A RECOMBINATION ALGORITHM FOR THE DECOMPOSITION OF MULTIVARIATE RATIONAL FUNCTIONS

GUILLAUME CHÈZE

Abstract. In this paper we show how we can compute in a deterministic way the decomposition of a multivariate rational function with a recombination strategy. The key point of our recombination strategy is the used of Darboux polynomials. We study the complexity of this strategy and we show that this method improves the previous ones. In appendix, we explain how the strategy proposed recently by J. Berthomieu and G. Lecerf for the sparse factorization can be used in the decomposition setting. Then we deduce a decomposition algorithm in the sparse bivariate case and we give its complexity.

Introduction

The decomposition of univariate polynomials has been widely studied since 1922, see [Rit22], and efficient algorithms are known, see [AT85, BZ85, KL89, Gat90a, Gat90b, Gie88, Klü99]. There also exist results and algorithms in the multivariate case [Dic87, Gat90a, Gie88, GGR03].
The decomposition of rational functions has also been studied, [Gie88, Zip91, AGR95, GW95]. In the multivariate case the situation is the following:
Let $f(X_1,\dots,X_n)=f_1(X_1,\dots,X_n)/f_2(X_1,\dots,X_n)\in\mathbb{K}(X_1,\dots,X_n)$ be a rational function, where $\mathbb{K}$ is a field and $n\geq 2$. It is commonly said to be composite if it can be written $f=u(h)$ where $h(X_1,\dots,X_n)\in\mathbb{K}(X_1,\dots,X_n)$ and $u\in\mathbb{K}(T)$ such that $\deg(u)\geq 2$ (recall that the degree of a rational function is the maximum of the degrees of its numerator and denominator after reduction), otherwise f is said to be non-composite. In this paper, we give an algorithm which computes a non-composite rational function $h\in\mathbb{K}(X_1,\dots,X_n)$ and a rational function $u\in\mathbb{K}(T)$ such that $f=u(h)$.

In [Chè10], the author shows that we can reduce the decomposition problem to a factorization problem and gives a probabilistic and a deterministic algorithm. The probabilistic algorithm is nearly optimal: it performs $\tilde{\mathcal{O}}(d^n)$ arithmetic operations. The deterministic one computes $\mathcal{O}(d^2)$ absolute factorizations and then performs $\tilde{\mathcal{O}}(d^{n+\omega+2})$ arithmetic operations, where d is the degree of f and ω is the *feasible matrix multiplication* exponent as defined in [GG03, Chapter 12]. We recall that $2\leq\omega\leq 2.376$. As in [Chè10], we suppose in this work that d tends to infinity and n is fixed. We use the classical $\mathcal{O}$ and $\tilde{\mathcal{O}}$ ("soft $\mathcal{O}$") notation in the neighborhood of infinity as defined in [GG03, Chapter 25.7]. Informally speaking, "soft $\mathcal{O}$"s are used for readability in order to hide logarithmic factors in complexity estimates. In this paper we improve the complexity of the deterministic algorithm. With this algorithm we just compute two factorizations in $\mathbb{K}[X_1,\dots,X_n]$ and then we

Date: November 2, 2010.

use a recombination strategy. Under some hypotheses this new method performs $\tilde{\mathcal{O}}(d^{n+\omega-1})$ arithmetic operations.

The decomposition of multivariate rational functions appears when we study the kernel of a derivation, see [MO04]. In [MO04] the author uses Darboux polynomials and gives an algorithm which works with $\tilde{\mathcal{O}}(d^{\omega n})$ arithmetic operations.
In this paper, we are also going to use Darboux polynomials (see Section 1 for a definition) and we add a recombination strategy. Roughly speaking, we are going to factorize the numerator and the denominator and then thanks to a property of Darboux polynomials we are going to show that we can recombine the factors and deduce the decomposition.

The decomposition of multivariate rational functions also appears when we study intermediate fields of an unirational field and the extended Lüroth's Theorem, see [GRS01, Chè10], and when we study the spectrum of a rational function, see [Chè10] and the references therein.
The study of decomposition is an active area of research: for a study on multivariate polynomial systems see e.g. [FP09, FGP10], for a study on symbolic polynomials see e.g. [Wat09], for a study on Laurent polynomials see e.g. [Wat08], for effective results on the reduction modulo a prime number of a non-composite polynomial or a rational function see e.g. [CN10, BDN09, BCN], for combinatorial results see e.g. [Gat08].

In this paper, we improve the strategy proposed in [MO04]. As in [MO04], we consider fields with characteristic zero. Furthermore, as we want to give precise complexity estimate we are going to suppose that:

Hypothesis (C):
$\mathbb{K}$ is a number field: $\mathbb{K} = \mathbb{Q}[\alpha]$, α is an algebraic number of degree r.

As in [Chè10], we are going to suppose that the following hypothesis is satisfied:

Hypothesis (H):

$$\begin{cases} (i)\deg(f_1+\Lambda f_2) = \deg_{X_n}(f_1+\Lambda f_2), \text{ where } \Lambda \text{ is a new variable,} \\ (ii)\, R(\Lambda) = Res_{X_n}\Big(f_1(\underline{0},X_n)+\Lambda f_2(\underline{0},X_n),\, \partial_{X_n} f_1(\underline{0},X_n)+\Lambda \partial_{X_n} f_2(\underline{0},X_n)\Big) \neq 0 \text{ in } \mathbb{K}[\Lambda]. \end{cases}$$

where $\deg_{X_n} f$ represents the partial degree of f in the variable X_n, $\deg f$ is the total degree of f and Res_{X_n} denotes the resultant relatively to the variable X_n.

This hypothesis is necessary, because we will use the factorization algorithms proposed in [Lec07], where this kind of hypothesis is needed. Actually, in [Lec07] the author studies the factorization of a polynomial F and uses hypothesis (L), where (L) is the following:

Hypothesis (L):

$$\begin{cases} (i)\deg_{X_n} F = \deg F, \text{ and } F \text{ is monic in } X_n, \\ (ii) Res_{X_n}\big(F(\underline{0},X_n), \frac{\partial F}{\partial X_n}(\underline{0},X_n)\big) \neq 0. \end{cases}$$

If F is squarefree, then hypothesis (L) is not restrictive since it can be assured by means of a generic linear change of variables, but we will not discuss this question here (for a complete treatment in the bivariate case, see [CL07, Proposition 1]).

Roughly speaking, our hypothesis (H) is the hypothesis (L) applied to the polynomial $f_1 + \Lambda f_2$. In (H,i) we do not assume that $f_1 + \Lambda f_2$ is monic in X_n. Indeed, the leading coefficient relatively to X_n can be written: $a + \Lambda b$, with $a, b \in \mathbb{K}$. In our algorithm, we evaluate Λ to λ such that $a + \lambda b \neq 0$. Then we can consider the monic part of $f_1 + \lambda f_2$ and we get a polynomial satisfying (L,i). Then (H,i) is sufficient in our situation. Furthermore, in this paper, we assume f_1/f_2 to be reduced, i.e. f_1 and f_2 are coprime. We recall in Lemma 9 that in this situation $f_1 + \Lambda f_2$ is squarefree. Thus hypothesis (H) is not restrictive.

Furthermore, hypothesis (H) will also be useful in a preprocessing step, see Section 2. In this preprocessing step we reduce the decomposition to two factorizations of squarefree polynomials.

Complexity model. In this paper the complexity estimates charge a constant cost for each arithmetic operation (+, −, ×, ÷) and the equality test. All the constants in the base fields are thought to be freely at our disposal.

In this paper we suppose that *the number of variables n is fixed* and that the degree d tends to infinity.

Polynomials are represented by dense vectors of their coefficients in the usual monomial basis. For each integer d, we assume that we are given a computation tree that computes the product of two univariate polynomials of degree at most d with at most $\tilde{\mathcal{O}}(d)$ operations, independently of the base ring, see [GG03, Theorem 8.23]. Then with a Kronecker substitution we can compute the product of two multivariate polynomials with degree d with n variables with $\tilde{\mathcal{O}}(d^n)$ arithmetic operations. We also recall, see [GG03, Corollary 11.8], that if $\mathbb{K}$ is an algebraic extension of $\mathbb{Q}$ of degree r then each field operation in $\mathbb{K}$ takes $\tilde{\mathcal{O}}(r)$ arithmetic operations in $\mathbb{Q}$.

We use the constant ω to denote a *feasible matrix multiplication* exponent as defined in [GG03, Chapter 12]: two $n \times n$ matrices over $\mathbb{K}$ can be multiplied with $\mathcal{O}(n^\omega)$ field operations. As in [BP94] we require that $2 \leq \omega \leq 2.376$. We recall that the computation of a solution basis of a linear system with m equations and $d \leq m$ unknowns over $\mathbb{K}$ takes $\mathcal{O}(md^{\omega-1})$ operations in $\mathbb{K}$ [BP94, Chapter 2] (see also [Sto00, Theorem 2.10]).

In [Lec06, Lec07] the author gives a deterministic algorithm for the multivariate rational factorization. The rational factorization of a polynomial f is the factorization in $\mathbb{K}[\underline{X}]$, where $\mathbb{K}$ is the coefficient field of f. This algorithm uses one factorization of a univariate polynomial of degree d and $\tilde{\mathcal{O}}(d^{n+\omega-1})$ arithmetic operations, where d is the total degree of the polynomial and $n \geq 2$ is the number of variables.

