A Superfast Randomized Algorithm to Decompose Binary Forms

Matías Bender, Jean-Charles Faugère, Ludovic Perret, Elias Tsigaridas

A Superfast Randomized Algorithm to Decompose Binary Forms

Matías R. Bender
matias.bender@inria.fr

Jean-Charles Faugère
jean-charles.faugere@inria.fr

Ludovic Perret
ludovic.perret@lip6.fr

Elias Tsigaridas
elias.tsigaridas@inria.fr

Sorbonne Universités, UPMC Univ Paris 06, CNRS, INRIA,
Laboratoire d'Informatique de Paris 6 (LIP6), Équipe POLSYS, 4 place Jussieu, 75252 Paris Cedex 05, France

ABSTRACT

Symmetric Tensor Decomposition is a major problem that arises in areas such as signal processing, statistics, data analysis and computational neuroscience. It is equivalent to write a homogeneous polynomial in n variables of degree D as a sum of D-th powers of linear forms, using the minimal number of summands. This minimal number is called the *rank* of the polynomial/tensor. We consider the decomposition of binary forms, that corresponds to the decomposition of symmetric tensors of dimension 2 and order D. This problem has its roots in Invariant Theory, where the decompositions are known as canonical forms. As part of that theory, different algorithms were proposed for the binary forms. In recent years, those algorithms were extended for the general symmetric tensor decomposition problem.

We present a new randomized algorithm that enhances the previous approaches with results from *structured linear algebra* and techniques from *linear recurrent sequences*. It achieves a *softly linear* arithmetic complexity bound. To the best of our knowledge, the previously known algorithms have quadratic complexity bounds. We compute a symbolic minimal decomposition in $O(\mathtt{M}(D)\log(D))$ arithmetic operations, where $\mathtt{M}(D)$ is the complexity of multiplying two polynomials of degree D. We approximate the terms of the decomposition with an error of $2^{-\varepsilon}$, in $O\big(D\log^2(D)\big(\log^2(D)+\log(\varepsilon)\big)\big)$ arithmetic operations. To bound the size of the representation of the coefficients involved in the decomposition, we bound the algebraic degree of the problem by $\min(\mathit{rank}, D-\mathit{rank}+1)$. When the input polynomial has integer coefficients, our algorithm performs, up to poly-logarithmic factors, $\widetilde{O}_B(D\ell + D^4 + D^3\tau)$ bit operations, where τ is the maximum bitsize of the coefficients and $2^{-\ell}$ is the relative error of the terms in the decomposition.

Keywords

Binary Form Decomposition; Superfast Algorithm; Hankel Matrix; Complexity; Algebraic Degree; Canonical Form; Waring's problem

1. INTRODUCTION

The Symmetric Tensor Decomposition problem (STD) consists in writing a symmetric tensor of dimension n and order D as a sum of rank-1 symmetric tensors, using the minimal number of summands. The length of the shortest sum is known as the rank of the symmetric tensor. A symmetric tensor is said to have rank 1 when it is, roughly speaking, the k-th outer-product of a vector. This definition of rank is also called symmetric rank [8]. Under different formulations, STD appears in many various areas like signal processing, computer vision, statistics, psychometrics, chemometrics, data mining, computational neuroscience. For an up-to-date introduction of the state-of-the-art in STD, we refer to e.g. [6, 20].

We can express STD in terms of homogeneous polynomials, e.g. [8]: given a homogeneous polynomial in n variables of degree D, we want to write it as a sum of D-th powers of linear forms, using the minimal numbers of summands. The rank of the polynomial, and equivalently of the tensor, is this minimal number.

Under this formulation, STD dates back to the origin of modern linear algebra, as it was studied in Invariant Theory. There, the decompositions are known as the canonical forms [11, 34, 35]. This problem, and the related theory of apolarity, were important for this classical theory, because they give information on the behavior of the polynomials under a linear change of variables [19]. It was also studied as a polynomial version of Waring's problem [8, 13].

Binary Form Decomposition (BFD). We consider the decomposition of symmetric tensors of order D and dimension 2. In terms of polynomials, given a binary form

$$f(x,y) := \sum_{i=0}^{D} \binom{D}{i} a_i x^i y^{D-i}, \tag{1}$$

with $a_i \in \mathbb{F} \subset \mathbb{C}$ and $\mathbb{F}$ some field of characteristic zero, we want to compute a decomposition

$$f(x,y) = \sum_{j=1}^{r} (\alpha_j x + \beta_j y)^D \tag{2}$$

where $\alpha_1,\ldots,\alpha_r,\beta_1,\ldots,\beta_r \in \overline{\mathbb{F}}$ (the algebraic closure of $\mathbb{F}$) and r is minimal. A decomposition is said to be *unique* when for all the decompositions, the set of points $\{(\alpha_j,\beta_j) : 1 \le j \le r\} \subset \mathbb{P}(\overline{\mathbb{F}})$ is unique, where $\mathbb{P}(\overline{\mathbb{F}})$ is the projective space of $\overline{\mathbb{F}}$ [32].

Previous work. The decomposition problem of Eq. (2) has been extensively studied for $\mathbb{F} = \mathbb{C}$. The classical work of Sylvester [34, 35] described the necessary and sufficient conditions for a decomposition to exist (see Sec. 2.1). He showed how the decompositions are linked to Hankel matrices. He proved that the vectors, of dimension $r+1$, in the kernel of these matrices are related to decompositions of rank r. For a modern approach of this topic, we refer to [14, 18, 19, 32]. Sylvester's work can be extended to a more general kind of polynomial decompositions [11, 14, 31].

Sylvester's ideas lead straightforwardly to Sylvester's algorithm (Alg. 1) of [7, Sec. 3.4.3]. In addition, several algorithms were proposed to decompose binary forms, but with complexity at least quadratic in the degree of the binary form. For binary forms of odd degree we can modify Berlekamp-Massey algorithm [2, 22] to compute the decompositions [9]. When the decomposition is unique, the Catalectican algorithm, which also works for symmetric tensors of bigger dimension [14, 26], improves Alg. 1. For an arbitrary binary form, Helmke [13] presented a randomized algorithm based on Padé approximates and continued fractions and described the different possible decompositions.

Besides of BFD, the subproblems of estimating the rank and deciding the uniqueness of the decompositions were considered

[3, 5, 13, 34, 35]. Sylvester considered them [34, 35] for generic binary forms; that is for binary forms with coefficients belonging to a dense algebraic open subset of $\overline{\mathbb{F}}^{D+1}$ [7, Sec. 3]. He proved that when the degree is $2k$ or $2k+1$, the rank is $k+1$ and that the minimal decomposition is unique only when the degree is odd. In the non-generic case, the rank is related with the kernel of a Hankel matrix [5, 13, 14]. Moreover, the decomposition of a binary form of degree $2k$ or $2k-1$ and rank r, is unique if and only if $r \leq k$. Different algorithms were proposed to compute only the rank [3, 5, 8]. They do not provide complexity estimates, but using recent superfast algorithms for Hankel matrices [27], we can deduce a nearly-optimal arithmetic complexity bound for [5].

For the general STD problem, Sylvester's work was successfully extended to cases where the decomposition is unique [4, 26].

Formulation of the problem. To control the algebraic degree [1, 24] of the terms in the decompositions, we consider an equivalent problem. Instead of decomposing the polynomials as in Eq. (2), we compute $\lambda_1 \dots \lambda_r$, $\alpha_1 \dots \alpha_r$, $\beta_1 \dots \beta_r \in \overline{\mathbb{F}}$, where r is minimal, such that,

$$f(x,y) = \sum_{j=1}^{r} \lambda_j (\alpha_j x + \beta_j y)^D. \tag{3}$$

Since $\lambda_j \in \overline{\mathbb{F}}$, the problems are equivalent. However, if $\lambda_j, \alpha_j, \beta_j \in \mathbb{F}$, then the task is not the same. We do not study this problem, and we refer to [8, 13, 33], for $\mathbb{F} = \mathbb{R}$, and to [31, 32], $\mathbb{F} \subset \mathbb{C}$.

Main results and organization of the paper. We extend Sylvester's algorithm to achieve a nearly-optimal complexity bound in the degree of the binary form. Exploiting the relation between the kernel of the Hankel matrices and the decompositions, we bound the possible values of the rank of the decomposition, and we identify when the decomposition is unique. We build upon [13] and we use Berlekamp-Massey algorithm to provide a better complexity bound than what is currently known. As Sylvester's algorithm does, our approach decomposes successfully any binary form.

We present a superfast randomized algorithm to compute a *symbolic decomposition* in $O(\mathtt{M}(D)\log(D))$, where $\mathtt{M}(D)$ is the arithmetic complexity of polynomial multiplication. Such a decomposition refers to a representation of the decomposition as a sum of a rational function evaluated at the roots of a univariate polynomial. Using this representation, we can approximate the terms in the decomposition, with a relative error of $2^{-\varepsilon}$, in $O\big(D\log^2(D)\big(\log^2(D)+\log(\varepsilon)\big)\big)$ arithmetic operations [23, 28]. From the symbolic decomposition, we can deduce for free the rank and the border rank of the tensor (see [5, Sec. 1]). The algorithm is randomized because it performs a random linear change of variables to fulfill some genericity assumptions. We stress that we can check if the genericity assumptions hold in $O(\mathtt{M}(D)\log(D))$.

