Linear Algebra for Computing Gröbner Bases of Linear Recursive Multidimensional Sequences
Jérémy Berthomieu, Brice Boyer, Jean-Charles Faugère

To cite this version:

HAL Id: hal-01237861
https://hal.inria.fr/hal-01237861
Submitted on 7 Dec 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Linear Algebra for Computing Gröbner Bases of Linear Recursive Multidimensional Sequences

Jérémy Berthomieu a,b,c jeremy.berthomieu@lip6.fr Brice Boyer a,b,c brice.boyer@lip6.fr Jean-Charles Faugère c,a,b jean-charles.faugere@inria.fr

a Sorbonne Universités, UPMC Univ Paris 06, Équipe POLSYS, LIP6, F-75005, Paris, France
b CNRS, UMR 7606, LIP6, F-75005, Paris, France
c INRIA, Équipe POLSYS, Centre Paris – Rocquencourt, F-75005, Paris, France

ABSTRACT
Sakata generalized the Berlekamp – Massey algorithm to n dimensions in 1988. The Berlekamp – Massey – Sakata (BMS) algorithm can be used for finding a Gröbner basis of a 0-dimensional ideal of relations verified by a table. We investigate this problem using linear algebra techniques, with motivations such as accelerating change of basis algorithms (FGLM) or improving their complexity.

We first define and characterize multidimensional linear recursive sequences for 0-dimensional ideals. Under genericity assumptions, we propose a randomized preprocessing of the table that corresponds to performing a linear change of coordinates on the polynomials associated with the linear recurrences. This technique then essentially reduces our problem to using the efficient 1-dimensional Berlekamp – Massey (BM) algorithm. However, the number of probes to the table in this scheme may be elevated. We thus consider the table in the black-box model: we assume probing the table is expensive and we minimize the number of probes to the table in our complexity model. We produce an FGLM-like algorithm for finding the relations in the table, which lets us use linear algebra techniques. Under some additional assumptions, we make this algorithm adaptive and reduce further the number of table probes. This number can be estimated by counting the number of distinct elements in a multi-Hankel matrix (a multivariate generalization of Hankel matrices); we can relate this quantity with the geometry of the final staircase. Hence, in favorable cases such as convex ones, the complexity is essentially linear in the size of the output. Finally, when using the LEX ordering, we can make use of fast structured linear algebra similarly to the Hankel interpretation of Berlekamp – Massey.

Categories and Subject Descriptors
I.1.2 [Symbolic and Algebraic Manipulation]: Algorithms

General Terms
Theory, Algorithms

Keywords
BMS and FGLM algorithms, Gröbner basis computation, 0-dimensional ideal, multidimensional linear recursive sequence

1. INTRODUCTION
A fundamental problem in Computer Science is to estimate the linear complexity of an infinite sequence $S$: this is the smallest length of a recurrence satisfied by $S$ or the length of the shortest linear feedback shift register (LFSR) which generates it. From an algorithmic point of view, the Berlekamp – Massey algorithm (BM) [1, 14] solves this problem in the one dimensional case. Generalizations of linear recurrence sequences to $n$ dimensions were proposed by several authors [3, 17, 19]. Sakata generalized the BM algorithm [19, 21] to $n$ dimensions; in particular, the so-called BMS algorithm is able to compute a Gröbner basis [19, Lem. 5] of the ideal of relations satisfied by the input sequence.

Direct and important application of such generalization can be found in Coding Theory: the BMS algorithm can be used to decode $n$-dimensional cyclic codes [20] which are generalization of Reed Solomon codes. Another application is the computation of Gröbner bases since recent versions of the Sparse-FGLM algorithm [5] rely heavily on BM and BMS algorithms.

Related work
Linear Prediction dates back to Gauß in the 18th century: given a discrete set of original values $(u_i)_{i \in \mathbb{N}}$, the goal is to find the best coefficients, in the least-squares sense, $(\alpha_i)_{i \in \mathbb{N}}$ that will approximate $u_i$ by $- \sum_{d=1}^{D} \alpha_d u_{i-d}$. This problem is equivalent to solving a linear system which is indeed a symmetric Toeplitz matrix. This problem has been extensively used in Digital Signal Processing theory and applications. In the numerical world, methods such as the Levinson – Durbin recursion can be used to solve this problem. Hence, to some extent, the original Levinson – Durbin problem in Norbert Wiener’s Ph.D. thesis [13, 22] predates the Hankel interpretation of the Berlekamp – Massey algorithm (for instance [7]).