Main Theorem. The following theorem gives the complexity result of our algorithm.

Theorem 1. *Let f be a multivariate rational function in $\mathbb{Q}[\alpha](X_1, \dots, X_n)$ of degree d, where α is an algebraic number of degree r. Under hypotheses (C) and (H), we can compute in a deterministic way the decomposition of f with $\tilde{\mathcal{O}}(rd^{n+\omega-1})$ arithmetic operations over $\mathbb{Q}$ plus two factorizations of univariate polynomials of degree d with coefficients in $\mathbb{Q}[\alpha]$.*

Comparison with other algorithms. There already exist several algorithms for the decomposition of rational functions. They all use the same global strategy: first compute h, and then deduce u. The first step is the difficult part of the problem.

In [Chè10], we explain how we can perform the second step, i.e. compute u from h and f, with $\tilde{\mathcal{O}}(d^n)$ arithmetics operations.
In [GRS01], the authors provide two algorithms to decompose a multivariate rational function. These algorithms run in exponential time in the worst case. In the first one we have to factorize polynomials with $2n$ variables $f_1(\underline{X})f_2(\underline{Y}) - f_1(\underline{Y})f_2(\underline{X})$ and to look for factors of the following kind $h_1(\underline{X})h_2(\underline{Y}) - h_1(\underline{Y})h_2(\underline{X})$. The authors say that in the worst case the number of candidates to be tested is exponential in $d = \deg(f_1/f_2)$. Indeed, the authors test all the possible factors.
In the second algorithm, for each pair of factors (h_1, h_2) of f_1 and f_2 (i.e. h_1 divides f_1 and h_2 divides f_2), we have to test if there exists $u \in \mathbb{K}(T)$ such that $f_1/f_2 = u(h_1/h_2)$. Thus in the worst case we also have an exponential number of candidates to be tested.
To the author's knowledge, the first polynomial time algorithm is due to J. Moulin-Ollagnier, see [MO04] . This algorithm relies on the study of the kernel of the following derivation: $\delta_\omega(F) = \omega \wedge dF$, where $F \in \mathbb{K}[\underline{X}]$ and $\omega = f_2 df_1 - f_1 df_2$. In [MO04] the author shows that we can reduce the decomposition of a rational function to linear algebra. The bottleneck of this algorithm is the computation of the kernel of a matrix. The size of this matrix is $\mathcal{O}(d^n) \times \mathcal{O}(d^n)$, then the complexity of this deterministic algorithm belongs to $\mathcal{O}(d^{n\omega})$.

The reduction of the decomposition problem to a factorization problem is classical, see e.g. [Klü99, Gie88, GW95, GRS01]. In [Chè10] the author shows that if we choose a probabilistic approach then two factorizations in $\mathbb{K}[X_1, \dots, X_n]$ are sufficient to get h and furthermore we do not have a recombination problem. This gives a nearly optimal algorithm. For the deterministic approach the author uses a property on the pencil $f_1 - \lambda f_2$ and shows that with $\mathcal{O}(d^2)$ absolute factorization (i.e. factorization in the algebraic closure of $\mathbb{K}$) we can get h. This deterministic strategy works with $\tilde{\mathcal{O}}(d^{n+\omega+2})$ arithmetic operations.
In this paper, we are going to show that we can obtain a deterministic algorithm with just two factorizations in $\mathbb{K}[X_1, \dots, X_n]$ and a recombination strategy. Our algorithm uses at most $\tilde{\mathcal{O}}(d^{n+\omega-1})$ arithmetic operations. This cost corresponds to the cost of the factorization and the recombination step.

Our recombination problem comes from this factorization:
If $f_1/f_2 = u_1/u_2(h_1/h_2)$ then

$$f_1 - \lambda f_2 = e(h_1 - t_1 h_2) \cdots (h_1 - t_k h_2)$$

where $\lambda, e \in \mathbb{K}$, $k = \deg(u_1/u_2)$ and t_i are the roots of the univariate polynomial $u_1(T) - \lambda u_2(T)$, see Lemma 10.
Thus with the factors $h_1 - t_1 h_2$ and $h_1 - t_2 h_2$ we can deduce h. Unfortunately these factors are not necessarily in $\mathbb{K}[X_1, \dots, X_n]$ and are not necessarily irreducible. In this paper we show how we can reduce the problem to a factorization problem in $\mathbb{K}[X_1, \dots, X_n]$ and how we can recombine the irreducible factors of $f_1 - \lambda f_2$ to get h.

We can see our recombination scheme as a *logarithmic derivative method*. Roughly speaking, the logarithmic derivative method works as follow:
If $F(X,Y) = \prod_{j=1}^{t} \mathcal{F}_j(X,Y)$, where $\mathcal{F}_j(X,Y) \in \mathbb{A}$ and $\mathbb{A} \supset \mathbb{K}[X,Y]$ (for example $\mathbb{A} = \mathbb{K}[[X]][Y]$), then we can write the irreducible factors $F_i(X,Y) \in \mathbb{K}[X,Y]$ of F

in the following way: $F_i = \prod_{j=1}^{t} \mathcal{F}_j^{e_{i,j}}$, where $e_{i,j} \in \{0,1\}$. Thus we just have to compute the exponents $e_{i,j}$ to deduce F_i. We compute these exponents thanks to this relation:

$$\frac{\partial_X F_i}{F_i} = \sum_{j=1}^{t} e_{i,j} \frac{\partial_X \mathcal{F}_j}{\mathcal{F}_j}.$$

With this relation the exponents $e_{i,j}$ are now coefficients, and we can compute them with linear algebra.
This strategy has already been used by several authors in order to factorize polynomials see e.g. [BHKS09, BLS+04, Lec06, CL07, Wei10]. Here, we use this kind of technique for the decomposition problem. With this strategy the recombination part of our algorithm corresponds to the computation of the kernel of a $\mathcal{O}(d^n) \times \mathcal{O}(d)$ matrix.
In our context, we do not use exactly a logarithmic derivative. We use a more general derivation, but we use the same idea: if a mathematical object transforms a product into a sum then the recombination problem becomes a linear algebra problem. In this paper this mathematical object is the cofactor, see Proposition 8.

Structure of this paper. In Section 1, we recall some results about the Jacobian derivative and Darboux polynomials. In Section 2, we describe a reduction step which eases the recombination strategy. In other words we explain how we can reduce the decomposition problem to a factorization problem. In Section 3, we explain how we can get h with a recombination strategy. In Section 4, we describe our algorithm with two examples. In Section 5 we conclude this paper with a remark on Darboux method and the logarithmic derivative method. In appendix, we explain how the strategy proposed recently by J. Berthomieu and G. Lecerf in [BL10] for the sparse factorization can be used in the decomposition setting. Then we deduce a decomposition algorithm in the sparse bivariate case and we give its complexity.

Notations. All the rational functions are supposed to be reduced.
Given a polynomial f, $\deg(f)$ denotes its total degree.
Given a rational function $f = f_1/f_2$, $\deg(f)$ denotes $\max\big(\deg(f_1), \deg(f_2)\big)$.
For the sake of simplicity, sometimes we write $\mathbb{K}[\underline{X}]$ instead of $\mathbb{K}[X_1, \dots, X_n]$, for $n \geq 2$.
$u \circ h$ means $u(h)$.
$Res(A, B)$ denotes the resultant of two univariate polynomials A and B.
$|S|$ is the cardinal of the set S.

1. Derivation and Darboux polynomials

We introduce the main tool of our algorithm.

Definition 2. A $\mathbb{K}$-derivation D of the polynomial ring $\mathbb{K}[X_1, \dots, X_n]$ is a $\mathbb{K}$-linear map from $\mathbb{K}[X_1, \dots, X_n]$ to itself that satisfies the Leibniz rule for the product

$$D(f.g) = D(f).g + f.D(g).$$

A $\mathbb{K}$-derivation has a unique extension to $\mathbb{K}(X_1, \dots, X_n)$ and then we will also denote by D the extended derivation.

Definition 3. Given a rational function f_1/f_2, the Jacobian derivative associated to f_1/f_2 is the following vector derivation, i.e. an $(n-1)$-tuple of derivations:

$$\begin{array}{rcl} D_{f_1/f_2} : \mathbb{K}[X_1,\dots,X_n] & \longrightarrow & \Big(\mathbb{K}[X_1,\dots,X_n]\Big)^{n-1} \\ F & \longmapsto & f_2^2 . \begin{pmatrix} \partial_{X_1}(f_1/f_2)\partial_{X_2}F - \partial_{X_2}(f_1/f_2)\partial_{X_1}F \\ \vdots \\ \partial_{X_1}(f_1/f_2)\partial_{X_n}F - \partial_{X_n}(f_1/f_2)\partial_{X_1}F \end{pmatrix}. \end{array}$$

The Jacobian derivative has the following property:

Proposition 4. *Given $f = f_1/f_2$ and $g \in \mathbb{K}(X_1,\dots,X_n) \setminus \mathbb{K}$ the following propositions are equivalent:*

1. *The rank of the Jacobian matrix*
$$Jac(f,g) = \begin{pmatrix} \dfrac{\partial f}{\partial X_1} & \cdots & \dfrac{\partial f}{\partial X_n} \\ \dfrac{\partial g}{\partial X_1} & \cdots & \dfrac{\partial g}{\partial X_n} \end{pmatrix}$$
is equal to one;
2. $D_{f_1/f_2}(g) = 0$*;*
3. *there exists h in $\mathbb{K}(X_1,\dots,X_n)$ such that $f = u(h)$ and $g = v(h)$ for $u, v \in \mathbb{K}(T)$.*

Proof. See [PI07] for a proof. In [PI07], $\mathbb{K}$ is supposed to be algebraically closed. However, we can remove this hypothesis because we have the equivalence: f is composite over $\mathbb{K}$ if and only if f is composite over $\overline{\mathbb{K}}$, see e.g. [BCN, Theorem 13]. □

Definition 5. Given D a vector derivation i.e. an m-tuple of derivations, a polynomial $F \in \mathbb{K}[\underline{X}]$ is said to be a Darboux polynomial of D if there exists $\mathcal{G} \in \big(\mathbb{K}[\underline{X}]\big)^m$ such that $D(F) = F.\mathcal{G}$. $\mathcal{G}$ is called the cofactor of F for the derivation D.