Using results from Kaltofen and Yagati [17], we bound the algebraic degree of the decompositions by $\min(r, D-r+1)$. For polynomials with integer coefficients, we bound the bit complexity, up to poly-logarithmic factors, by $\widetilde{O}_B(D\ell + D^4 + D^3\tau)$, where τ is the maximum bitsize of the coefficients of the binary form and $2^{-\ell}$ is the error of the terms in the decomposition.

The rest of the paper is organized as follows. First we present the notation. In Sec. 2 we present the preliminaries needed by the algorithm. Sec. 3 contains the main algorithm and its proof of correctness. In Sec. 4, we study the algebraic degree of the problem and the arithmetic and bit complexity of our algorithm

Notation. We denote by O, resp. O_B, the arithmetic, resp. bit, complexity and we use $\widetilde{O}$, resp. $\widetilde{O}_B$, to ignore (poly-)logarithmic factors. $\mathtt{M}(n)$ is the arithmetic complexity of multiplying two polynomial of degree n. Let $\mathbb{F}$ be a subfield of $\mathbb{C}$, and $\overline{\mathbb{F}}$ its algebraic closure. If $v = (v_0, \dots, v_n)^\mathsf{T}$ then $P_v = P_{(v_0,\dots,v_n)} := \sum_{i=0}^{n} v_i x^i y^{n-i}$.

Algorithm 1 INCRDECOMP [7, Fig. 1]

1. $r := 1$
2. Get a random $c \in \mathtt{Ker}(H^r)$
3. If P_c is not square-free, $r := r+1$ and GO TO 2
4. Write P_c as $\prod_{j=1}^{r}(\beta_j x - \alpha_j y)$
5. Solve the Vandermonde transposed system:
$$\begin{pmatrix} \beta_1^D & \cdots & \beta_r^D \\ \beta_1^{D-1}\alpha_1 & \cdots & \beta_r^{D-1}\alpha_r \\ \vdots & \ddots & \vdots \\ \alpha_1^D & \cdots & \alpha_r^D \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_r \end{pmatrix} = \begin{pmatrix} a_0 \\ \vdots \\ a_D \end{pmatrix} \tag{5}$$
6. Return $\sum_{j=1}^{r} \lambda_j (\alpha_j x + \beta_j y)^D$

Given a binary form $f(x,y)$, we refer as $f(x)$ to the univariate polynomial $f(x) := f(x,1)$. By $f'(x)$ we denote the derivative with respect to x. For a matrix M, we refer as $\mathtt{rk}(M)$ and $\mathtt{Ker}(M)$ to the rank and the kernel of M.

2. PRELIMINARIES

2.1 An algorithm based on Sylvester's theorem

Sylvester's theorem (Thm. 2.2) relates the minimal decomposition of a binary form to the kernel of a Hankel matrix. Moreover, it implies an (incremental) algorithm for computing the minimal decomposition. The version that we present in Alg. 1 comes from Common and Mourrain [7, Sec. 3.2].

Definition 2.1 *Given a vector $a = (a_0, \dots, a_D)^\mathsf{T}$, we denote by $\{H_a^k\}_{1 \leq k \leq D}$ the family of Hankel matrices indexed by k, where $H_a^k \in \mathbb{F}^{(D-k+1)\times(k+1)}$ and*

$$H_a^k := \begin{pmatrix} a_0 & a_1 & \cdots & a_{k-1} & a_k \\ a_1 & a_2 & \cdots & a_k & a_{k+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{D-k-1} & a_{D-k} & \cdots & a_{D-2} & a_{D-1} \\ a_{D-k} & a_{D-k+1} & \cdots & a_{D-1} & a_D \end{pmatrix}. \tag{4}$$

We may omit the a in H_a^k when it is clear from the context.

Theorem 2.2 (Sylvester, 1851) *Let $f = \sum_{i=0}^{D} \binom{D}{i} a_i x^i y^{D-i}$ with $a_i \in \mathbb{F} \subseteq \mathbb{C}$. Also, consider $c = (c_0, \dots, c_r)^\mathsf{T} \in \mathbb{F}^{r+1}$ non-zero, such that the polynomial*

$$P_c = \sum_{i=0}^{r} c_i x^i y^{r-i} = \prod_{j=1}^{r} (\beta_j x - \alpha_j y)$$

is square-free and $\alpha_i, \beta_i \in \overline{\mathbb{F}}$. Then, there are $\lambda_1, \dots \lambda_r \in \overline{\mathbb{F}}$ such that $f(x,y) = \sum_{j=1}^{r} \lambda_j (\alpha_j x + \beta_j y)^D$, iff $(c_0, \dots, c_r)^\mathsf{T} \in \mathtt{Ker}(H_a^r)$.

For a proof of Thm. 2.2 we refer to the work of Reznick [32, Thm. 2.1 & Cor. 2.2]. His proof is of interest to us because it relates the decomposition with linear recurrent sequences (Sec. 2.3).

Thm. 2.2 implies Alg. 1. This algorithm will loop as many times as the rank. In the i-th loop it will compute the kernel of H^i. The dimension of this kernel is $\leq i$, and each vector in the kernel has $i+1$ coordinates. As the rank of the binary form can be as big as the degree of the binary form, a straightforward bound for the arithmetic complexity of Alg. 1 is at least cubic in the degree.

We can improve the complexity of Alg. 1 by a factor of D by noticing that the rank of the binary form is either $\mathtt{rk}(H^{\lceil \frac{D}{2} \rceil})$ or $D - \mathtt{rk}(H^{\lceil \frac{D}{2} \rceil}) + 2$ [5, Sec. 3][13, Thm. B]. Another way to compute the rank is by using the minors [3, Alg. 2].

The bottleneck of the previous approaches is that they have to compute the kernel of a Hankel matrix. However, even if we know that the rank of the binary form is r, the dimension of the kernel of

H^r, as well as the dimension of the vectors in the kernel, can be as big as $O(D)$. Hence, the complexity is lower bounded by $O(D^2)$.

Our approach avoids the incremental construction. We exploit the structure of the kernel of the Hankel matrices and we prove that the rank has just two possible values (Lem. 3.1). Moreover, we use a compact representation of the vectors in the kernel. We describe them as a combination of two polynomials of degree $O(D)$.

2.2 Kernel of the Hankel matrices

The Hankel matrices are one of the most well known structured matrices [27]. We present results about the structure of their kernel. For details, we refer to Heinig and Rost [12, Ch. 5].

Consider a family of Hankel matrices $\{H_a^k\}_{1\le k\le D}$ (Def. 2.1). There is a formula for the dimension of the kernel of each matrix in the family, involving two numbers, N_1^a and N_2^a.

Proposition 2.3 *For a family of Hankel matrices $\{H_a^k\}_{1\le k\le D}$ there are two constants, N_1^a and N_2^a, such that*

1. $0 \le N_1^a \le N_2^a \le D$,
2. $(\forall k : 1 \le k \le D)$
   $\mathtt{dim}(\mathtt{Ker}(H_a^k)) = \max(0; k-N_1^a) + \max(0; k-N_2^a)$, *and*
3. $N_1^a + N_2^a = D$.

We may skip a in N_1^a, N_2^a when it is clear from the context.

Fig. 1 illustrates Prop. 2.3. The dimension of the kernels is a piecewise-linear and increasing function. If $N_1 = N_2$, then the curve degenerates to two line segments, one of slope 0, from 1 to N_1, and one of slope 2, from N_1 to D.

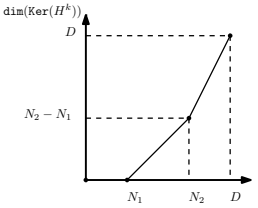

Figure 1: Dimension of the kernel of H^k

The elements of the kernel of the matrices in $\{H^k\}$ are related. To express this relation from a linear algebra point of view, we to introduce the **U-chains**.

Definition 2.4 ([12, Def. 5.1]) *A **U-chain** of length k of a vector $v = (v_0, \dots, v_n)^\mathsf{T} \in \mathbb{F}^{n+1}$ is a set of vectors $\{U_k^0 v, U_k^1 v, \dots, U_k^{k-1} v\} \subset \mathbb{F}^{n+k}$.*

The i-th element, $0 \le i \le k-1$, is

$$U_k^i v = (\underbrace{0 \dots 0}_{i}, \underbrace{v_0 \dots v_n}_{n+1}, \underbrace{0 \dots 0}_{k-1-i})$$

where U_k^i is a $(n+k)\times(n+1)$ i-shifting matrix [12, Page 11].

If v is not zero, then all the elements in a U-chain of v are linearly independent. The following theorem uses the U-chains to relate the vectors of the kernels in a family of Hankel matrices.