We refer to [9, 10] for a very nice classification of the BM algorithms for solving this problem and generalization to matrix sequences, see also [8]. Of particular importance for us is the solution of the underlying linear system in Toeplitz/Hankel form. Let us also mention that a call of BM on sequence $(u_0, u_1, ... , u_{2d-1})$ will behave as the extended Euclidean algorithm with input polynomials $x^{2d}$ and $u_0 x^{2d-1} + u_1 x^{2d-2} + \cdots + u_{2d-1}$ making BM a simplified version of the extended Euclidean algorithm.

BMS extends the algebraic form of BM to $n$ dimensions [18]. In the case of 0-dimensional ideals, this algorithm can be applied for Gröbner basis computations [19].

Contributions
First of all, we define and characterize linear recurrence sequences in $n$ dimensions. More precisely, we link them with 0-dimensional ideals and define their order as the degree of the ideal generated by the relations satisfied by the sequence (see Sec. 2). Classically, this number is also the size of the staircase of the Gröbner basis (the canonical set of generators for the residue class ring).

A first idea is to try to use the standard BM algorithm to solve the $n$ dimensional case: we give a randomized preprocessing on the input sequence; this preprocessing will yield a new table which,
with good probability, will only have one linear recurrence relation (see Sec. 3). Exploiting this property yields Th. 1.

**Theorem 1.** Let \( u = (u_1, \ldots, u_n) \) be a \( n \)-dimensional linear recursive sequence over \( \mathbb{K} \). Let \( d \in \mathbb{N} \). When the size of \( \mathbb{K} \) is large enough, we can find an equivalent basis of its relations for all \( i_1 + \cdots + i_n \leq 2d \) in randomized time in \( O(n^{2d} + n \log d) \) operations in \( \mathbb{K} \), where \( M(d) \) is the complexity of multiplying two polynomials of degree at most \( d - 1 \).

Next, we propose two FGLM-like algorithms for computing Gröbner bases of the ideal of relations in Sec. 4 and 5. Both algorithms are based only on simple linear algebra operations: they search to extract maximal full rank submatrices of a multi-Hankel matrix (a multivariate generalization of Hankel matrices). Given a bound on the maximal degree of the elements in the final Gröbner basis, the first algorithm is able to compute it. This algorithm is efficient when the order of the sequence is relatively big. On the other hand, when the order of the sequence is abnormally small we propose an output sensitive probabilistic algorithm: this time an estimate of the order of the sequence is given.

An important parameter of the complexity of the algorithms is the number of table queries. Indeed, in some applications, it is very costly to compute one element \( u_i, \ldots, u_n \) of the table; thus the number of table queries has to be minimized. For instance, in the FGLM algorithm, each element of the table requires a matrix-vector product to be computed. This number can be estimated by counting the number of distinct elements in a multi-Hankel matrix; moreover, we can relate this quantity with the geometry of the final staircase:

**Theorem 2.** The number of queries to the table is the cardinal of set \( 2S = \{ w \mid (u, v) \in S^2 \} \) where \( S \) is the staircase of the ideal.

We show that in favorable cases such as convex ones, the complexity of computing \( u_i, \ldots, u_n \) is \( d^{|S|} \) in \( \mathbb{K}[x] \). The number of queries to the table is the cardinal of \( S \) and \( |S| \) grows quadratically. Hence, we can compute the table in \( O(|S|) \) time for the lex ordering, multi-Hankel matrices are strongly structured and can be solved with fast algorithms. In Sec. 6, we give two approaches based on the notion of displacement rank and the polynomial multiplication interpretation, we refer to [2] for both. If \( d_i \) is the maximal degree of the polynomials in \( x_i \), then solving the \( mth \) multi-Hankel matrix can be done in \( O(M(2m^2 - 1d_1 \cdots d_m)) \) operations in the base field.