We deduce easily the following classical propositions.

Proposition 6. *f_1 and f_2 are Darboux polynomials of D_{f_1/f_2}.*

Proposition 7. *$D_{f_1/f_2}(h_1/h_2) = 0$ if and only if h_1 and h_2 are Darboux polynomials with the same cofactor.*

The following proposition is the main tool of our algorithm. Indeed, this proposition shows that cofactors transform a product into a sum. Then thanks to the cofactors it will be possible to apply a kind of logarithmic derivative recombination scheme.

Proposition 8. *Let $F \in \mathbb{K}[X_1,\dots,X_n]$ be a polynomial and let $F = F_1^{e_1}\cdots F_r^{e_r}$ be its irreducible factorization in $\mathbb{K}[X_1,\dots,X_n]$. Then:*
F is a Darboux polynomial with cofactor $\mathcal{G}_F$ if and only if all the F_i are Darboux polynomials with cofactor $\mathcal{G}_{F_i}$. Furthermore, $\mathcal{G}_F = e_1\mathcal{G}_{F_1} + \cdots + e_r\mathcal{G}_{F_r}$.

Proof. See for example Lemma 8.3 page 216 in [DLA06]. □

2. Reduction to a rational factorization problem

In this section, we recall how the decomposition problem can be reduced to a factorization problem. Furthermore, we show that we can reduce our problem to a situation where f_1 and f_2 are squarefree. First, we recall some useful lemmas.

Lemma 9. *If f_1/f_2 is reduced in $\mathbb{K}(X_1, \dots, X_n)$, where $n \geq 1$ and Λ is a variable, then $f_1 + \Lambda f_2$ is squarefree.*

Lemma 10. *Let $h = h_1/h_2$ be a rational function in $\mathbb{K}(\underline{X})$, $u = u_1/u_2$ a rational function in $\mathbb{K}(T)$ and set $f = u \circ h$ with $f = f_1/f_2 \in \mathbb{K}(\underline{X})$. For all $\lambda \in \mathbb{K}$ such that $\deg(u_1 - \lambda u_2) = \deg u$, we have*

$$f_1 - \lambda f_2 = e(h_1 - t_1 h_2) \cdots (h_1 - t_k h_2)$$

where $e \in \mathbb{K}$, $k = \deg u$ and t_i are the roots of the univariate polynomial $u_1(T) - \lambda u_2(T)$.

Proof. See [Chè10, Lemma 8, Lemma 39]. □

Remark 11. If $\lambda = f_1(\underline{a})/f_2(\underline{a})$, where $\underline{a} = (a_1, \dots, a_n) \in \mathbb{K}^n$, then we can suppose that $t_1 \in \mathbb{K}$. Indeed, $t_1 = h_1(\underline{a})/h_2(\underline{a}) \in \mathbb{K}$.

The following lemma says that we can always suppose that $\deg u_1 = \deg u_2 = \deg u$.

Lemma 12. *Let $h = h_1/h_2$ be a rational function in $\mathbb{K}(\underline{X})$, $u = u_1/u_2$ a rational function in $\mathbb{K}(T)$ and set $f = u \circ h$ with $f = f_1/f_2 \in \mathbb{K}(\underline{X})$. There exists an homography $H(T) = (aT + b)/(\alpha T + \beta) \in \mathbb{K}(T)$ such that:*
$u \circ H = \tilde{u}_1/\tilde{u}_2$, $\deg \tilde{u}_1 = \deg \tilde{u}_2$, and $f = \dfrac{\tilde{u}_1}{\tilde{u}_2} \circ \tilde{h}$, where $\tilde{h} = H^{-1} \circ h$ and H^{-1} is the inverse of H for the composition.

Proof. If $\deg u_1 = \deg u_2$ then we set $H(T) = T$.
If $\deg u_2 > \deg u_1$ then we have:

$$\frac{u_1}{u_2}\big(H(T)\big) = \frac{\prod_{i=1}^{\deg u_1} \big(aT + b - \lambda_i(\alpha T + \beta)\big)}{\prod_{i=1}^{\deg u_2} \big(aT + b - \mu_i(\alpha T + \beta)\big)} . (\alpha T + \beta)^{\deg u_2 - \deg u_1},$$

where $u_1(\lambda_i) = 0$ and $u_2(\mu_i) = 0$.
We set:

$$\begin{aligned}
\tilde{u}_1(T) &= (\alpha T + \beta)^{\deg u_2 - \deg u_1} . \prod_{i=1}^{\deg u_1} \big(aT + b - \lambda_i(\alpha T + \beta)\big) \\
&= u_1\big(H(T)\big).(\alpha T + \beta)^{\deg u_2} \in \mathbb{K}[T] \\
\tilde{u}_2(T) &= \prod_{i=1}^{\deg u_2} \big(aT + b - \mu_i(\alpha T + \beta)\big) \\
&= u_2\big(H(T)\big).(\alpha T + \beta)^{\deg u_2} \in \mathbb{K}[T].
\end{aligned}$$

If $a - \lambda_i \alpha \neq 0$, $\alpha \neq 0$, and $a - \mu_i \alpha \neq 0$ then we get $\deg \tilde{u}_1 = \deg u_2 = \deg \tilde{u}_2$.
To conclude the proof we just have to remark that $\deg H = 1$, thus H is invertible for the composition. □

In order to ease the recombination scheme we reduce our problem to a situation where *the rational function is squarefree, i.e. the numerator and the denominator are squarefree.* The following algorithm shows that if f_1 or f_2 are not squarefree then we can compute an homography $U(T) \in \mathbb{K}(T)$ such that $U(f_1/f_2)$ is squarefree. Furthermore, if we know a decomposition $U(f_1/f_2) = u(h)$ then we can easily deduce a decomposition $f_1/f_2 = U^{-1}\big(u(h)\big)$. We recall that U is invertible for the composition because $\deg U = 1$. Now, we describe an algorithm which computes a good homography.

Good homography

Input: $f = f_1/f_2 \in \mathbb{K}(X_1, \dots, X_n)$ of degree d, such that (C) and (H) are satisfied and a finite subset S of $\mathbb{K}^n$ such that $|S| = 2d^2 + 2d$.
Output: $U(T) = (T-\lambda_a)/(T-\lambda_b)$ such that $U(f)$ is squarefree, $\lambda_a = f_1/f_2(\underline{a}), \lambda_b = f_1/f_2(\underline{b})$ where $\underline{a}, \underline{b} \in \mathbb{K}^n$, $\lambda_a \neq \lambda_b$, and $\deg_{X_n}(f_1 - \lambda_a f_2) = \deg_{X_n}(f_1 - \lambda_b f_2) = d$.

(1) Compute $\overline{f}_1(X_n) := f_1(\underline{0}, X_n)$, and $\overline{f}_2(X_n) := f_2(\underline{0}, X_n)$.
(2) Construct an empty list L.
(3) For i from 1 to $2d^2 + 2d$ do:
    (a) Compute $\overline{f} := \overline{f}_1(i)/\overline{f}_2(i),$.
    (b) If $\overline{f} \notin L$ then $L :=$ concatenate$(L, [\overline{f}])$.
(4) Construct an empty list $\mathcal{L}$.
(5) For k from 1 to $2d + 2$ do:
    (a) Compute $R := Res_{X_n}\Big(\overline{f}_1(X_n) - L[k]\overline{f}_2(X_n),\, \partial_{X_n}\overline{f}_1(X_n) - L[k]\partial_{X_n}\overline{f}_2(X_n)\Big)$.
    (b) If $R \neq 0$ and $\deg_{X_n}(\overline{f}_1 - L[k]\overline{f}_2) = d$, then $\mathcal{L} :=$ concatenate$(\mathcal{L}, [L[k]])$.
(6) $\lambda_a := \mathcal{L}[1]$, $\lambda_b := \mathcal{L}[2]$.
(7) Return $U(T) = (T - \lambda_a)/(T - \lambda_b)$.