Proposition 2.5 (Vectors v and w) *Given a family of Hankel matrices $\{H^k\}_{1\le k\le D}$, let N_1 and N_2 be the constants of Prop. 2.3. There are two vectors, $v \in \mathbb{F}^{N_1+1}$ and $w \in \mathbb{F}^{N_2+1}$, such that,*

- *If $0 \le k \le N_1$, then $\mathtt{Ker}(H^k) = \{0\}$.*
- *If $N_1 < k \le N_2$, then the U-chain of v of length $(k-N_1)$ is a basis of $\mathtt{Ker}(H^k)$, that is*
$$\mathtt{Ker}(H^k) = \langle U_{k-N_1}^0 v, \dots, U_{k-N_1}^{k-N_1-1} v \rangle.$$
- *If $N_2 < k \le D$, the U-chain of v of length $k-N_1$ together with the U-chain of w of length $k-N_2$ is a basis of $\mathtt{Ker}(H^k)$, that is*
$$\mathtt{Ker}(H^k) = \langle U_{k-N_1}^0 v \dots U_{k-N_1}^{k-N_1-1} v, U_{k-N_2}^0 w \dots U_{k-N_2}^{k-N_2-1} w \rangle.$$

The vectors v and w of Prop. 2.5 are not unique. Vector v could be any vector in $\mathtt{Ker}(H^{N_1+1})$. Vector w could be any vector in $\mathtt{Ker}(H^{N_2+1})$ that does not belong to the vector space generated by the U-chain of v of length N_2-N_1+1. From now on, given a family of Hankel matrices, we refer to v and w as the vectors of Prop. 2.5.

Let u be a vector in the kernel of H^k and P_u the corresponding polynomial (see Notation). We call P_u a **kernel polynomial**.

As $P_{U_k^j v} = x^j y^{k-1-j} P_v$, we write any kernel polynomial of a family of Hankel matrices as a combination of P_v and P_w [12, Prop. 5.1 & 5.5]. Moreover, P_v and P_w are relative prime.

Proposition 2.6 *Consider the family of Hankel matrices $\{H^k\}_{1\le k\le D}$. The kernel polynomials P_v and P_w are relative prime. The set of kernel polynomials of the matrix H^k are*

- *If $0 < k \le N_1$, then it is $\{0\}$.*
- *If $N_1 < k \le N_2$, then $\{P_\mu \cdot P_v : \mu \in \mathbb{F}^{k-N_1}\}$.*
- *If $N_2 < k \le D$, $\{P_\mu \cdot P_v + P_\rho \cdot P_w : \mu \in \mathbb{F}^{k-N_1}, \rho \in \mathbb{F}^{k-N_2}\}$.*

2.3 Linear recurrent sequences

Linear recurrent sequences are related to kernels of Hankel matrices. Essentially, their generating sequences generate these kernels.

Definition 2.7 *A sequence S (finite or not) is **linearly recurrent** when there is a finite sequence $(u_0, \dots, u_n)$, known as the **generating sequence**, such that: $S_{n+1+i} = \sum_{k=0}^{n} u_k \cdot S_{i+k}, (0 \le i)$.*

*A **minimal generating sequence (m.g.s.)** of S is a generating sequence whose length is the shortest with respect to the length of all the generating sequences of S.*

Berlekamp-Massey algorithm computes a minimal generating sequence [2, 22]. Moreover, Massey [22, Thm. 3] showed that a m.g.s. is unique, if and only if, its length is at most half of the length of the recurrent sequence. He also proposed to use rational function reconstruction to compute the m.g.s., when it is unique.

When the m.g.s. is unique, we compute it in softly-linear time [10, Sec. 12.3]. When it is not unique, to the best of our knowledge, the fastest deterministic algorithm to compute it has quadratic complexity , with well-known constants [25].

The Hankel matrices are related to the Berlekamp-Massey algorithm [15, 16]. In particular, we have Prop. 2.8.

Proposition 2.8 *The sequence $(u_0, \dots, u_k)$ is a generating sequence of $a = (a_0, \dots, a_D)$ iff $(u_0, \dots, u_k, -1)^\mathsf{T} \in \mathtt{Ker}(H_a^{k+1})$.*

2.4 Linear change of variables

Instead of working with the input binary form, we perform a linear change of variables to ensure that we only compute unique m.g.s. A linear change of variables L_M, in our case, is associated to an invertible matrix $M := \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$, where

$$L_M(f)(x,y) := f(M \cdot \tbinom{x}{y}) = f\big((ax+by), (cx+dy)\big).$$

This transformation preserves the rank of the binary form.

Lemma 2.9 *Let M be a non-singular 2×2 matrix. Given f and a minimal decomposition $f(x,y) = \sum_{j=1}^{r} \lambda_j (\alpha_j x + \beta_j y)^D$, a minimal decomposition for $L_M(f)$ is $\sum_{j=1}^{r} \lambda_j ((\alpha_j, \beta_j) \cdot M \cdot \binom{x}{y})^D$.*

We compute any linear change of variables in $O(\mathtt{M}(D) \cdot \log(D))$ using multi-point evaluation and interpolation algorithms for univariate polynomials [10, Ch. 10].

For a 2×2 non-singular matrix, it holds $f = L_{M^{-1}}(L_M(f))$.

We want to perform a linear change of variables so that the matrices of the transformed polynomial have generic rank profile.

Definition 2.10 *A matrix has **Generic Rank Profile** if its leading principal minors of dimension less or equal to its rank are not zero.*

Proposition 2.11 ([21, Cor. 2.9]) *Let $M(t) := \left(\begin{smallmatrix} 1 & t \\ 0 & 1 \end{smallmatrix}\right)$. Given the binary form f associated to the family of Hankel matrices $\{H_a^k\}_k$, there are at most $\sum_{i=1}^{N_1+1} i(D-2i)$ values for t such that the Hankel matrices related to $L_{M(t)}(f)$ do not have generic rank profile.*

3. THE ALGORITHM

One of the drawbacks of Alg. 1, and its variants, is that they rely on the computation of the kernels of many Hankel matrices, and they ignore the particular structure that is present. Using Lem. 3.1, we can skip many calculations by computing only two vectors, v and w (Prop. 2.5). This is the main idea behind Alg. 2 that leads to a softly-linear arithmetic complexity (Sec. 4.2).

Alg. 2 performs as follows: First, (step 1) it performs a linear change of variables to ensure that the Hankel matrices have generic rank profile (see Sec. 2.4). Next, in step 2 computes two vectors, which are intended to be v and w from Prop. 2.5, to obtain the kernel of the Hankel matrices (see Sec. 3.1). Step 3 checks if the vectors computed are in fact the v and w of Prop. 2.5 (see Lem. 3.7). If they are not, this means that the linear change of variables was not appropriate. The algorithm goes back to the first step to perform a new linear change of variables. If the vectors are the correct ones, step 4 computes a square-free kernel polynomial of the minimum degree r (see Sec. 3.2). Next, step 5 computes the coefficients $\lambda_1,\ldots,\lambda_r$ (see Sec. 4.1.2). Finally, step 6 recovers a decomposition for the original binary form.

Let f be a binary form as in Eq. (1) and let $\{H^k\}_{1\le k\le D}$ be its corresponding family of Hankel matrices (see Def. 2.1).

The next lemma establishes the rank of f.

Lemma 3.1 *Assume f, $\{H^k\}_k$, N_1 and N_2 of Prop. 2.3, and v and w of Prop. 2.5. If P_v (Prop. 2.6) is square-free then the rank of f is N_1+1, else, it is N_2+1.*

PROOF. By Prop. 2.3, for $k < N_1+1$, the kernel of H^k is trivial. Hence, by Sylvester's theorem (Thm. 2.2), there is no decomposition with a rank smaller than N_1+1. Recall that $v \in \mathtt{Ker}(H^{N_1+1})$. So, if P_v is square-free, by Sylvester's theorem, there is a decomposition of rank N_1+1.

Assume P_v is not square-free. For $N_1+1 \le k \le N_2$, P_v divides all the kernel polynomials of the matrices H^k (Prop. 2.6). Therefore, none of them is square-free, and so the rank is at least N_2+1. By Prop. 2.6, P_v and P_w do not share a root. So, there is a polynomial P_μ of degree N_2-N_1 such that $Q_\mu := P_v\cdot P_\mu + P_w$ is square-free. A formal proof of this appears in Thm. 3.8. By Prop. 2.6, Q_μ is a square-free kernel polynomial of degree N_2+1. Consequently, by Sylvester's theorem, there is a decomposition with rank N_2+1. □

Link with the classical theory. To relate Lem. 3.1 with the theory of BFD, we recall that the decompositions are identified with the square-free polynomials in the annihilator of f [19][14, Chp. 1]. All the kernel polynomials of $\{H_k\}_k$ belong to the annihilator of f, which is an ideal. If f is a binary form of degree $D=2k$ or $2k+1$, then this ideal is generated by two binary forms of degrees $\mathtt{rk}(H^k)$ and $D+2-\mathtt{rk}(H^k)$, with no common zeros [14, Thm. 1.44]. These are the polynomials P_v and P_w. Using this interpretation, Alg. 1, and its variants, computes a (redundant) generating set of the annihilator, while Alg. 2 computes a basis.

3.1 Computing the vectors v and w

We use Prop. 2.8 to compute the vectors v and w as minimal generating sequences (m.g.s.) in Alg. 3.