Finally, we left as an open question whether our algorithms could be seen as a matrix version of BMS.

2. **DEFINITION AND CHARACTERIZATION OF LINEAR RECURSIVE SEQUENCES**

In Def. 2, we generalize the notion of a linear recursive sequence from the 1-dimensional case to the multidimensional case based on the terms of the sequence. We also give a characterization of such a sequence based on the linear recurrence relations [17, Def. 21] the sequence satisfies and provide a proof in Prop. 3 that this characterization is equivalent to Def. 2. This characterization is related to but more restrictive than [3, Def. 2]. Finally, in Sec. 2.2, we adopt an FGLM viewpoint to describe a multidimensional sequence.

Let us recall that a 1-dimensional sequence \( u = (u_i)_{i \in \mathbb{N}} \) defined over a field \( \mathbb{K} \) is linear recursive of order \( d \) if \( \exists \alpha_0, \ldots, \alpha_{d-1} \in \mathbb{K} \) s.t. for all \( i \in \mathbb{N}, u_{i+d} + \sum_{j=0}^{d-1} \alpha_j u_{i+j} = 0 \) and \( d \) is minimal. We shall denote \( R(i) = u_{i+d} + \sum_{j=0}^{d-1} \alpha_j u_{i+j} \). In general, \( \text{pol}_g(R)(x) = x^d + \sum_{k=0}^{d-1} \alpha_k x^k \) is called the characteristic polynomial of the sequence. It is well-known that the vector space of linear sequences with the linear recurrence relation \( R \) has dimension \( d \) whose canonical basis is \((u_0, \ldots, u_{d-1})\) verifying \( \forall i, 0 \leq i, d - 1, u_{i+d} = \delta_{i,0} \) Kronecker's delta. Other bases are natural, let us mention the one using the roots of \( \text{pol}_g(R) \): \((u_0, \ldots, u_{d-1}, \ldots, u_{-1}, \ldots, u_{-d+1})\) verifying \( u_i e^{\mu i} = \zeta e^{\mu i} \), where \( \zeta \) is a root of multiplicity \( \mu \). In other words, a linear sequence of order \( d \) is uniquely determined by its characteristic polynomial, or equivalently its minimal linear recurrence relation of order \( d \), and its \( d \) first terms.

To simplify notations, let \( i = (i_1, \ldots, i_n) \in \mathbb{N}^n \) and \( x = (x_1, \ldots, x_n) \). As usual, we let \( x^i \) denote \( x_1^{i_1} \cdots x_n^{i_n} \) and \( |i| = i_1 + \cdots + i_n \). We also denote \( e_i \) the \( i \)th vector of the canonical basis of \( \mathbb{Z}^n \). In the following, we consider a \( n \)-dimensional sequence \( u = (u_i)_{i \in \mathbb{N}^n} \).

As in the one-dimensional case, we can define a linear recurrence relation and its associated polynomial. Such a relation was called \( n \)-dimensional linear recursion relation in [17, Def. 21].

**Definition 1.** Let \( u = (u_i)_{i \in \mathbb{N}^n} \) be a sequence with coefficients in \( \mathbb{K} \). Let \( \mathcal{K} \subseteq \mathbb{N}^n \) be finite. The set \( \{ a_k \in \mathcal{K} \} \) defines a linear recurrence relation \( R \) if for all \( i \in \mathbb{N}^n \), \( R(i) = \sum_{k \in \mathcal{K}} a_k u_{i-k} = 0 \).

The associated polynomial is then \( \text{pol}_g(R)(x) = \sum_{k \in \mathcal{K}} a_k x_k \). Conversely, from any polynomial \( P \in \mathbb{K}[x], P = \sum_{k \in \mathcal{K}} a_k x_k \), we define the non-instantiated associated relation \( \text{rel}_a(P)(i) = \sum_{k \in \mathcal{K}} a_k u_{i-k} \).

In the end we can always shift any polynomials so that it is enough to evaluate such a relation in \( I = \{0, \ldots, 0\} \); consequently we will use the following convention: \( [x_1^0 \cdots x_n^0] = [x_1^0 \cdots x_n^0] = [u_0, \ldots, u_n] \) and in general \( [P] = [P]_{i \in I} = \sum_{k \in \mathcal{K}} a_k u_{i-k} \).