Proposition 13. *The algorithm* Good homography *is correct.*

Proof. In Step 3 we construct a list with at least $2d + 2$ distinct elements because $\deg(f) = d$.
By hypothesis (H), $R(\Lambda) \neq 0$ and by [GG03, Theorem 6.22], $\deg(R) \leq 2d - 1$. Thus $\mathcal{L}$ contains at least two distincts elements.
As $R(\lambda_a)$ and $R(\lambda_b)$ are not equal to zero, and thanks to Step 5b the condition on the degree is satisfied, we deduce that $f_1 - \lambda_a f_2$ and $f_1 - \lambda_b f_2$ are squarefree. □

Proposition 14. *The algorithm* Good homography *can be performed with at most $\tilde{\mathcal{O}}(d^n)$ arithmetic operations over $\mathbb{K}$.*

Proof. Step 1 can be done with $\tilde{\mathcal{O}}(d^n)$ arithmetic operations with Horner's method.
In Step 3 we use a fast multipoint evaluation strategy, then we can perform this step with at most $\tilde{\mathcal{O}}(d^2)$ arithmetic operations, see [GG03, Corollary 10.8].
In Step 5, the computation of the resultant can be done with $\tilde{\mathcal{O}}(d)$ arithmetic operations, see [GG03, Corollary 11.16]. Thus Step 5 can be done with $\tilde{\mathcal{O}}(d^2)$ arithmetic operations.
In conclusion the algorithm can be performed with the desired complexity. □

Remark 15. Suppose $f_1/f_2 = v_1/v_2(h)$. With the algorithm Good homography we can write $U(f_1/f_2) = u_1/u_2(h)$ with $u_1/u_2 \in \mathbb{K}(T)$, $h \in \mathbb{K}(X_1, \dots, X_n)$, and u_1 (resp. u_2) has a root α_1 (resp. α_2) in $\mathbb{K}$. Indeed, we have $u_1 = v_1 - \lambda_a v_2$ (resp.

$u_2 = v_1 - \lambda_b v_2$) and $\lambda_a = f_1/f_2(\underline{a})$ (resp. $\lambda_b = f_1/f_2(\underline{b})$) then we deduce that $\alpha_1 = h_1/h_2(\underline{a})$ (resp. $\alpha_2 = h_1/h_2(\underline{b})$).

3. The recombination method

In this section we describe our recombination method. First, we introduce some notations. By Proposition 6, F_1 and F_2 are Darboux polynomials of D_{F_1/F_2}.We denote by

$$\mathcal{G}_{F_k} = (\mathcal{G}^{(2)}_{F_k}, \dots, \mathcal{G}^{(n)}_{F_k})$$

the cofactor of F_k, where $k = 1, 2$, and $\mathcal{G}^{(l)}_{F_k} \in \mathbb{K}[X_1, \dots, X_n]$. We set:

$$F_k = \prod_{j=1}^{s_k} F_{k,j}$$

for $k = 1, 2$, and

$$\mathcal{G}_{F_{k,j}} = (\mathcal{G}^{(2)}_{F_{k,j}}, \dots, \mathcal{G}^{(n)}_{F_{k,j}}).$$

In $\mathbb{Q}[\alpha][X_1, \dots, X_n]$ polynomials are denoted in the following way:

$$\mathcal{P} = \sum_{|\tau| \le d} \sum_{\epsilon=0}^{r-1} a_{\epsilon,\tau} \alpha^\epsilon X_1^{\tau_1} \cdots X_n^{\tau_n} \in \mathbb{Q}[\alpha][X_1, \dots, X_n],$$

where α is an algebraic number of degree r, $\tau = (\tau_1, \dots, \tau_n)$, $|\tau| = \tau_1 + \cdots + \tau_n$, and $a_{\epsilon,\tau} \in \mathbb{Q}$. We set

$$\operatorname{coef}\big(\mathcal{P}, \alpha^\epsilon \underline{X}^\tau\big) = a_{\epsilon,\tau}.$$

Now we define the linear system $\mathcal{S}$:

$$\mathcal{S} := \sum_{j=1}^{s_1} x_{1,j} \operatorname{coef}\big(\mathcal{G}^{(l)}_{F_{1,j}}, \alpha^\epsilon \underline{X}^\tau\big) - \sum_{j=1}^{s_2} x_{2,j} \operatorname{coef}\big(\mathcal{G}^{(l)}_{F_{2,j}}, \alpha^\epsilon \underline{X}^\tau\big) = 0,$$

where $|\tau| \le d$, $0 \le \epsilon \le r-1$, and $2 \le l \le n$.
We denote by $\ker \mathcal{S}$ the kernel of this linear system, and we remark that

$$x = (x_{1,1}, \dots, x_{2,s_2}) \in \ker \mathcal{S} \iff \sum_{j=1}^{s_1} x_{1,j} \mathcal{G}_{F_{1,j}} - \sum_{j=1}^{s_2} x_{2,j} \mathcal{G}_{F_{2,j}} = 0.$$

We define the following maps:

$$\begin{array}{rcl} \pi_1 : \mathbb{K}^{s_1+s_2} & \longrightarrow & \mathbb{K}^{s_1} \\ (x_{1,1}, \dots, x_{2,s_2}) & \longmapsto & (x_{1,1}, \dots, x_{1,s_1}) \end{array}$$

$$\begin{array}{rcl} \pi_2 : \mathbb{K}^{s_1+s_2} & \longrightarrow & \mathbb{K}^{s_2} \\ (x_{1,1}, \dots, x_{2,s_2}) & \longmapsto & (x_{2,1}, \dots, x_{2,s_2}) \end{array}$$

The following proposition will be the key of our algorithm:

Proposition 16. *Suppose that $F_1/F_2 \in \mathbb{K}(X_1, \dots, X_n)$ comes from the algorithm* Good Homography *and $F_1/F_2 = u(h)$ where $h = h_1/h_2 \in \mathbb{K}(X_1, \dots, X_n)$ is a non-composite reduced rational function and $u = u_1/u_2 \in \mathbb{K}(T)$ is a reduced rational function, with $\deg u_1 = \deg u_2$.*
We denote by $u_k = \prod_{i=1}^{t_k} u_{k,i}$ the factorization of u_k in $\mathbb{K}[T]$, where $k = 1$, 2.
We denote by $F_k = \prod_{j=1}^{s_k} F_{k,j}$ the factorization of F_k in $\mathbb{K}[X_1, \dots, X_n]$, where

$k = 1,\ 2$.
Then:

(1) $u_{k,i}\Big(\dfrac{h_1}{h_2}\Big).h_2^{\deg u_{k,i}} = \prod_{j=1}^{s_k} F_{k,j}^{e_{k,i,j}} \in \mathbb{K}[X_1,\dots,X_n]$ *and* $e_{k,i,j} \in \{0,1\}$.
Furthermore, if we set $e_{k,i} := (e_{k,i,1},\dots,e_{k,i,s_k})$*, then the vectors* $e_{k,i}$*,* $i = 1,\dots,t_k$*, are orthogonal for the usal scalar product.*
(2) *We have* $e_{k,i} \in \pi_k(\ker \mathcal{S})$.
(3) $\{e_{k,1},\dots,e_{k,t_k}\}$ *is a basis of* $\pi_k(\ker \mathcal{S})$.

Proof. (1) By Lemma 10 applied to F_1/F_2 (resp. F_2/F_1) with $\lambda = 0$, we get

$$F_k = u_k(h_1/h_2).h_2^{\deg u_k} = \prod_{i=1}^{t_k} u_{k,i}(h_1/h_2).h_2^{\deg u_{k,i}}.$$

Then we deduce

$$u_{k,i}(h_1/h_2).h_2^{\deg u_{k,i}} = \prod_{j=1}^{s_k} F_{k,j}^{e_{k,i,j}} \text{ in } \mathbb{K}[X_1,\dots,X_n]$$

with $e_{k,i,j} \in \{0,1\}$ because F_k are squarefree. Furthermore, the vectors $e_{k,i}$ are orthogonal for the usual scalar product because F_k are squarefree.

(2) We show this item for $k = 1$, the case $k = 2$ can be proved in a similar way. As F_1/F_2 comes from the algorithm Good Homography and as explained in Remark 15 we can suppose that:

$$u_{k,1}(T) = (T - \alpha_k), \text{ with } \alpha_k \in \mathbb{K}.$$

The previous item allows us to write:

$$\frac{u_{1,i}}{u_{2,1}^{\deg u_{1,i}}}\left(\frac{h_1}{h_2}\right) = \frac{\Big(\prod_{j=1}^{s_1} F_{1,j}^{e_{1,i,j}}\Big).\Big(h_2\Big)^{\deg u_{1,i}}}{\Big(\prod_{j=1}^{s_2} F_{2,j}^{e_{2,1,j}}\Big)^{\deg u_{1,i}}.\Big(h_2\Big)^{\deg u_{1,i}}} = \frac{\prod_{j=1}^{s_1} F_{1,j}^{e_{1,i,j}}}{\Big(\prod_{j=1}^{s_2} F_{2,j}^{e_{2,1,j}}\Big)^{\deg u_{1,i}}}.$$

By Proposition 4 applied to $\dfrac{u_{1,i}}{u_{2,1}^{\deg u_{1,i}}}\Big(\dfrac{h_1}{h_2}\Big)$, we get then:

$$D_{F_1/F_2}\left(\frac{\prod_{j=1}^{s_1} F_{1,j}^{e_{1,i,j}}}{\prod_{j=1}^{s_2} F_{2,j}^{e_{2,1,j}.\deg u_{1,i}}}\right) = 0.$$

Now, we recall that $F_{k,j}$ are Darboux polynomials, see Proposition 6 and Proposition 8. Then by Proposition 8, we deduce

$$\sum_{j=1}^{s_1} e_{1,i,j}\mathcal{G}_{F_{1,j}} - \deg(u_{1,i})\sum_{j=1}^{s_2} e_{2,1,j}\mathcal{G}_{F_{2,j}} = 0.$$

It follows $(e_{1,i,1},\dots,e_{1,i,s_1},\deg(u_{1,i}).e_{2,1,1},\dots,\deg(u_{1,i}).e_{2,1,s_2}) \in \ker \mathcal{S}$. Thus, $e_{1,i} \in \pi_1(\ker \mathcal{S})$.