There are different variants of the Berlekamp-Massey algorithm to compute m.g.s. (see Sec. 2.3). The algorithm with no preconditions on the uniqueness of the m.g.s., as Berlekamp's iterative algorithm [22, Sec. 3], returns two polynomials which are, essentially, P_v and P_w of our Prop. 2.6. It is not known to us if there is a variant with softly-linear complexity in this case. Therefore, we propose a randomized alternative that uses the variant of the Berlekamp-Massey algorithm for sequences with a unique m.g.s.; The arithmetic complexity of such variant is softly-linear [10, Sec. 12.3].

Algorithm 2 FASTDECOMP

Input: A binary form $\hat{f}(x,y)$ of degree D
Output: A decomposition for $\hat{f}(x,y)$ of rank r.

1. **Perform a random linear change of variables**
   We choose a 2×2 random matrix M, and we take
   $$f(x,y) := L_M(\hat{f})(x,y) = \textstyle\sum_{i=0}^{D}\binom{D}{i}a_i x^i y^{D-i}$$
2. **Compute v and w of $\{\mathbf{H_a^k}\}_\mathbf{k}$**
   We use Alg. 3 with $a := (a_0,\ldots,a_D)^\mathsf{T}$.
3. **Test if Step 2 computed v and w**. If not, GO TO Step 1.
   Check if the preconditions of Alg. 3 hold (Lem. 3.7).
4. **IF $P_v(x,y)$ is square-free**, $Q \longleftarrow P_v$
   ELSE Compute a square-free binary form Q
   We compute a vector μ of length (N_2-N_1+1),
   such that $(P_\mu\cdot P_v + P_w)$ is square-free (Sec. 4.1.1).
   $Q \longleftarrow P_\mu\cdot P_v + P_w$
5. **Compute the coefficients** $\lambda_1,\ldots,\lambda_r$ by solving the system of Eq. (5) where $Q(x,y)=\prod_{j=1}^{r}(\beta_j x-\alpha_j y)$.
   For details and the representation of λ_j, see Sec. 4.1.2.
6. **Return** $\hat{f}(x,y)=\sum_{j=1}^{r}\lambda_j((\alpha_j,\beta_j)M^{-1}\binom{x}{y})^D$

We only consider families of Hankel matrices that have generic rank profile (see Def. 2.10). We prove that this property implies that v is related to a m.g.s. of a, and w is related to the unique m.g.s. of the first $2N_1^a$ elements of a, where N_1^a is defined by Prop. 2.3. This approach guarantees a softly-linear complexity bound (Sec. 4.2).

The assumption on the generic rank profile is not restrictive. As Prop. 2.11 indicates, we can achieve this property by applying a generic linear change of variables to f.

Lemma 3.2 *If $\{H_a^k\}_k$ has generic rank profile, then there is a vector $(v_0,\ldots,v_{N_1^a},-1)^\mathsf{T}$ in the kernel of $H_a^{N_1^a+1}$.*

PROOF. We skip the index a to simplify the notation.

If $N_1\neq N_2$, then the kernel of $H^{N_1+1}\in\mathbb{F}^{(N_2)\times(N_1+2)}$ has dimension 1 (Prop. 2.3). Hence, by the Rank-Nullity theorem, the rank of the matrix H^{N_1+1} is N_1+1.

The matrix $H^{N_1+1}\in\mathbb{F}^{(N_2)\times(N_1+2)}$ has generic rank profile (Def. 2.10) and $N_2\ge N_1+1$, hence its $(N_1+1)\times(N_1+1)$ leading principal submatrix of H^{N_1+1} is invertible. Let $(u_0,\ldots,u_{N_1},0)^\mathsf{T}$ be a vector in $\mathtt{Ker}(H^{N_1+1})$. Therefore $(u_0,\ldots,u_{N_1})^\mathsf{T}$ must be in the kernel of its $(N_1+1)\times(N_1+1)$ leading principal submatrix. Hence, $(u_0,\ldots,u_{N_1})^\mathsf{T}$ is the 0 vector. As the dimension of the kernel of H^{N_1+1} is one, there exists a vector $(v_0,\ldots,v_{N_1},v_{N_1+1})^\mathsf{T}$ in this kernel such that $v_{N_1+1}\neq 0$. After normalization, we take a vector in this kernel such that $v_{N_1+1}=-1$.

If $N_1=N_2$, the dimension of the kernel $H^{N_1+1}\in\mathbb{F}^{(N_1)\times(N_1+2)}$ is 2 (Prop. 2.3) and its rank is N_1 (Rank-Nullity theorem). As it has generic rank profile, its $N_1\times N_1$ leading principal submatrix is invertible. Hence, as in the previous paragraph, there are two vectors $(w_0,\ldots,w_{N_2-1},-1,0)$ and $(v_0,\ldots,v_{N_1-1},0,-1)$ in $\mathtt{Ker}(H^{N_1+1})$. □

Lemma 3.3 *If $(v_0,\ldots,v_{N_1^a},-1)^\mathsf{T}\in\mathtt{Ker}(H_a^{N_1^a+1})$, for a sequence a, then $(v_0,\ldots,v_{N_1^a})$ is a m.g.s. of a.*

PROOF. By Prop. 2.8, every generating sequence of a, of length k, is related to a non-zero vector in the kernel of H_a^k. By Prop. 2.3, the kernel of the matrices $\{H_a^k\}_{1\le k\le N_1^a}$ is trivial. Thus, there is no generating sequence of length smaller than N_1^a+1. Since $(v_0,\ldots,v_{N_1^a},-1)^\mathsf{T}\in\mathtt{Ker}(H_a^{N_1^a+1})$, $(v_0,\ldots,v_{N_1^a})$ is a m.g.s. of a. □

Algorithm 3 COMPUTE_V_AND_W

Input: Hankel matrices $\{H_a^k\}_{1\le k\le D}$ with **Generic Rank Profile**.
Output: Vectors v and w as Prop. 2.5

1. $(v_0,\dots,v_{N_1}) \leftarrow$ Min. gen. seq. of $(a_0,\dots,a_D)$
2. $v \leftarrow (v_0 \dots v_{N_1}, -1)^\mathsf{T}$
3. $(w_0,\dots,w_{N_1-1}) \leftarrow$ Min. gen. seq. of $(a_0,\dots,a_{2N_1-1})$
4. $w \leftarrow (w_0,\dots,w_{N_1-1},-1,\underbrace{0,\dots,0}_{N_2-N_1+1})^\mathsf{T}$
5. **Return** v and w

Lemma 3.4 *Alg. 3 computes v of Prop. 2.5.*

PROOF. When the corresponding Hankel matrices have generic rank profile, by Lem. 3.2, there is vector $(v_0,\dots,v_{N_1^a},-1)^\mathsf{T} \in \mathtt{Ker}(H_a^{N_1^a+1})$. Hence, by Lem. 3.3, $(v_0,\dots,v_{N_1^a})$ is a m.g.s. of a, and so the length of the m.g.s. of a is N_1^a+1. Given a m.g.s. $(\hat{v}_0,\dots,\hat{v}_{N_1^a})$ of a, by Prop. 2.8, $(\hat{v}_0,\dots,\hat{v}_{N_1^a},-1)^\mathsf{T} \in \mathtt{Ker}(H_a^{N_1^a})$. Hence, we can take $v=(\hat{v}_0,\dots,\hat{v}_{N_1^a},-1)^\mathsf{T}$, as the algorithm does. □

Lemma 3.5 *Given a family of Hankel matrices $\{H_a^k\}_k$ with generic rank profile, there is a vector $w \in \mathtt{Ker}(H^{N_2+1})$ linearly independent to all the elements in the U-chain of v of length N_2-N_1+1. It is of the form $w=(w_0\dots w_{N_1-1},-1,0,\dots,0)^\mathsf{T}$, where $(w_0,\dots,w_{N_1-1})$ is the unique m.g.s. of $(a_0,\dots,a_{2N_1-1})$, its number of tailing 0's is N_2-N_1+1, and v is from step 2 of Alg. 3.*

PROOF. By step 2 of Alg. 3, $v_{N_1+1} \neq 0$. Hence, none of the vectors in the vector space generated by the elements in the U-chain of v of length N_2-N_1+1 has its last N_2-N_1+1 positions equal to zero. As this property holds for the candidate vector for w, it cannot belong to this space.

The dimension of the kernel of $H^{N_2+1} \in \mathbb{F}^{(D-N_2)\times(N_2+2)}$ is N_2-N_1+2 (Prop. 2.3). As $N_1=D-N_2$ and we assumed having generic rank profile, the $N_1\times N_1$ leading principal submatrix of H^{N_2+1} is invertible. Hence, the following system has a unique solution,

$$\begin{pmatrix} a_0 & \cdots & a_{N_1-1} \\ \vdots & \ddots & \vdots \\ a_{N_1-1} & \cdots & a_{2N_1-2} \end{pmatrix} \cdot \begin{pmatrix} w_0 \\ \vdots \\ w_{N_1-1} \end{pmatrix} = \begin{pmatrix} a_{N_1} \\ \vdots \\ a_{2N_1-1} \end{pmatrix} \tag{6}$$

So, the vector $w=(w_0\dots w_{N_1-1},-1,0\dots 0)^\mathsf{T}$ is in $\mathtt{Ker}(H^{N_2+1})$.