**Example 1.** If \( P(x, y) = xy - x - 1 \in \mathbb{K}[x, y] \) then \( [P] = u_1 - u_0, u_1 u_0 - u_0, 0 \) and \( [x^2 y^2] = u_3 - u_2 u_1 - u_2^2 \).

We are now in a position to define a linear recursive sequence.

**Definition 2.** Let \( u = (u_i)_{i \in \mathbb{N}^n} \) is a \( n \)-dimensional sequence with coefficients in \( \mathbb{K} \). The sequence \( u \) is linear recursive if from a nonzero finite number of initial terms \( u_i \in \mathbb{S} \) and a finite number of linear recurrence relations, without any contradiction, one can compute any term of the sequence.

For contradictions arising, we refer to Sakata's example with initial terms \( \{0,0, u_1,0,u_0,1\} \) and relations \( u_{i+2} - u_{i+1} - u_{i+1} - u_{i+1} - u_{i+1} - u_{i+1} - u_{i+1} = 0 \), in [21, p. 147]. Indeed, one can derive a new relation \( u_{i+1} - u_{i+1} - 0 \) meaning that \( u_0,1 \) is determined by the knowledge of \( u_0,0 \) and the set of initial terms is only \( \{u_0,0,1\} \).

In other words, a linear recursive sequence is a special case of a holonomic (or P-recurasive) sequence whose recurrence relations once have constant coefficients (see [11]).

2.1 **Gröbner bases and characterization of linear recursive sequences**

Before giving another characterization of such sequences based on Gröbner bases, we recall some definitions and properties of Gröbner bases and admissible monomial orders.

An admissible monomial order \( \prec \) is an order on monomials of \( \mathbb{K}[x] \) s.t. for any monomial \( x^s \neq 1, 1 \prec s \) and for any monomials \( t, m \), s.t. \( t \prec s, ms \prec mt \). This implies that there does not exist any infinite strictly decreasing sequences of monomials.

The leading term of a polynomial \( P \) for \( \prec \), denoted \( \text{LT}_\prec(P) \) or \( \text{LT}(P) \) if no confusion can arise, is the greatest monomial of \( P \) multiplied by its coefficient.

Whenever an ideal \( I \) is homogeneous, i.e. spanned by homogeneous polynomials, a truncated Gröbner basis \( B_0 \) up to degree \( d \) for \( \prec \), or \( d \)-truncated Gröbner basis, is a set of polynomials \( \mathcal{G} = \{g_1, \ldots, g_\ell\} \) s.t. for all \( f \in S \), if \( \deg f \leq d \) then there exists \( g_i \in \mathcal{G} \) s.t. \( \text{LT}(g_i) \) divides \( \text{LT}(f) \). This can be computed using any Gröbner basis algorithm by discarding critical pairs of degree greater than \( d \).
For an affine ideal, we can also define a d-truncated Gröbner basis as the output of a Gröbner basis algorithm discarding critical pair of degree higher than d. That is to say, for any critical pair $\{f_i, f_j\}$, if $\deg \text{LT}(f_i) + \deg \text{LT}(f_j) > \deg \text{lcm}(\text{LT}(f_i), \text{LT}(f_j)) > d$, then $\{f_i, f_j\}$ is not taken into account. In this situation, a truncated Gröbner basis $\mathcal{G} = \{g_1, \ldots, g_r\}$ up to degree d will span the subspace of polynomials $\mathcal{I}_d$ of degree up to d.

PROPOSITION 3. Equivalently, a n-dimensional sequence defined over a field $K$ is linear recursive if the ideal I spanned by all the polynomials associated to its linear recurrence relations has dimension 0, i.e. has a finite number of solutions in the algebraic closure of $K$.