(3) The vectors $e_{k,1},\dots,e_{k,t_k}$ are linearly independant because they are orthogonal. We just have to prove that these vectors generate $\pi_k(\ker \mathcal{S})$.
Suppose that $\rho = (\rho_1,\dots,\rho_{s_1+s_2}) \in \ker \mathcal{S}$. First, we clear the denominators and we suppose that $\rho \in \mathbb{Z}^{s_1+s_2}$ instead of $\mathbb{Q}^{s_1+s_2}$.
In a first time we explain the strategy of the proof for this item, and in a

second time we will detail the proof.
We set

$$\frac{\mathcal{F}_1}{\mathcal{F}_2} = \frac{\prod_{i=1}^{s_1} F_{1,j}^{\rho_j}}{\prod_{j=1}^{s_2} F_{2,j}^{\rho_{s_1+j}}},$$

where $\mathcal{F}_1, \mathcal{F}_2 \in \mathbb{K}[\underline{X}]$ and $\mathcal{F}_1/\mathcal{F}_2$ is a reduced rational function.

Our goal is to get this kind of equality:

$$(\mathcal{E}),\ \frac{\mathcal{F}_1}{\mathcal{F}_2} = \frac{\prod_{j=1}^{s_1} F_{1,j}^{\rho_j}}{\prod_{j=1}^{s_2} F_{2,j}^{\rho_{s_1+j}}} = \frac{\prod_{k=1}^{2} \prod_{(i,k)\in I_{\mathrm{num}}} \left(\prod_{j=1}^{k} F_{k,j}^{e_{k,i,j}}\right)^{m_{u_{k,i}}}}{\prod_{k=1}^{2} \prod_{(i,k)\in I_{\mathrm{den}}} \left(\prod_{j=1}^{k} F_{k,j}^{e_{k,i,j}}\right)^{m_{u_{k,i}}}},$$

where $m_{u_{k,i}} \in \mathbb{N}$, $I = \{(1,1), \dots, (t_1,1), (1,2), \dots, (t_2,2)\}$, $I_{\mathrm{num}} \subset I$, $I_{\mathrm{den}} \subset I$ and $I_{\mathrm{num}} \cap I_{\mathrm{den}} = \emptyset$.

By the unicity of the factorization in irreducible factors we deduce:

$$\pi_1(\rho) = \sum_{(i,1)\in I_{\mathrm{num}}} m_{u_{1,i}} e_{1,i} - \sum_{(i,1)\in I_{\mathrm{den}}} m_{u_{1,i}} e_{1,i},$$

$$\pi_2(\rho) = \sum_{(i,2)\in I_{\mathrm{num}}} m_{u_{2,i}} e_{2,i} - \sum_{(i,2)\in I_{\mathrm{den}}} m_{u_{2,i}} e_{2,i}.$$

We get: $\{e_{k,1}, \dots, e_{k,t_k}\}$ generates $\pi_k(\ker \mathcal{S})$, and this is the desired result.

Now we detail the proof with four steps:

(a) We remark:

$$\frac{F_1}{F_2} = \frac{u_1}{u_2}(h) = \frac{\prod_{i=1}^{\deg u}(h_1 - \mu_{1,i} h_2)}{\prod_{i=1}^{\deg u}(h_1 - \mu_{2,i} h_2)},$$

where $\mu_{k,i}$ are roots of u_k.

(b) We have:

$$\frac{\mathcal{F}_1}{\mathcal{F}_2} = \frac{\prod_{j=1}^{d_1}(h_1 - \lambda_j h_2)^{m_j}}{\prod_{j=d_1+1}^{d_1+d_2}(h_1 - \lambda_j h_2)^{m_j}}.h_2^{\kappa},\ \text{with } \kappa \in \mathbb{Z}, m_j \in \mathbb{N}.$$

Indeed, as $\rho \in \ker \mathcal{S}$, we have

$$\sum_{j=1}^{s_1} \rho_j \mathcal{G}_{F_{1,j}} - \sum_{j=1}^{s_2} \rho_{s_1+j} \mathcal{G}_{F_{2,j}} = 0.$$

Thus $\prod_{j=1}^{s_1} F_{1,j}^{\rho_j}$ and $\prod_{j=1}^{s_2} F_{2,j}^{\rho_{s_1+j}}$ are Darboux polynomials with the same cofactor. By Proposition 7, we deduce:

$$D_{F_1/F_2}\left(\frac{\prod_{j=1}^{s_1} F_{1,j}^{\rho_j}}{\prod_{j=1}^{s_2} F_{2,j}^{\rho_{s_1+j}}}\right) = 0.$$

Then $D_{F_1/F_2}(\mathcal{F}_1/\mathcal{F}_2) = 0$ and thus $\mathcal{F}_1/\mathcal{F}_2 = v_1/v_2(h)$ by Proposition 4. We denote by λ_j the roots of v_1 and v_2 and we get the desired result.

(c) We claim:

$$\frac{\mathcal{F}_1}{\mathcal{F}_2}=\frac{\prod_{k=1}^2\prod_{(i,k)\in I_{\mathrm{num}}}\Big(u_{k,i}(h_1/h_2)h_2^{\deg u_{k,i}}\Big)^{m_{u_{k,i}}}}{\prod_{k=1}^2\prod_{(i,k)\in I_{\mathrm{den}}}\Big(u_{k,i}(h_1/h_2)h_2^{\deg u_{k,i}}\Big)^{m_{u_{k,i}}}},\text{ where } m_{u_{k,i}}\in\mathbb{N}.$$

Indeed, we have: for all j there exists $\delta(j)$ such that $\lambda_j=\mu_{\delta(j)}$.
(To prove this remark we suppose the converse: There exists j_0 such that $\lambda_{j_0}\neq\mu_{k,i}$, for $k=1,2$ and $i=1,\dots,\deg u$.
By definition of $\mathcal{F}_k$ and by step 3b, there exists (k_1,j_1) such that F_{k_1,j_1} and $h_1-\lambda_{j_0}h_2$ have a common factor in $\mathbb{C}[\underline{X}]$. We call $\mathcal{P}$ this common factor.
By step 3a, there exists (k_2,i_2) such that $\mathcal{P}$ is a factor of $h_1-\mu_{k_2,i_2}h_2$. Thus $h_1-\lambda_{j_0}h_2$ and $h_1-\mu_{k_2,i_2}h_2$ have a common factor. As $\lambda_{j_0}\neq\mu_{k_2,i_2}$ we deduce that $\mathcal{P}$ divides h_1 and h_2. This is absurd because h_1/h_2 is reduced.)
Thus $\kappa=0$, and for all j there exists $k(j)\in\{1,2\}$ and such that $u_{k(j)}(\lambda_j)=0$.
As $v_1,v_2\in\mathbb{K}[T]$, by conjugation, we deduce that if λ_j and $\lambda_{j'}$ are roots of the same irreducible polynomial $u_{k,i}\in\mathbb{K}[T]$ then $m_j=m_{j'}$. We denote by $m_{u_{k,i}}$ this common value.
This gives the claimed equality with $I_{\mathrm{num}}\cap I_{\mathrm{den}}=\emptyset$, because $\mathcal{F}_1/\mathcal{F}_2$ is reduced.

(d) Now we can prove equality $(\mathcal{E})$.

$$\begin{aligned}\frac{\mathcal{F}_1}{\mathcal{F}_2}&=\frac{\prod_{k=1}^2\prod_{(i,k)\in I_{\mathrm{num}}}\Big(u_{k,i}(h_1/h_2)h_2^{\deg u_{k,i}}\Big)^{m_{u_{k,i}}}}{\prod_{k=1}^2\prod_{(i,k)\in I_{\mathrm{den}}}\Big(u_{k,i}(h_1/h_2)h_2^{\deg u_{k,i}}\Big)^{m_{u_{k,i}}}},\text{ by step 3}c,\\&=\frac{\prod_{k=1}^2\prod_{(i,k)\in I_{\mathrm{num}}}\Big(\prod_{j=1}^k F_{k,j}^{e_{k,i,j}}\Big)^{m_{u_{k,i}}}}{\prod_{k=1}^2\prod_{(i,k)\in I_{\mathrm{den}}}\Big(\prod_{j=1}^k F_{k,j}^{e_{k,i,j}}\Big)^{m_{u_{k,i}}}},\text{ by the first item.}\end{aligned}$$

This gives the desired equality $(\mathcal{E})$.

□

Now we describe our recombination algorithm:

Recombination for Decomposition

Input: $f=f_1/f_2\in\mathbb{K}(X_1,\dots,X_n)$, such that (C) and (H) are satisfied.
Output: A decomposition of f if it exists, with $f=u\circ h$, $u=u_1/u_2$ with $\deg u\geq 2$, and $h=h_1/h_2$ non-composite.

