecerf, B. Salvy, É. Schost, and B. Wiebelt. 2004. Complexity Issues in Bivariate Polynomial Factorization. In *ISSAC '04.* ACM, New York, NY, USA, 42–49.

[9] Matthew Thomas Dickerson. 1989. *The Functional Decomposition of Polynomials.* Ph.D. Dissertation. Ithaca, NY, USA.

[10] Ralph Freese. 1997. Partition Algorithms (unpublished). *Available at* http://math.hawaii.edu/~ralph/Notes (1997).

[11] Ralph Freese. 2008. Computing congruences efficiently. *Algebra universalis* 59, 3 (2008), 337–343.

[12] Mark William Giesbrecht. 1988. *Some Results on the Functional Decomposition of Polynomials.* Master's thesis. Toronto, Ontario, Canada.

[13] Dexter Kozen and Susan Landau. 1989. Polynomial decomposition algorithms. *Journal of Symbolic Computation* 7, 5 (1989), 445 – 456.

[14] Grégoire Lecerf. 2007. Improved dense multivariate polynomial factorization algorithms. *Journal of Symbolic Computation* 42, 4 (2007), 477 – 494.

[15] J. F. Ritt. 1922. Prime and Composite Polynomials. *Trans. Amer. Math. Soc.* 23, 1 (1922), 51–66.

[16] Henning Stichtenoth. 2008. *Algebraic Function Fields and Codes* (2nd ed.). Springer Publishing Company.

[17] Jonas Szutkoski and Mark van Hoeij. 2016. The Complexity of Computing all Subfields of an Algebraic Number Field. (2016). arXiv:1606.01140

[18] B. L. van Der Waerden. 1964. *Modern Algebra.* New York.

[19] Mark van Hoeij, Jurgen Klueners, and Andrew Novocin. 2013. Generating subfields. *Journal of Symbolic Computation* 52 (2013), 17 – 34.

[20] Joachim von zur Gathen. 1990. Functional decomposition of polynomials: the tame case. *Journal of Symbolic Computation* 9, 3 (1990), 281 – 299.

[21] Joachim von zur Gathen. 1990. Functional decomposition of polynomials: the wild case. *Journal of Symbolic Computation* 10, 5 (1990), 437 – 452.

[22] Joachim von zur Gathen and Jurgen Gerhard. 2003. *Modern Computer Algebra* (2nd ed.). Cambridge University Press, NY, USA.

[23] Richard Zippel. 1991. Rational Function Decomposition. In *ISSAC '91.* ACM, New York, NY, USA, 1–6.