We shall prove that both definitions are equivalent. First, let us consider a sequence verifying Def. 2. Let a be a monomial ordering. For a linear recurrence relation $R$ with support in $\mathcal{S}$, there is a maximal element $d$ in $\mathcal{S}$ for which $R$ is reduced to $R(d) = a_{i+1} - a_i \sum_{j \leq i} b_{i-j} a_j = 0, \forall a_i \in K$. Let $R_{1, \ldots, R_n}$ be a set of linear recurrence relations, written as above, sufficient to compute $u$ together with $\{a_i \in S\}$. Let us show that for all $i$, there exist $a_i, b_j \in K$ such that $u_i = \sum_{j \leq i} b_{i-j} a_j = 0$. Obviously, this is true if $i = 3$. Let us assume this is true for any $k < i$. As $a_i$ can be computed from the terms before, there exists a finite set $S_{i-k} \subseteq K$ for all $k \leq i$ and $b_j \in K$, such that $\sum_{j \leq i} b_{i-j} a_j = 0$. That is to say, for any critical pair $\{f_i, f_j\}$, the polynomial $\sum_{j \leq i} b_{i-j} a_j f_j$ is in $\mathcal{I}_k$. This leads to $f$ being spanned by polynomials $\sum_{j \leq i} b_{i-j} a_j f_j$ for all $k < i$. These polynomials form a Gröbner basis for $\langle f \rangle$ with a finite linear combination of these elements.

Conversely, let $\mathcal{G} = \{g_1, \ldots, g_n\}$ be a minimal reduced Gröbner basis of $I$ for a monomial ordering $\prec$. There exists a finite subset $S$ of finite support for all $i$, $1 \leq j \leq n$, $g_i = x^j \sum_{d \leq j} b_{d} x^d$ with $b_i \in K$. Let us prove we can set a finite number of terms of $u$ and then compute any term. Let $u_i$ be any term of the sequence. If $x^i$ is in the staircase of $\mathcal{G}$, then we set $u_i$. Otherwise there exist $j$ and $i$ such that $x^i \prec x^j$, hence $x^i g_j = x^i \sum_{d \leq j} b_{d} x^d \in I$. By recurrence on the $x^j$ and $x^i$, there exist $a_k \in K$ such that $x^j \sum_{d \leq j} b_{d} x^d \in I$. Therefore $u_i = \sum_{d \leq j} b_{d} x^d u_j$, and one can compute any $u_i$ from a finite number of terms.

In other words, if $u$ is linear recursive, then $K[x]/I$ is a finite dimensional $K$-vector space. This is related to the definition of holonomic function in an Ore algebra $\mathcal{O}(K)$ (see [11, Def. 1]). If $\text{Ann}(f)$ is the left ideal of polynomials vanishing on $f = \sum_{n \in \mathbb{N}} a_n x^n$, then $\mathcal{O}(\text{Ann}(f))$ is a finite dimensional vector space over $K$. This equivalent definition is also related to more restrictive than [3, Def. 2], in which the author only assumes that the ideal is not reduced to zero.

EXAMPLE 2. Consider the following sequences:
(a) $u_{ij} = \left(2^i + 3^j\right)$ for all $i, j \in \mathbb{N}$ is linear recursive of order 2, a minimal set of linear recurrence relations is $\{u_{i+2} - 4u_{i+1} + 4u_i = 0, u_{i+1} - 3u_i = 0\}$. The associated ideal $(x^2 - (x - 3), y - 3) = 2$ has two solutions of multiplicity 1.
(b) $u_{ij} = (i+1)^2 x^j$ for all $i, j \in \mathbb{N}$ is linear recursive of order 2, a minimal set of linear recurrence relations is $\{u_{i+2} - 4u_{i+1} + 4u_i = 0, u_{i+1} - 5u_i = 0\}$. The associated ideal $(x^2 - (x - 3), y - 5) = 2$ has one solution of multiplicity 2.
(c) $u_{ij} = i$ for all $i, j \in \mathbb{N}$ is not linear recursive, however it is holonomic. Indeed, while calling BMS or our algorithms on this sequence for all $u_{ij}$, $i + j \leq 2d$, one obtains relations $u_{i+1,j+1} = u_{i,j+1} - u_{i-1,j+1} = 0$, the famous Pascal’s triangle relation, together with $\sum_{j=0}^{d} \binom{d}{j}(-1)^{d-j} u_{i+j} = 0$ and $u_{i+1,j} = 0$. From the polyomino point of view, they form a d-truncated Gröbner basis of $I = \langle x, y - (x - 3) \rangle$.