(1) Compute $F=F_1/F_2:=U(f)$ with the algorithm Good homography.
(2) For k=1, 2, factorize $F_k=\prod_{i=1}^{s_k}F_{k,i}$ in $\mathbb{K}[\underline{X}]$ with $F_{k,i}$ irreducible.
(3) For each $F_{k,i}$ compute the corresponding cofactor $\mathcal{G}_{F_{k,i}}:=D_{F_1/F_2}(F_{k,i})/F_{k,i}$.
(4) Build the system $\mathcal{S}$ and compute the basis in reduced row echelon form $\mathcal{B}_1$ of $\pi_1(\ker\mathcal{S})$ and $\mathcal{B}_2$ of $\pi_2(\ker\mathcal{S})$.
(5) For k=1, 2, find $v_k=(v_{k,1},\dots,v_{k,s_k})\in\mathcal{B}_k$ such that: $\sum_{i=1}^{s_k}v_{k,i}\deg F_{k,i}=\min_{w\in\mathcal{B}_k}\sum_{i=1}^{s_k}w_i\deg F_{k,i}$, where $w:=(w_1,\dots,w_{s_k})$.
(6) For k=1, 2, compute $H_k:=\prod_{i=1}^{s_k}F_{k,i}^{v_{k,i}}$.

(7) Set $H := H_1/H_2$.
(8) Compute u such that $u(H) = f$.
(9) Return H, and u.

Proposition 17. *The algorithm Recombination for Decomposition is correct.*

Proof. Consider $F_1/F_2 := U(f)$. As we want to decompose f_1/f_2, we just have to decompose F_1/F_2, because $\deg U = 1$ and then U is invertible.
As F_1/F_2 comes from the algorithm Good Homography we can suppose, see Remark 15, that $u_{k,1}(T) = (T - \alpha_k)$ with $\alpha_k \in \mathbb{K}$, and $k = 1, 2$. Furthermore, by Lemma 12 we can also suppose that $\deg u_1 = \deg u_2$.
Then by Proposition 16, the basis $\mathcal{B}_k$ of $\pi_k(\ker \mathcal{S})$ are $\{e_{k,1}, \dots, e_{k,t_k}\}$.
The vector $e_{k,i}$ gives the polynomial $\mathcal{H}_{k,i} = \prod_{j=1}^{s_k} F_{k,j}^{e_{k,i,j}} = u_{k,i}(h)h_2^{\deg u_{k,i}}$.
Furthermore $\deg \mathcal{H}_{k,i} = \sum_{j=1}^{s_k} e_{k,i,j} \deg F_{k,j} = \deg u_{k,i} \deg h$. Thus in Step 5

$$\min_{w \in \mathcal{B}_k} \sum_{i=1}^{s_k} w_i \deg F_{k,i} = \deg h,$$

because this minimum is reached with $e_{k,1} \in \mathcal{B}_k$. Hence v_k in Step 6 gives $H_k = u_{k,i(k)}(h)h_2^{\deg u_{k,i(k)}}$ with $\deg u_{k,i(k)} = 1$.
It follows $H = (h_1 - \alpha h_2)/(h_1 - \beta h_2)$ with $\alpha, \beta \in \mathbb{K}$. Thus $H = v(h)$ with $\deg v = 1$, then the algorithm is correct.

□

Proposition 18. *The algorithm Recombination for Decomposition can be performed with $\tilde{\mathcal{O}}(rd^{n+\omega-1})$ arithmetic operations over $\mathbb{Q}$ and two factorizations of univariate polynomials of degree d with coefficients in $\mathbb{K}$.*

We recall that in our complexity analysis the number of variables is fixed and the degree d tends to infinity.

Proof. Step 1 uses $\tilde{\mathcal{O}}(d^n)$ arithmetic operations over $\mathbb{K}$ by Proposition 14, thus it uses $\tilde{\mathcal{O}}(rd^n)$ arithmetic operations over $\mathbb{Q}$.
Step 2 uses $\tilde{\mathcal{O}}(d^{n+\omega-1})$ arithmetic operations over $\mathbb{K}$ because we can use Lecerf's algorithm, see [Lec07]. Thus we use $\tilde{\mathcal{O}}(rd^{n+\omega-1})$ arithmetic operations over $\mathbb{Q}$ and two factorizations of univariate polynomials of degree d with coefficients in $\mathbb{K}$.
In Step 3, we compute $D_{F_1/F_2}(F_{k,i})$, thus we perform $2(n-1)$ multiplications of multivariate polynomials. We can do this with a fast multiplication technique, and then this computation costs $\tilde{\mathcal{O}}(nrd^n)$ arithmetic operations over $\mathbb{Q}$. Then we divide $D_{F_1/F_2}(F_{k,i})$ by $F_{k,i}$. We have to perform $n-1$ exact divisions, thus with a Kronecker subsitution we reduce this problem to $n-1$ univariate divisions, and the cost of one such division belongs then to $\tilde{\mathcal{O}}(rd^n)$. As s_1 and s_2 are smaller than d, Step 3 costs $\tilde{\mathcal{O}}(nrd^{n+1})$ arithmetic operations over $\mathbb{Q}$.
Step 4 needs $\tilde{\mathcal{O}}(nrd^n d^{\omega-1})$ arithmetic operations over $\mathbb{Q}$ with Storjohann's method, see [Sto00, Theorem 2.10]. Indeed, $\mathcal{S}$ has $\mathcal{O}((n-1)rd^n)$ equations and $s_1 + s_2$ unknowns, thus at most $2d$ unknowns.
Step 5 has a negligeable cost because $\dim_{\mathbb{Q}} \pi_k(\ker \mathcal{S}) = t_k$ is smaller than d and s_k is also smaller than d.
In Step 6, we use a fast multiplication technique and we compute H_k with $\tilde{\mathcal{O}}(rd^n)$ arithmetic operations over $\mathbb{Q}$.
Step 8 can be done with $\tilde{\mathcal{O}}(rd^n)$ arithmetic operations over $\mathbb{Q}$, see [Chè10].

Thus the global cost of the algorithm belongs to $\tilde{\mathcal{O}}(rd^{n+\omega-1})$ arithmetic operations over $\mathbb{Q}$. $\square$

4. Examples

In this section we show the behavior of the algorithm Recombination for Decomposition with two examples. We consider bivariate rational functions with rational coefficients. Thus hypothesis (C) is satisfied.

4.1. f is non-composite.

We set:

$$\begin{aligned} f_1 &= \left(1+X+Y^2\right)(X+Y) = X+X^2+XY^2+Y+YX+Y^3, \\ f_2 &= f_1 - \left(Y^2-X-1\right)(Y-2X+1) = -X^2+3XY^2+2Y+2YX-Y^2+1 \end{aligned}$$

We have $\deg(f_1+\Lambda f_2) = \deg_Y(f_1+\Lambda f_2) = 3$, and

$$Res_Y\Big(f_1(0,Y)+\Lambda f_2(0,Y),\, \partial_Y f_1(0,Y)+\Lambda\partial_Y f_2(0,Y)\Big) = -4-24\,\Lambda-92\,\Lambda^2-64\,\Lambda^3+8\,\Lambda^4.$$

Thus hypothesis (H) is satisfied.
The algorithm Good homography gives:
$\lambda_a = f_1(0,0)/f_2(0,0) = 0$ and $\lambda_b = f_1(0,1)/f_2(0,1) = 1$.

Then

$$\begin{aligned} F_1 &= \left(1+X+Y^2\right)(X+Y), \\ F_{1,1} &= 1+X+Y^2, \\ F_{1,2} &= X+Y \\ F_2 &= \left(Y^2-X-1\right)(Y-2X+1), \\ F_{2,1} &= Y^2-X-1 \\ F_{2,2} &= Y-2X+1 \end{aligned}$$

The cofactors are:

$$\begin{aligned} \mathcal{G}_{F_{1,1}} &= 3X^2+8YX^2+2X-2YX+7XY^2-1+3Y^2-6Y^3-6Y^4+2Y \\ \mathcal{G}_{F_{1,2}} &= 3X^2+8YX^2+4YX+6X-6Y^2-4Y+3-3Y^4-2Y^3 \\ \mathcal{G}_{F_{2,1}} &= 3X^2+8YX^2+XY^2-6YX+2X-1-2Y-6Y^4-11Y^2-6Y^3 \\ \mathcal{G}_{F_{2,2}} &= 3X^2+8YX^2+6XY^2+8YX+6X-3Y^4+8Y^2-2Y^3+3 \end{aligned}$$

The linear system $\mathcal{S}$ is the following:

$$\begin{bmatrix} -1 & 3 & -1 & 3 \\ 2 & 6 & 2 & 6 \\ 3 & 3 & 3 & 3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & -4 & -2 & 0 \\ -2 & 4 & -6 & 8 \\ 8 & 8 & 8 & 8 \\ 0 & 0 & 0 & 0 \\ 3 & -6 & -11 & 8 \\ 7 & 0 & 1 & 6 \\ 0 & 0 & 0 & 0 \\ -6 & -2 & -6 & -2 \\ 0 & 0 & 0 & 0 \\ -6 & -3 & -6 & -3 \end{bmatrix}$$

A basis of $\ker(\mathcal{S})$ is given by: $\{(-1,-1,1,1)\}$.
Then it follows that f_1/f_2 is non-composite.