As there is a unique solution to the system of Eq. (6), by Prop. 2.8, $(w_0\dots w_{N_1-1})$ is the unique generating sequence of $(a_0,\dots,a_{2N_1-1})$ with length N_1. Note that if $(u_0,\dots,u_n)$ is a generating sequence, then for any ρ, $\big(\rho(u_0,\dots,u_n,-1)+(0,u_0,\dots,u_n)\big)$ is a generating sequence too. Hence, it is the unique m.g.s. of $(a_0,\dots,a_{2N_1-1})$. □

Lemmas 3.2, 3.3 and 3.5 imply the correctness of Alg. 3.

Theorem 3.6 *Given the Hankel matrices $\{H_a^k\}_{(1\le k\le D)}$ with generic rank profile, Alg. 3 computes the vectors v and w of Prop. 2.5.*

Alg. 3 outputs vectors v and w of 2.5 when then input family of Hankel matrices have generic rank profile. The following lemma allows to check when two vectors are the vectors of Prop. 2.5. Recall that those vectors are not unique.

Lemma 3.7 *Given a sequence a of length $D+1$ and two non-zero vectors $u_1 := (v_0,\dots,v_{\hat{N}_1+1})^\mathsf{T}$ and $u_2 := (w_0,\dots,w_{\hat{N}_2+1})^\mathsf{T}$, they are the vectors v and w of Prop. 2.5 if and only if $u_1 \in \mathtt{Ker}(H_a^{\hat{N}_1+1})$, $u_2 \in \mathtt{Ker}(H_a^{\hat{N}_2+1})$, $\hat{N}_1+\hat{N}_2=D$ and P_{u_1} and P_{u_2} are relative prime.*

PROOF. With out loss of generality, let $\hat{N}_1 \le \hat{N}_2$. Clearly, if u_1 and u_2 are v and w, all the conditions hold (see Sec. 2.2).

For the other direction, if $u_1 \in \mathtt{Ker}(H_a^{\hat{N}_1+1})$, the kernel of $\mathtt{Ker}(H_a^{\hat{N}_1+1})$ is not trivial. Hence, by definition of N_1^a (Prop. 2.3), $N_1^a \le \hat{N}_1$. If $u_2 \in \mathtt{Ker}(H_a^{\hat{N}_2+1})$ and $\hat{N}_2 < N_2^a$, by Prop. 2.6, P_{u_1} and P_{u_2} should be divisible by P_v. But, as P_{u_1} and P_{u_2} are relative prime, this cannot happen. Hence, $u_2 \in \mathtt{Ker}(H_a^{\hat{N}_2+1})$ and $N_2^a \le \hat{N}_2$. By Prop. 2.3, $N_1^a+N_2^a=D$ and $N_1^a \le N_2^a$. At the same time we assumed $D=\hat{N}_1+\hat{N}_2$ and $\hat{N}_1 \le \hat{N}_2$. Therefore, $N_1^a=\hat{N}_1$ and $N_2^a=\hat{N}_2$. So, $u_1 \in \mathtt{Ker}(H_a^{N_1^a+1})$ and $u_2 \in \mathtt{Ker}(H_a^{N_2^a+1})$. Hence, u_1 is the vector v.

By Prop. 2.5, all the kernel polynomials spanned by the elements in the U-chain of u_1 are divisible by P_{u_1}. As P_{u_1} and P_{u_2} are relative primes, the vector u_2 that does not belong to the vector space generated by the elements in the U-chain of u_1 of length $(N_2^a-N_1^a+1)$. This implies that u_2 is the vector w. □

3.2 Computing a square-free polynomial Q

We can compute Q at step 4 of Alg. 2 in different ways. If P_v is square-free, then we set Q equal to P_v. If P_v is not square-free, by Lem. 3.1, we need to find a vector $\mu \in \mathbb{F}^{(N_2-N_1+1)}$ such that $Q_\mu := P_\mu \cdot P_v + P_w$ is square-free. By Prop. 2.6, P_v and P_w are relative prime. Thus, if we take a random vector μ, generically, Q_μ would be square-free. For this to hold, we have to prove that the discriminant of Q_μ is not identically zero. To simplify notation, in the following theorem we dehomogenize the polynomials.

Theorem 3.8 *Given two relative prime univariate polynomials $P_v(x)$ and $P_w(x)$ of degrees N_1+1 and N_2+1 respectively, let $Q_\mu(x) := P_\mu \cdot P_v + P_w \in \mathbb{F}[\mu_0,\dots,\mu_{N_2-N_1}][x]$. The discriminant of $Q_\mu(x)$ with respect to x is a non-zero polynomial.*

PROOF. The zeros in $\mathbb{F}$ of the discriminant of $Q_\mu(x)$ with respect to x, considering it a polynomial in μ, is the set $\{\mu \in \mathbb{F}^{N_2-N_1+1} : Q_\mu \text{ has double roots}\}$.

A univariate polynomial is square-free if and only if it does share any root with its derivative. Hence, $(\mu_0,\dots,\mu_{N_2-N_1})^\mathsf{T} \in \{\mu \in \mathbb{F}^{N_2-N_1+1} : Q_\mu \text{ has double roots}\}$ if and only if, there is $(\mu_0,\dots,\mu_{N_2-N_1},\alpha) \in \mathbb{F}^{N_2-N_1+1}\times\overline{\mathbb{F}}$ such that it is a solution of

$$\begin{cases} (P_\mu \cdot P_v + P_w)(x) = 0 \\ (P_\mu \cdot P_v' + P_\mu' \cdot P_v + P_w')(x) = 0. \end{cases} \tag{7}$$

We can rewrite P_μ as $\mu_0 + xP_{\hat{\mu}}$, where $\hat{\mu} := (\mu_1,\dots,\mu_{N_2-N_1})^\mathsf{T}$. From Eq. (7), we can eliminate μ_0 to obtain the equation $\big(P_v(P_{\hat{\mu}} \cdot P_v + P_w)' - P_v'(P_{\hat{\mu}} \cdot P_v + P_w)\big)(x) = 0$. As P_v and P_w are relative prime, this polynomial is not identically 0. Hence, for each $\hat{\mu}$, there is a finite number of solutions for this equation, bounded by the degree of the polynomial. Moreover, as the polynomials of Eq. (7) are linear in μ_0, each solution of the deduced equation is extensible to a finite number of solutions of Eq. (7). Therefore, there is a $\mu \in \mathbb{F}^{N_2-N_1+1}$, such that Q_μ is square-free. For this reason, the discriminant of $Q_\mu(x)$ is not identically zero. □

Corollary 3.9 *For every vector $(\mu_1,\dots,\mu_{N_2-N_1})^\mathsf{T} \in \mathbb{F}^{N_2-N_1}$, there are at most $D+1$ different values for $\mu_0 \in \mathbb{F}$ such that the polynomial $Q_\mu(x,y)$ is not square-free, where $\mu=(\mu_0,\dots,\mu_{N_2-N_1})^\mathsf{T}$.*

3.3 Correctness of Algorithm 2

For computing a decomposition for a binary form f, we need to compute the kernel of a Hankel matrix (Thm. 2.2). Alg. 3 computes correctly the vectors v and w that characterize the kernels of the family of Hankel matrices associated to f, if they have generic rank profile. To ensure this precondition, in step 1, we perform a linear change of variables using a random non-singular matrix M. We will decompose $L_M(f)$ to recover a decomposition for f (Lem. 2.9).

By Lem. 3.7, step 3 tests if the output of Alg. 3 is correct. By definition of Alg. 3, the algorithm computes v and w if the input

Hankel matrices, associated to $L_M(f)$, have generic rank profile (Thm. 3.6). Therefore, if the output is not correct, then the Hankel matrices do not have generic rank profile. Hence, we perform a new random change of variables and try again. There is a bounded number of "bad" changes of variables (Prop. 2.11), so we can always obtain a family of Hankel matrices with generic rank profile.

Once we obtain the vectors v and w, step 4 (see Cor. 3.9) and step 5 computes the coefficients $\alpha_j, \beta_j, \lambda_j$ of the decomposition. Hence, we have a decomposition for $L_M(f)$. As $L_{M^{-1}}(L_M(f)) = f$, by Lem. 2.9, a minimal decomposition of f is, $f(x,y) = \sum_{j=1}^{r} \lambda_j \cdot ((\alpha,\beta)\cdot M^{-1} \cdot \binom{x}{y})^D$.

Example. Consider $f(x,y) = y^4 + 8xy^3 + 18x^2y^2 + 16x^3y + 5x^4$. The Hankel matrices associated to f are related to the vector $(1,2,3,4,5)^\mathsf{T}$. Those matrices have generic rank profile. We apply Alg. 3 to compute v and w. The m.g.s. of $(1,2,3,4,5)$ is $(-1,2)$. Therefore, $v = (-1,2,-1)^\mathsf{T}$, $N_1 = 1$ and $N_2 = 3$. The m.g.s. of $(1,2)$ is (2), then $w = (2,-1,0,0,0)^\mathsf{T}$.