DEFINITION 3. Let $u = \{u_{ij}\}_{i,j \in \mathbb{N}}$ be a recursive linear sequence and $I$ be its ideal of relations. The order of $u$ is equivalently the minimal number of initial terms of $u$ needed to compute any term or the degree of $I$.

2.2 Matrix multiplications in the quotient ring point of view

In this section, with a FGLM point of view, we shall show that given the ideal of relations and the initial terms of a sequence, one can express all the terms of said sequence as a scalar product. This point of view will be a key for the adaptive algorithm designed in Sec. 5.

Let $u = \{u_{ij}\}_{i,j \in \mathbb{N}}$ be a recursive linear sequence over $K$. Let $S$ be the staircase of a Gröbner basis of its ideal of relations $I$, then $K[x]/I$ is a $K$-algebra whose canonical basis as a vector space is made of the monomials $x^i \in S$. Defining $T_1, \ldots, T_n$ the multiplication matrices by respectively $x_1, \ldots, x_n$ in $K[x]/I$, then $u_i = \langle r^0, \ldots, r^{n-1} \rangle$, the vector of initial conditions, and $v = (1, 0, \ldots, 0)^T$, the vector representing 1 in the canonical basis of $\langle x_1, \ldots, x_n \rangle$.

Indeed, let $i \in \mathbb{N}$. By definition, $T^i \cdots T^2 \cdot 1$ is the vector representing $x^i$ in the canonical basis of $\langle x_1, \ldots, x_n \rangle$. Then, the scalar product allows one to map this vector to the corresponding linear combination of the initial terms of the sequence.

3. RANDOMIZED REDUCTION: BMS/BM

In this section, we exploit a randomized preprocessing on the table which can simplify the computation of the linear recurrence relations of the sequence. This preprocessing will yield a new table which, with good probability, will have one linear recurrence relation of the form $u_{i+d} - \sum_{j=0}^{d-1} a_{i+j} x_j = 0$, for all $i$ and other relations of the type $u_{i+k} = \sum_{j=0}^{k-1} b_{j} x_{i+k-j} = 0$. In other words, all the other variables can be deduced from the first one. Therefore, the bottleneck of the execution of BMS on this sequence would be the computation of the first relation. This comes down essentially to running BM on the subsequence $\{u_{i+1}\}_{i \in \mathbb{N}}$.

3.1 Linear Transformation of the Table

This preprocessing can be seen as a linear transformation on the variables appearing in the ideal of relations.

DEFINITION 4. Let $u = \{u_{ij}\}_{i,j \in \mathbb{N}}$ be a linear recurrent sequence. We define a change of basis on $u$ as an invertible matrix $A \in \text{GL}_n(\mathbb{R})$. The sequence $v = \{v_{ij}\}_{i,j \in \mathbb{N}} = A \cdot u$ is defined as $v_1 = [\text{Tr}(A)]u_1$.

In other words, for instance, $v_0 = u_0$, $v_1 = \sum_{j=0}^{n-1} a_j u_j$, etc. The following proposition links the ideals of relations of $u$ and $v$.

PROPOSITION 4. Let $P$ be a polynomial associated to a relation of $u$. Let $v = A \cdot u$ for $A$ invertible. Then $P(A^{-1} \cdot x)$ is a polynomial associated to a relation of $v$.

PROOF. Since $u_1$ is merely the polynomial $\langle A(x) \rangle$ evaluated in $u$, any polynomial $P(A^{-1} \cdot x)$ evaluated in $v$ will yield $P(x)$ evaluated in $u$. Therefore, $[P(A^{-1} \cdot x)]_v = 0$ iff $[P(x)]_u = 0$.