4.2. f **is composite.** Here we set:

$$\begin{aligned} h_1 &= \left(1+X+Y^2\right)(X+Y) \\ h_2 &= h_1 - \left(Y^2 - X - 1\right)(Y - 2\,X + 1) \\ u_1 &= T.(T-1) \\ u_2 &= T^2 + 1 \\ f_1/f_2 &= u_1/u_2(h_1/h_2). \end{aligned}$$

We have constructed a composite rational function f_1/f_2 and now we illustrate how our algorithm computes a decomposition. We can already remark that in the previous example we have shown that h_1/h_2 is non-composite.

In this situation the hypothesis (H) is satisfied and the algorithm Good Homography gives:
$\lambda_a = f_1(0,0)/f_2(0,0) = 0$ and $\lambda_b = f_1(0,2)/f_2(0,2) = 90/101$.
Then:

$$\begin{aligned} F_{1,1} &= 1 + X + Y^2 \\ F_{1,2} &= 2X - Y - 1 \\ F_{1,3} &= Y^2 - X - 1 \\ F_{1,4} &= X + Y \\ F_{2,1} &= 2\,X^2 + 11\,X + 9 + 29\,XY + 29\,Y + 38\,XY^2 - 9\,Y^2 + 11\,Y^3 \\ F_{2,2} &= 11\,X^2 + X - 10 - 19\,YX - 19\,Y - 29\,XY^2 + 10\,Y^2 + Y^3 \end{aligned}$$

The basis in reduced row echelon form of $\pi_1(\ker \mathcal{S})$ (resp. $\pi_2(\ker \mathcal{S})$) is $\{(1,0,0,1);(0,1,1,0)\}$ (resp. $\{(1,0);(0,1)\}$).
Step 5 in the algorithm Recombination for Decomposition gives: $v_1 = (1,0,0,1)$ and $v_2 = (1,0)$.
Then we have $H_1 := F_{1,1}.F_{1,4}$ and $H_2 := F_{2,1}$.
We remark that $H_1 = h_1$ and that $H_2 = 11h_1 + 9h_2$. Then $H_1/H_2 = w(h_1/h_2)$, where $w(T) = T/(11T+9)$. As h_1/h_2 is non-composite and $\deg w = 1$, we get a correct output.

5. Conclusion

In conclusion, we summarize our algorithm with a "derivation point of view".
In order to decompose f_1/f_2, we have computed with Darboux method a rational first integral of D_{f_1/f_2} with minimum degree. That is to say we have computed $h_1/h_2 \in \mathbb{K}(X_1,\dots,X_n)$ such that $D_{f_1/f_2}(h_1/h_2) = 0$ and $\deg(h_1/h_2)$ is minimum.
In a general setting, Darboux method works as follows: If we want to compute a rational first integral of a derivation D, first we compute all the Darboux polynomials F_i and their associated cofactors $\mathcal{G}_{F_i}$, second we solve the linear system

$$\sum_i e_i \mathcal{G}_{F_i} = 0.$$

Then thanks to Proposition 8, we deduce that $\prod_i F_i^{e_i}$ is a first integral, i.e. $D(\prod_i F_i^{e_i}) = 0$.

When we consider the derivation D_{f_1/f_2} the computation of Darboux polynomials is reduced to the factorization of $f_1 + \lambda f_2$. Thus this step can be done efficiently. In the general setting, we can also reduce the computation of Darboux polynomials to a factorization problem, see [Chè11].
During the second step, we compute the kernel of $\sum_i e_i \mathcal{G}_{F_i} = 0$. It is actually a recombination step. Indeed, this system explains how we have to recombine F_i in order to get a rational first integral. Furthermore, the cofactor $\mathcal{G}_{F_i} = D(F_i)/F_i$ can be viewed as a logarithmic derivative.
In conclusion, the recombination scheme used in this paper is called nowadays the logarithmic derivative method, but this method is Darboux original method.

Appendix A. Convex-dense bivariate decomposition

In this appendix we give complexity results for the decomposition of sparse bivariate rational functions. These results rely on a strategy proposed by J. Berthomieu and G. Lecerf in [BL10].

Given a polynomial $f(X,Y) \in \mathbb{K}[X,Y]$, its support is the set S_f of integer points $(i;j)$ such that the monomial X^iY^j appears in f with a non zero coefficient. The convex hull, in the real space $\mathbb{R}^2$ of S_f is denoted by $N(f)$ and called the Newton's polygon of f. We denote by $|N(f)|$ the number of integral points of $N(f)$. We called $|N(f)|$ the convex-size of f.

Roughly speaking, the transformation proposed in [BL10] consists in a monomial transformation that preserves the convex-size but decreases the dense size. The considered transformation $\mathcal{T}$ can be described in the following way:

$$\mathcal{T} = \mathcal{B} \circ \mathcal{L}, \text{ where}$$

$$\mathcal{B}(X^iY^j) = X^{i+b_1}Y^{j+b_2},\ b_1, b_2 \in \mathbb{Z},$$
$$\mathcal{L}(X^iY^j) = X^{a_1i+a_2j}Y^{a_3i+a_4j},\ a_1a_4 - a_2a_3 = \pm 1.$$

$\mathcal{T}$ can be defined on $\mathbb{K}[X, Y, X^{-1}, Y^{-1}]$, and we define: $\mathcal{T}(\sum_{i,j} f_{i,j}X^iY^j) = \sum_{i,j} f_{i,j}\mathcal{T}(X^iY^j)$.
The transformation $\mathcal{L}$ corresponds to the linear map: $(i, j) \mapsto \mathcal{A}^t(i, j)$, where

$$\mathcal{A} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}.$$

We denote by $\mathcal{L}^{-1}$ the transformation corresponding to $\mathcal{A}^{-1}$.
If $f(X, Y) \in \mathbb{K}[X, Y]$, then $\mathcal{L}(f) \in \mathbb{K}[X, Y, X^{-1}, Y^{-1}]$ and $\mathcal{L}(f)$ can be written $\mathcal{L}(f) = c_{\mathcal{L}}(f).\mathcal{L}_0(f)$, where $\mathcal{L}_0(f) \in \mathbb{K}[X, Y]$ and $c_{\mathcal{L}}(f) = X^iY^j \in \mathbb{K}[X, Y, X^{-1}, Y^{-1}]$. Furthermore, we also have $\mathcal{L}(F_1.F_2) = \mathcal{L}(F_1).\mathcal{L}(F_2)$.

Let S be a finite subset of $\mathbb{Z}^2$. Set S is said to be normalized if it belongs to $\mathbb{N}^2$ and if it contains at least one point in $\{0\} \times \mathbb{N}$, and also at least one point in $\mathbb{N} \times \{0\}$. For such a normalized set, we write d_x (resp. d_y) for the largest abscissa (resp. ordinate) involved in S, so that the bounding rectangle is $\mathcal{R} = [0, d_x] \times [0, d_y]$. The following result is proved in [BL10, Theorem 2]:
For any normalized finite subset S of $\mathbb{Z}^2$, of cardinality σ, convex-size π, and bounding rectangle $[0, d_x] \times [0, d_y]$, and dense size $\delta = (d_x+1)(d_y+1)$, one can compute an affine map $\mathcal{T} = \mathcal{B} \circ \mathcal{L}$, with $\mathcal{O}(\sigma \log^2 \delta)$ bit-operations, such that $\mathcal{T}(S)$ is normalized of dense size at most 9π.

We are going to use this transformation in order to prove:

Theorem 19. *Let $f_1/f_2(X, Y) \in \mathbb{K}(X, Y)$ such that $\deg(f_1/f_2) = d$, $N(f_1) \subset N$, $N(f_2) \subset N$ and N is normalized. Then*

(1) *If $\mathbb{K}$ is field with characteristic 0 or at least $d(d-1)+1$ and (H) is satisfied, then there exists a probabilistic algorithm which computes the decomposition of f_1/f_2 with at most $\tilde{\mathcal{O}}(|N|^{1,5})$ operations in $\mathbb{K}$ and two factorizations of a univariate polynomial of degree at most $9|N|$ over $\mathbb{K}$.*
(2) *If (C) and (H) are satisfied, then there exists a deterministic algorithm which computes the decomposition of f_1/f_2 with at most $\tilde{\mathcal{O}}(r.|N|^{(\omega+1)/2})$ operations over $\mathbb{Q}$ and two factorizations of an univariate polynomials of degree at most $9|N|$ over $\mathbb{Q}[\alpha]$.*

Now, we explain how we use the transformation $\mathcal{T}$ in the decomposition setting.

Proposition 20. *If $f_1/f_2 = u(h_1/h_2)$ then $\mathcal{T}(f_1)/\mathcal{T}(f_2) = u\big(\mathcal{L}(h_1)/\mathcal{L}(h_2)\big)$.*
If $\mathcal{T}(f_1)/\mathcal{T}(f_2) = u(H_1/H_2)$ then $f_1/f_2 = u\big(\mathcal{L}^{-1}(H_1)/\mathcal{L}^{-1}(H_2)\big)$.