As $P_v = -(y-x)^2$, the rank of f is $N_2+1 = 4$ and we need to get a square-free kernel polynomial of degree 4. We consider the square-free kernel polynomial $(\frac{28}{5}y^2 - \frac{4}{5}xy - x^2)P_v + \frac{36}{5}P_w = \frac{36}{5}Q_{(\frac{7}{9},-\frac{1}{9},-\frac{5}{36})^\mathsf{T}} = (x - \frac{11}{5}y)(x-2y)(x+2y)(x+y)$. Solving the system of Eq. (5), we obtain the decomposition $f(x,y) = -\frac{625}{336}(\frac{11}{5}x+y)^4 + 3(2x+y)^4 + \frac{1}{21}(-2x+y)^4 - \frac{3}{16}(-x+y)^4$.

4. COMPLEXITY

In this section we study the complexity of Alg. 2.

4.1 Algebraic degree of the problem

4.1.1 The complexity of computing Q

Following the discussion of Sec. 3.2, we prove that, when the rank of the binary form is N_2+1, we can compute a square-free kernel polynomial Q of this degree such that the largest degree of its irreducible factors is N_1. Moreover, we prove that for almost all the choices of $(N_2 - N_1 + 1)$ pairwise linearly independent linear forms in $\mathbb{F}[x,y]$ there is a square-free kernel polynomial of H^{N_2+1} divisible by these forms.

Lemma 4.1 *Let f be a binary form of rank N_2+1. Given $(N_2 - N_1 + 1)$ pairwise linearly independent linear forms $(\beta_i x - \alpha_i y)$ such that none of them divides P_v, then there is a unique binary form P_μ of degree $N_2 - N_1$, such that the kernel polynomial $Q_\mu := P_\mu \cdot P_v + P_w$ is divisible by $\prod_i (\beta_i x - \alpha_i y)$.*

PROOF. Without loss of generality, we assume $\beta_i = 1$. By Prop. 2.6, for any polynomial P_μ of degree $N_2 - N_1$, Q_μ is a kernel polynomial. Since $\prod_i (x - \alpha_i y)$ divides $Q_\mu(x,y)$, it holds that for every α_i, $Q_\mu(\alpha_i, 1) = 0$. Hence, we can interpolate P_μ by noticing that $P_\mu(\alpha_j) = -\frac{P_w(\alpha_j)}{P_v(\alpha_j)}$.

The degree of P_μ is $(N_2 - N_1)$ and we interpolate it at $(N_2 - N_1 + 1)$ different points. Hence there is a unique interpolation polynomial P_μ. So, Q_μ is the unique kernel polynomial of H^{N_2+1} divisible by all those linear forms. □

Example (cont.) For the example of Sec. 3.3, we obtained the square-free kernel polynomial by choosing the factors $(x-2y)$, $(x+2y)$ and $(x+y)$. If we choose other factors such that Q_μ is square-free, we will obtain a different decomposition. Hence, f does not have a unique decomposition. This holds in general.

Corollary 4.2 *A decomposition is not unique iff the rank is N_2+1.*

Theorem 4.3 *Let the rank of f be N_2+1. There is a square-free kernel polynomial Q such that the largest degree of its irreducible factors is at most N_1.*

PROOF. Assume $\mathbb{F} = \mathbb{C}$. If the rank of f is N_2+1, then for each set of $N_2 - N_1 + 1$ linear forms $(x - \alpha_i y)$, following the assumptions of Lem. 4.1, there is a unique kernel polynomial. There is a rational map that realizes this relation (see the proof of Lem. 4.1). Let this map be $Q_{[\overline{\alpha}]}$, where $\overline{\alpha} = (\alpha_0, \ldots, \alpha_{N_2-N_1})$. The image of the map is contained in $\{P_\mu \cdot P_v + P_w : \mu \in \mathbb{F}^{N_2-N_1+1}\}$. This set and the domain of the rational map have the same dimension, $N_2 - N_1 + 1$.

Given a kernel polynomial $\hat{Q}(x,y)$, there is a finite number of distinct linear forms $(x - \alpha_i y)$ that divides it, because $\hat{Q}(x)$ is a non-zero univariate polynomial. Hence, the pre-image of an element in the image of $Q_{[\overline{\alpha}]}$ is a finite set. Therefore, the dimension of the image and the dimension of the domain are the same.

By Thm. 3.8, the non-square-free kernel polynomials form a hypersurface in the space of kernel polynomials of the shape $P_\mu \cdot P_v + P_w$. If we consider the pre-image of the intersection between this hypersurface and the image of the rational map, then its dimension is smaller than $N_2 - N_1 + 1$.

Therefore, generically, for $N_2 - N_1 + 1$ linear forms, the map $Q_{[\overline{\alpha}]}(x,y)$ results a square-free kernel polynomial.

If $\mathbb{F} \subset \mathbb{C}$, this proof can be adapted taking into consideration that $\mathbb{F}$ is a dense algebraic open subset of $\mathbb{C}$. □

Corollary 4.4 *Given a binary form f of rank r and degree D, there is a square-free kernel polynomial of degree r such that the biggest degree of its irreducible factors is* $\min(r, D-r+1)$.

PROOF. If the rank is $r = N_2+1$, then $\min(r, D-r+1) = N_1$. By Thm. 4.3, such a square-free kernel polynomial exists. If the rank is $r = N_1+1$ and $N_1 < N_2$, by Lem. 3.1, there is a square-free kernel polynomial of degree $\min(r, D-r+1) = N_1+1$. □

The previous result is related to the decomposition of tensors of the same border rank [5, Thm. 2][3, Thm. 23].

4.1.2 Complexity of computing λ

We compute the coefficients λ_j of the decomposition by solving a linear system involving a transposed Vandermonde matrix (Step 5 of Alg. 2). We follow Kaltofen and Yagati [17] to write the solution of Eq. (5) as the evaluation of a rational function over the roots of a univariate polynomial.

Definition 4.5 *Given a polynomial $P(x) := \sum_{i=0}^{n} a_i x^i$ and $0 < k \le n$, let $Quo(P,k)(x) := \sum_{i=k}^{n} a_i x^{i-k}$.*

Proposition 4.6 ([17, Sec. 5]) *If $\alpha_j \neq \alpha_i$, for all $i \neq j$, then there is a unique solution to the system of Eq. (8). Moreover, if the solution is $\lambda = (\lambda_1, \ldots, \lambda_r)^\mathsf{T}$ then, $\lambda_j = \frac{T}{Q'}(\alpha_j)$*

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_r \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{r-1} & \alpha_2^{r-1} & \cdots & \alpha_r^{r-1} \end{pmatrix} \lambda = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_r \end{pmatrix} \tag{8}$$

where $Q'(x)$ is the derivative of $Q(x) := \prod_{i=1}^{r}(x - \alpha_i)$, $R(x) := \sum_{i=1}^{r} a_{r-i} x^{i-1}$ and $T(x) := Quo\big(Q(x)\cdot R(x), r\big)$.

Lemma 4.7 *Given a binary form $f(x,y) := \sum_{i=0}^{D} \binom{D}{i} a_i x^i y^{D-i}$, let Q be a square-free kernel polynomial of degree r, obtained after step 5 of Alg. 2. Assume that y does not divide Q. Let α_j be the j-th roots of $Q(x)$, $Q'(x)$ be the derivative of $Q(x)$ and the polynomial $T(x) := Quo\big(Q(x)\cdot R(x), r\big)$, with $R(x) := \sum_{i=1}^{r} a_{r-i} x^{i-1}$. Then, each λ_j from step 5 in Alg. 3 can be written as $\lambda_j = \frac{T}{Q'}(\alpha_j)$.*

PROOF. As y does not divide Q, we can write it as $Q(x,y) = \prod_i (x - \alpha_i y)$, where all the α_i are different. Hence, as the $r \times r$

leading principal submatrix of Eq. (5) is invertible, we can restrict the problem to solve that $r \times r$ leading principal subsystem. This system is Eq. (8). Therefore, the proof follows from Prop. 4.6. □

Corollary 4.8 *Let Q be a square-free kernel polynomial related to a minimal decomposition of a binary form f of degree D, such that y does not divide Q. In this case, we can write f as $f(x,y) = \sum_{\{\alpha \in \overline{\mathbb{F}} | Q(\alpha)=0\}} \frac{T}{Q'}(\alpha) \cdot (\alpha x + y)^D$.*

If the square-free kernel polynomial related to a decomposition of rank r is divisible by y, we can compute $\{\lambda_j\}_{j<r}$ of Eq. (5) as in Lem. 4.7, by taking $\frac{Q}{y}$ as the kernel polynomial. It is wlog to consider $Q = P_{(u_0,\ldots,-1,0)^\intercal}$, because Q is square-free, and so y^2 can not divide it. Hence, $\lambda_r = a_D - \sum_{i=0}^{r-2} u_i a_{D-r+i+1}$ [32, Eq. 2.12].

To summarize the section, given a binary form f of rank r and a generic non-singular 2×2 matrix M, there is a square-free kernel polynomial Q of the degree r, such that the largest degree of its irreducible factors is bounded by $\min(r, D-r+1)$ (Cor. 4.8). If $Q(x,y)$ is not divisible by y, the decomposition is

$$f(x,y) = \sum_{\{\alpha \in \overline{\mathbb{F}} | Q(\alpha)=0\}} \frac{T}{Q'}(\alpha) \cdot ((\alpha,1) \cdot M^{-1} \cdot \tbinom{x}{y})^D,$$

where T and Q' are polynomials whose coefficients belong to $\mathbb{F}$ and whose degrees are bounded by r, defined in Lem. 4.7. When y divides Q, the form is similar.

4.2 Arithmetic complexity

Lemma 4.9 (Complexity of Alg. 3) *Given a binary form f of degree D, if its Hankel matrices $\{H_a^k\}_{(1 \le k \le D)}$ has generic rank profile, Alg. 3 computes v and w in $O(\mathtt{M}(D) \cdot \log(D))$.*

PROOF. Alg. 3 computes v and w of Prop. 2.5 as minimal generating sequences (m.g.s.). The superfast version of Berlekamp-Massey (BM) algorithm computes a m.g.s. in $O(\mathtt{M}(D) \cdot \log(D))$ when it is unique [10, Sec. 12.3].

We consider the computation of v (step 1 of Alg. 3). The m.g.s. of $a = (a_0, \ldots, a_D)$ has always length $N_1 + 1$ (Lem. 3.4). But, it is not always unique. The length of a is $D+1$ and $N_1 + N_2 = D$. Therefore $N_1 < N_2$ if and only if $2(N_1+1) \le D+1$. Massey [22] states that the m.g.s. is unique if and only if its length is less or equal to half of the length of the recurrent sequence. Hence, if $N_1 \neq N_2$, the m.g.s. is unique and so we can compute it superfast.

We consider the case where $N_1 = N_2$. As $N_1 + N_2 = D$, it holds that $N_1 = \frac{D}{2}$. Thus, the matrix $H_a^{N_1}$ is square and has a trivial kernel (Prop. 2.3). Hence, it is invertible and the system $H_a^{N_1} \cdot u = \hat{a}$ has always a solution; where u is a vector of unknowns, $\hat{a} = (a_{N_1^a+1}, \ldots, a_D, c)^\intercal$, and c is arbitrary.

By Prop. 2.8, for each c, the solution of the system is a generating sequence of $(a_0, \ldots, a_D, c)$ of length $N_1^a + 1$. Reasoning as in the proof of Lem. 3.5, since this generating sequence of length $N_1^a + 1$ is unique, it has to be the unique m.g.s. of $(a_0, \ldots, a_D, c)$. In addition it is also a generating sequence of $(a_0, \ldots, a_D)$. As it has the length, $N_1^a + 1$, of the m.g.s. of $(a_0, \ldots, a_D)$, it is minimal. We compute it superfast as an m.g.s. of $(a_0, \ldots, a_D, c)$ for any c.

It remains to identify when $N_1 = N_2$. We adopt a try and error approach. For an input $a = (a_0, \ldots, a_d)$ we run the superfast version of BM. The algorithm terminates and outputs a sequence u in $O(\mathtt{M}(D) \cdot \log(D))$, independently of the uniqueness of m.g.s. [10, Sec. 12.3]. We verify the correctness of the output.

First we check if u is a generating sequence of a. By Prop. 2.8 this is equivalent to test if u belongs to the kernel of certain Hankel matrix; it corresponds to polynomial multiplication [27, Sec. 2.4].

If u is not a generating sequence of a, then the m.g.s. of a is not unique, and so $N_1 = N_2$. If u is a generating sequence of a, then we proceed as follows: Let L be the length of u. We compare $2L$ and $D+1$. If $2L \le D+1$ then m.g.s. of a is unique and $N_1 \neq N_2$. If L_m is the length of m.g.s., then $L_m \le L$ and $2L_m \le D+1$. Hence, by Massey's uniqueness argument [22], the m.g.s. is unique. In this case the m.g.s. of a is u. If $2L > D+1$, then the m.g.s. of a is not unique and $N_1 = N_2$. To see this notice that if the m.g.s. of a is unique then u has to be the unique m.g.s., by the correctness of BM. And so by Massey's uniqueness argument [22] it holds $2L \le D+1$ and $N_1 \neq N_2$. Therefore, we identify when $N_1 = N_2$ in $O(\mathtt{M}(D))$.

The vector w always corresponds to a unique m.g.s (Lem. 3.5). Therefore, we can always compute it efficiently. □

Lemma 4.10 (Complexity of computing Q) *Given the vectors v and w from Alg. 3, we compute a square-free polynomial $Q_\mu := P_\mu \cdot P_v + P_w$ with the algebraic degree of Cor. 4.4 in $O(\mathtt{M}(D) \cdot \log(D))$.*

PROOF. To compute the vector μ, we choose randomly $N_2 - N_1 + 1$ linear forms and we proceed as in Lem. 4.1. The complexity bound is due to multi-point evaluation and interpolation of a univariate polynomial [10, Ch. 10]. □

Theorem 4.11 *Alg. 2 computes a symbolic decomposition (see Introduction) of a binary form in $O(\mathtt{M}(D) \cdot \log(D))$.*

PROOF. The complexity of performing the random linear change of variables of step 1 is bounded by $O(\mathtt{M}(D) \cdot \log(D))$ (Sec. 2.4). In step 2, we call Alg. 3. It computes the candidate vectors v and w of Prop. 2.5 in $O(\mathtt{M}(D) \cdot \log(D))$ (Lem. 4.9). At step 3 we check if the candidates are valid using Lem. 3.7 in $O(\mathtt{M}(D) \cdot \log(D))$. After random linear change of variables at step 1, they are correct with probability 1. Using the vectors v and w, step 4 computes a square-free polynomial (Thm. 4.3) in $O(\mathtt{M}(D) \cdot \log(D))$ (Cor. 4.4). In step 5 we compute the rational function that describes the solution of the system in Eq. (5), in $O(\mathtt{M}(D) \cdot \log(D))$ [17]. Finally, step 6 returns the decomposition. □

Remark 4.12 *To output an approximation of the terms of the minimal decomposition, with a relative error of $2^{-\varepsilon}$, we use Pan's algorithm [28] [23, Thm. 15.1.1] to approximate the roots of Q. In this case the complexity becomes $O\big(D \log^2(D)\big(\log^2(D) + \log(\varepsilon)\big)\big)$.*

4.3 Bit complexity

Let $f \in \mathbb{Z}[x,y]$ be a binary form as in Eq. (1), of degree D and let τ be the maximum bitsize of the coefficients a_i. We study the bit complexity of computing suitable approximations of the α_j's, β_j's, and λ_j's of Eq. (3), say $\widetilde{\alpha}_j$, $\widetilde{\beta}_j$ and $\widetilde{\lambda}_j$ respectively, that induce an approximate decomposition correct up to ℓ bits. That is a decomposition such that $\|f - \sum_j \widetilde{\lambda}_j (\widetilde{\alpha}_j x + \widetilde{\beta}_j y)^D\|_\infty \le 2^{-\ell}$. We need to estimate an upper bound on the number of bits that are necessary to perform all the operations of the algorithm.

Initially the algorithm performs a linear change of variables, according to the discussion in Sec. 2.4. The element t of the matrix $M = \left(\begin{smallmatrix} 1 & t \\ 0 & 1 \end{smallmatrix}\right)$ that we use for the linear change of variables has less than D^3 forbidden values. Therefore, at least one of the first D^3 integers is valid to perform the transformation. Hence, in the worst case, t has bitsize $O(\log(D))$. After the transformation we obtain a new polynomial, F, of the same degree as f and of maximum coefficient bitsize $O(D \lg(D) + \tau) = \widetilde{O}(D + \tau)$.

It is wlog to consider $y = 1$, because we have already performed the linear change of variables, and the degree of the binary form does not change with this substitution.

In the sequel, the algorithm computes the vectors v and w and, through them, the polynomial Q. This costs $\widetilde{O}(D)$. The degree of Q is $\le D$ and its maximum coefficient bitsize is $\widetilde{O}(D^2 + D\tau) =$

σ as it is the Bézout coefficient of an EGCD computation [10, Sec. 12.3]. Let α_j be the roots of Q. We isolate them in $\widetilde{O}_B(D^2\sigma)$ [28]. For the (aggregate) separation bound of the roots it holds that $-\lg\prod_j \Delta(\alpha_j) = O(D\sigma + D\lg(D))$. We approximate all the roots up to accuracy $2^{-\ell_1}$ in $\widetilde{O}_B(D^2\sigma + D\ell_1)$ [29]. That is, we compute absolute approximations of α_j, say $\widetilde{\alpha}_j$, such that $|\alpha_j - \widetilde{\alpha}_j| \leq 2^{-\ell_1}$.