Proof. We prove the first item, the second can be proved in a similar way.
We have: $\dfrac{f_1}{f_2} = \dfrac{\prod_i (h_1 - \mu_{1,i}h_2)}{\prod_j (h_1 - \mu_{2,j}h_2)}$, where $\mu_{k,i}$ are roots of u_k. Then,

$$\begin{aligned}
\frac{\mathcal{T}(f_1)}{\mathcal{T}(f_2)} &= \frac{\mathcal{B} \circ \mathcal{L}(f_1)}{\mathcal{B} \circ \mathcal{L}(f_2)} = \frac{X^{b_1}Y^{b_2}\mathcal{L}(f_1)}{X^{b_1}Y^{b_2}\mathcal{L}(f_2)} \\
&= \frac{\mathcal{L}(f_1)}{\mathcal{L}(f_2)} = \frac{\prod_i \big(\mathcal{L}(h_1) - \mu_{1,i}\mathcal{L}(h_2)\big)}{\prod_j \big(\mathcal{L}(h_1) - \mu_{2,j}\mathcal{L}(h_2)\big)} \\
&= \frac{u_1\big(\mathcal{L}(h_1)/\mathcal{L}(h_2)\big)}{u_2\big(\mathcal{L}(h_1)/\mathcal{L}(h_2)\big)} = u\big(\mathcal{L}(h_1)/\mathcal{L}(h_2)\big)
\end{aligned}$$

□

This gives the following algorithm:

Convex bivariate decomposition
Input: $f = f_1/f_2 \in \mathbb{K}(X,Y)$, where $N(f_1) \subset N$, $N(f_2) \subset N$ and N is normalized.
Output: A decomposition of f if it exists, with $f = u \circ h$, $u = u_1/u_2$ with $\deg u \geq 2$, and $h = h_1/h_2$ non-composite.

(1) Compute $F = \mathcal{T}(f_1)/\mathcal{T}(f_2)$.
(2) Decompose $F = u(H)$.
(3) Return $f = u(h)$, where $h = \dfrac{\mathcal{L}^{-1}(H_1)}{\mathcal{L}^{-1}(H_2)} = \dfrac{c_{\mathcal{L}^{-1}}(H_1).\mathcal{L}_0^{-1}(H_1)}{c_{\mathcal{L}^{-1}}(H_2).\mathcal{L}_0^{-1}(H_2)} \in \mathbb{K}(X,Y)$.

Proposition 21. *The algorithm* Convex bivariate decomposition *is correct.*

Proof. This follows from Proposition 20. □

Proposition 22. *The algorithm* Convex bivariate decomposition *uses one decomposition of a rational function of degree at most* $9|N|$ *and* $\mathcal{O}(\sigma^2\delta)$ *bit operations.*

Proof. We apply [BL10, Theorem 2] to N. □

The proof of Theorem 19 comes from Proposition 21 and Proposition 22 and complexity results given in Theorem 1 and [Chè10, Theorem 2].

References

[AGR95] C. Alonso, J. Gutierrez, and T. Recio. A rational function decomposition algorithm by near-separated polynomials. *J. Symbolic Comput.*, 19(6):527–544, 1995.

[AT85] V. S. Alagar and Mai Thanh. Fast polynomial decomposition algorithms. In *EUROCAL '85, Vol. 2 (Linz, 1985)*, volume 204 of *Lecture Notes in Comput. Sci.*, pages 150–153. Springer, Berlin, 1985.

[BCN] L. Busé, G. Chèze, and S. Najib. Noether's forms for the study of non-composite rational functions and their spectrum. *A. Arithmetica*, to appear.

[BDN09] A. Bodin, P. Dèbes, and S. Najib. Indecomposable polynomials and their spectrum. *A. Arithmetica*, 139(1):79–100, 2009.

[BHKS09] K. Belabas, M. van Hoeij, J. Klüners, and A. Steel. Factoring polynomials over global fields. *J. Theorie des Nombres de Bordeaux*, 21:15–39, 2009.

[BL10] J. Berthomieu and G. Lecerf. Convex-dense bivariate polynomial factorization. Manuscript, 2010.

[BLS+04] A. Bostan, G. Lecerf, B. Salvy, É. Schost, and B. Wiebelt. Complexity Issues in Bivariate Polynomial Factorization. In *Proceedings of ISSAC 2004*, pages 42–49. ACM, 2004.

[BP94] D. Bini and V.Y. Pan. *Polynomial and matrix computations. Vol. 1.* Progress in Theoretical Computer Science. Birkhäuser Boston Inc., Boston, MA, 1994. Fundamental algorithms.

[BZ85] D.R. Barton and R. Zippel. Polynomial decomposition algorithms. *J. Symbolic Comput.*, 1(2):159–168, 1985.

[Chè10] G. Chèze. Nearly optimal algorithms for the decomposition of multivariate rational functions and the extended Luroth's theorem. *J. Complexity*, 26(4):344–363, 2010.

[Chè11] G. Chèze. Computation of Darboux polynomials and rational first integrals with bounded degree in polynomial time. *J. Complexity*, to appear 2011.

[CL07] G. Chèze and G. Lecerf. Lifting and recombination techniques for absolute factorization. *J. Complexity*, 23(3):380–420, 2007.

[CN10] G. Chèze and S. Najib. Indecomposability of polynomials via jacobian matrix. *J. Algebra*, 324(1):1–11, 2010.

[Dic87] M. Dickerson. Polynomial decomposition algorithms for multivariate polynomials. Technical Report TR87-826, Comput. Sci., Cornell Univ., 1987.

[DLA06] F. Dumortier, J. Llibre, and J.C. Artés. *Qualitative theory of planar differential systems*. Universitext. Springer-Verlag, Berlin, 2006.

[FGP10] J.-C. Faugère, J. von zur Gathen, and L. Perret. Decomposition of generic multivariate polynomials. In *Proceedings of ISSAC 2010*, pages 131–137. ACM, 2010.

[FP09] J.-C. Faugère and L. Perret. An efficient algorithm for decomposing multivariate polynomials and its applications to cryptography. *J. Symbolic Comput.*, 44(12):1676–1689, 2009.

[Gat90a] J. von zur Gathen. Functional decomposition of polynomials: the tame case. *J. Symbolic Comput.*, 9(3):281–299, 1990.

[Gat90b] J. von zur Gathen. Functional decomposition of polynomials: the wild case. *J. Symbolic Comput.*, 10(5):437–452, 1990.

[Gat08] J. von zur Gathen. Counting decomposable multivariate polynomials. arXiv:0811.4726v2, 2008.

[GG03] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, Cambridge, second edition, 2003.

[GGR03] J. von zur Gathen, J. Gutierrez, and R. Rubio. Multivariate polynomial decomposition. *Appl. Algebra Engrg. Comm. Comput.*, 14(1):11–31, 2003.

[Gie88] M. Giesbrecht. Some results on the functional decomposition of polynomials. Master Thesis, University of Toronto, arXiv:1004.5433, 1988.

[GRS01] J. Gutierrez, R. Rubio, and D. Sevilla. Unirational fields of transcendence degree one and functional decomposition. In *ISSAC '01: Proceedings of the 2001 international symposium on Symbolic and algebraic computation*, pages 167–174, New York, NY, USA, 2001. ACM Press.

[GW95] J. von zur Gathen and J. Weiss. Homogeneous bivariate decompositions. *J. Symbolic Comput.*, 19(5):409–434, 1995.

[KL89] D. Kozen and S. Landau. Polynomial decomposition algorithms. *J. Symbolic Comput.*, 7(5):445–456, 1989.

[Klü99] J. Klüners. On polynomial decompositions. *J. Symbolic Comput.*, 27(3):261–269, 1999.

[Lec06] G. Lecerf. Sharp precision in Hensel lifting for bivariate polynomial factorization. *Math. Comp.*, 75(254):921–933 (electronic), 2006.

[Lec07] G. Lecerf. Improved dense multivariate polynomial factorization algorithms. *J. Symbolic Comput.*, 42(4):477–494, 2007.

[MO04] J. Moulin Ollagnier. Algebraic closure of a rational function. *Qual. Theory Dyn. Syst.*, 5(2):285–300, 2004.

[PI07] A.P. Petravchuk and O.G. Iena. On closed rational functions in several variables. *Algebra Discrete Math.*, (2):115–124, 2007.

[Rit22] J.F. Ritt. Prime and composite polynomials. *Trans. Amer. Math. Soc.*, 23(1):51–66, 1922.

[Sto00] A. Storjohann. *Algorithms for matrix canonical forms*. PhD thesis, ETH Zurich, Zurich, Switzerland, 2000.

[Wat08] S. Watt. Functional decomposition of symbolic polynomials. In *International Conference on Computatioanl Sciences and its Applications*, pages 353–362. IEEE Computer Society, 2008.

[Wat09] S. Watt. Algorithms for the functional decomposition of laurent polynomials. In *Conferences on Intelligent Computer Mathematics 2009: 16th Symposium on the Integration of Symbolic Computation and Mechanized Reasoning and 8th International Conference on Mathematical Knowledge Management , (Calculemus 2009)*, pages 186–200. Springer-Verlag LNAI 5625, 2009.

[Wei10] M. Weimann. A lifting and recombination algorithm for rational factorization of sparse polynomials. *J. Complexity*, 26(6):608–628, 2010.

[Zip91] R. Zippel. Rational function decomposition. In *Proceedings of the 1991 international symposium on Symbolic and algebraic computation*, pages 1–6. ACM Press, 1991.

Institut de Mathématiques de Toulouse, Université Paul Sabatier Toulouse 3, MIP Bât 1R3, 31 062 TOULOUSE cedex 9, France

E-mail address: guillaume.cheze@math.univ-toulouse.fr