The next step consists in solving the (transposed) Vandermonde system, $V(\widetilde{\alpha})^{\mathsf{T}}\lambda = a$, where $V(\widetilde{\alpha})$ is the Vandermonde matrix we construct with the roots of Q, λ is a vector contains the coefficients of decomposition, and a is a vector containing the coefficients of F, see also Eq. (5). We know the entries of $V(\widetilde{\alpha})$ up to ℓ_1 bits. Therefore, we can compute the elements of the solution vector λ with an absolute approximation correct up to $\ell_2 = \ell_1 - O(D\lg(D)\sigma - \lg\prod_j \Delta(\alpha_j)) = \ell_1 - O(D\lg(D)\sigma)$ bits [30, Thm. 29]. That is, we compute $\widetilde{\lambda}_j$'s such that $|\lambda_j - \widetilde{\lambda}_j| \leq 2^{-\ell_2}$.

At this point we have obtained the approximate decomposition $\sum_{j=1}^{r} \widetilde{\lambda}_j(\widetilde{\alpha}_j x + 1)^D$; it corresponds to a polynomial $\widetilde{F}$. We apply the inverse transformation, $M^{-1} = \left(\begin{smallmatrix} 1 & -t \\ 0 & 1 \end{smallmatrix}\right)$, to obtain an approximate decomposition, say $\widetilde{f}$, for f, which is

$$\widetilde{f}(x,y) = \sum_{j=1}^{r} \widetilde{\lambda}_j(\widetilde{\alpha}_j x + (1 - \widetilde{\alpha}_j t)y)^D.$$

To estimate the accuracy of $\widetilde{f}$ we need to expand the approximate decomposition and consider it as a polynomial in x. We do not actually perform this operation; we only estimate the accuracy as if we were. First, we expand each $(\widetilde{\alpha}_j x + (1 - \widetilde{\alpha}_j t)y)^D$. This results polynomials with coefficients correct up $\ell_3 = \ell_2 - O(D\sigma) = \ell_1 - O(D\lg(D)\sigma) - O(D\sigma) = \ell_1 - O(D\lg(D)\sigma)$ bits [30, Lemma 19]. Next, we multiply each such polynomial with $\widetilde{\lambda}_i$, and we collect the coefficients for the various powers of x. Each coefficient is the sum of $r \leq D$ terms. The last two operations do not affect, asymptotically, the precision. Therefore, the polynomial $\widetilde{f} = \sum_{j=1}^{r} \widetilde{\lambda}_j(\widetilde{\alpha}_j x + (1 - \widetilde{\alpha}_j t)y)^D$ that corresponds to the approximate decomposition has an absolute approximation such that $\|f - \widetilde{f}\| \leq 2^{-\ell_1 + O(D\lg(D)\sigma)}$. To achieve an accuracy of $2^{-\ell}$ in the decomposition, such that $\|f - \widetilde{f}\| \leq 2^{-\ell}$, we should choose $\ell_1 = \ell + O(D\lg(D)\sigma)$. Thus, all the computations should be performed with precision of $\ell + O(D\lg(D)\sigma)$ bits. The bit complexity of computing the decomposition of f up to ℓ bits is dominated by the solving and refining process and it is $\widetilde{O}_B(D\ell + D^2\sigma)$. If we substitute the value for σ, then we arrive at the complexity bound of $\widetilde{O}_B(D\ell + D^4 + D^3\tau)$.

Theorem 4.13 *Let $f \in \mathbb{Z}[x,y]$ be of degree D and maximum coefficient bitsize τ. We compute an approximate decomposition of accuracy $2^{-\ell}$ in $\widetilde{O}_B(D\ell + D^4 + D^3\tau)$.*

Acknowledgments

The authors thank the anonymous reviewers for their helpful comments. Matías Bender thanks Joos Heintz for supervising his Master's thesis. The authors were partially supported by French ANR-11-BS02-0013 HPAC project. Elias Tsigaridas was partially supported by an FP7 Marie Curie Career Integration Grant.

References

[1] C. Bajaj. The algebraic degree of geometric optimization problems. *Discrete & Computational Geometry*, 3(1):177–191, 1988.

[2] E. R. Berlekamp. *Nonbinary BCH decoding*. University of North Carolina, 1966.

[3] A. Bernardi, A. Gimigliano, and M. Ida. Computing symmetric rank for symmetric tensors. *J. Symbolic Comput.*, 46(1):34–53, 2011.

[4] J. Brachat, P. Comon, B. Mourrain, and E. Tsigaridas. Symmetric tensor decomposition. *Linear Algebra and its Applications*, 433(11):1851–1872, 2010.

[5] G. Comas and M. Seiguer. On the rank of a binary form. *Foundations of Computational Mathematics*, 11(1):65–78, 2011.

[6] P. Comon. Tensors: a brief introduction. *IEEE Signal Processing Magazine*, 31 (3):44–53, 2014.

[7] P. Comon and B. Mourrain. Decomposition of quantics in sums of powers of linear forms. *Signal Processing*, 53(2):93–107, 1996.

[8] P. Comon, G. Golub, L.-H. Lim, and B. Mourrain. Symmetric tensors and symmetric tensor rank. *SIAM J. on Matrix Analysis & Apps*, 30(3):1254–1279, 2008.

[9] A. Dür. On computing the canonical form for a binary form of odd degree. *J. Symbolic Comput.*, 8(4):327–333, Oct 1989. .

[10] J. v. z. Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, Cambridge, 2013.

[11] S. Gundelfinger. Zur theorie der binären formen. *Journal für die reine und angewandte Mathematik*, 100:413–424, 1887.

[12] G. Heinig and K. Rost. *Algebraic methods for Toeplitz-like matrices and operators*. Springer, 1984.

[13] U. Helmke. Waring's problem for binary forms. *Journal of pure and applied algebra*, 80(1):29–45, 1992.

[14] A. Iarrobino and V. Kanev. *Power sums, Gorenstein algebras, and determinantal loci*. Springer, 1999.

[15] K. Imamura and W. Yoshida. A simple derivation of the berlekamp-massey algorithm and some applications. *IEEE Transactions on Information Theory*, 33(1):146–150, 1987.

[16] E. Jonckheere and C. Ma. A simple Hankel interpretation of the Berlekamp-Massey algorithm. *Linear Algebra and its Applications*, 125:65–76, 1989.

[17] E. Kaltofen and L. Yagati. Improved sparse multivariate polynomial interpolation algorithms. In *Symbolic and Algebraic Computation*, pages 467–474. Springer, 1989.

[18] J. P. Kung. Canonical forms of binary forms: variations on a theme of Sylvester. *Institute for Mathematics and Its Applications*, 19:46, 1990.

[19] J. P. Kung and G.-C. Rota. The invariant theory of binary forms. *Bull. Amer. Math. Soc.*, 10(1):27–85, 1984.

[20] J. M. Landsberg. *Tensors: geometry and applications*. AMS, 2012.

[21] W. Manthey, U. Helmke, and D. Hinrichsen. Topological aspects of the partial realization problem. *Math. of Control, Signals and Systems*, 5(2):117–149, 1992.

[22] J. L. Massey. Shift-register synthesis and bch decoding. *Information Theory, IEEE Transactions on*, 15(1):122–127, 1969.

[23] J. M. McNamee and V. Y. Pan. *Numerical methods for roots of polynomials (II)*, chapter 15. Elsevier, 2013.

[24] J. Nie, K. Ranestad, and B. Sturmfels. The algebraic degree of semidefinite programming. *Mathematical Programming*, 122(2):379–405, 2010.

[25] G. Norton. On the minimal realizations of a finite sequence. *J. Symbolic Comput.*, 20(1):93–115, Jul 1995.

[26] L. Oeding and G. Ottaviani. Eigenvectors of tensors and algorithms for waring decomposition. *J. Symbolic Comput.*, 54:9–35, 2013.

[27] V. Pan. *Structured matrices and polynomials: unified superfast algorithms*. Springer, 2001.

[28] V. Y. Pan. Univariate polynomials: Nearly optimal algorithms for numerical factorization and root-finding. *J. Symbolic Comput.*, 33(5):701 – 733, 2002.

[29] V. Y. Pan and E. Tsigaridas. Accelerated Approximation of the Complex Roots and Factors of a Univariate Polynomial. *Theor. Comput. Sci.*, 2016. URL https://hal.inria.fr/hal-01105267. (To appear).

[30] V. Y. Pan and E. Tsigaridas. Nearly optimal computations with structured matrices. *Theor. Comput. Sci.*, 2016. URL https://hal.inria.fr/hal-01105263.

[31] B. Reznick. Homogeneous polynomial solutions to constant coefficient pde's. *advances in mathematics*, 117(2):179–192, 1996.

[32] B. Reznick. On the length of binary forms. In *Quadratic and Higher Degree Forms*, pages 207–232. Springer, 2013.

[33] B. A. Reznick. *Sums of even powers of real linear forms*, volume 463. American Mathematical Soc., 1992.

[34] J. Sylvester. An essay on canonical forms, supplement to a sketch of a memoir on elimination, transformation and canonical forms. In *The collected papers of JJ Sylvester*, volume 1, pages 203–216. Cambridge University Press, 1904.

[35] J. Sylvester. On a remarkable discovery in the theory of canonical forms and of hyperdeterminants. In *The collected papers of JJ Sylvester*, volume 1, pages 265–283. Cambridge University Press, 1904.