3.2 Essential reduction to BM

Let $u = \{u_{ij}\}_{i,j \in \mathbb{N}}$ be a $n$-dimensional linear recursive sequence. Let $v = \{v_{ij}\}_{i,j \in \mathbb{N}}$ be the $(n+1)$-dimensional sequence defined by $v_{ij} = u_i$ for all $i \in \mathbb{N}$, $j \in \mathbb{N}$. If $v \in K[x]$ is the ideal of relations of $u$ and is a new variable representing the last coordinate then $v = d + (r-1)$ is the ideal of relations of $v$. Let us now apply the change of coordinates in which each $x_i$ remains the same...
and j is mapped onto \( t + \sum_{k=1}^{n} c_k x_k \) for some \( c_k \in \mathbb{K} \). Generically, the minimal reduced Gröbner basis of the new ideal \( J' \) obtained from \( J \) for the LEX order with \( t < x_1 < \ldots < x_n \) is in shape position, i.e. is \( \langle f(t), x_1 - f(t), \ldots, x_n - f(t) \rangle \), with deg \( f(t) < \deg f_j \), see [6, 12]. As \( f(t) = \sum_{d \geq 0} a_d t^d \) depends only on \( t \), it is found by running BM on the subsequence \( (\nu_j)_{i \in \mathbb{N}} \). Each polynomial \( u - f_i(t) \), for \( 1 \leq i \leq n \), is then found by solving the linear system \( \sum_{j=0}^{d} B_j e_{d+j} = 0 \), whose matrix is Hankel.

Therefore, after applying a linear transformation on the table, finding its relations essentially reduces to running BM on a 1-dimensional subsequence. This is summed up in Alg. 1.

**Algorithm 1.** BM for n-dimensional sequences

**Input:** a n-dimensional sequence \( u = (u_i)_{i \in \mathbb{N}} \)

**Output:** an equivalent basis of relations of \( u \)

Pick at random \( c_1, \ldots, c_n \in \mathbb{K} \)

Compute \( \mathcal{V} = (\nu_j)_{j \in \mathbb{N}} \) with \( \nu_j = (t + c_1 x_1 + \ldots + c_n x_n) \)

Compute \( f(t) \) with BM on \( \mathcal{V} \) of degree \( d \).

For \( f \) from 1 to \( n \):

Solve the Hankel linear system

\[
\begin{pmatrix}
\alpha_{0,0} & \cdots & \alpha_{0,d-1} \\
\alpha_{1,0} & \cdots & \alpha_{1,d-1} \\
\vdots & \ddots & \vdots \\
\alpha_{n,0} & \cdots & \alpha_{n,d-1}
\end{pmatrix}
\begin{pmatrix}
\beta_{0,0} \\
\beta_{0,1} \\
\vdots \\
\beta_{0,n}
\end{pmatrix}
= \begin{pmatrix}
\alpha_{d,0} \\
\alpha_{d,1} \\
\vdots \\
\alpha_{d,n}
\end{pmatrix}
\]

Return \( f_j \) as \( \mathcal{V} \) of degree \( d \).

This section is devoted to the design of a FGLM-like algorithm for computing the Gröbner bases of the ideal of relations of a sequence \( u = (u_i)_{i \in \mathbb{N}} \), with \( i \leq n \). From now on, \( \mathcal{T} \) is the ordered set of terms that we can make from \( x_1, \ldots, x_n \). We fix \( \mathcal{T} \) to be an admissible monomial ordering. If \( P \subseteq \mathbb{K}[x_1, \ldots, x_n] \) then \( \mathcal{T}(P) \) is the set of terms composing \( P \) and \( \mathcal{T}(P) \) is the maximum of \( \mathcal{T}(P) \).

For any \( D, \mathcal{T}_D \) is the set of all terms of degree \( \leq D \) sorted by increasing order (wrt. \( \prec \)). Since we want to design an algorithm we will be unable to check that a relation is valid for all \( i \in \mathbb{N} \). Indeed, at some point of the algorithm we will have a finite subset of indices \( T \subseteq \mathbb{N} \) and we will try to find relations that are valid for those indices: for \( i \in T \), \( u_i + d + \sum_{k \in \mathbb{N}} a_k x_k u_k = 0 \).

**Definition 5.** Let \( T \) be a finite subset of \( \mathbb{N} \). We say that a polynomial \( P \in \mathbb{K}[x_1, \ldots, x_n] \) is valid up to \( T \) if \( \text{rel}_T(P)[i] = 0 \) for all \( i \in T \).

Let \( T \) be a finite subset of \( \mathcal{T} \). We say that a polynomial \( P \in \mathbb{K}[x_1, \ldots, x_n] \) is valid up to \( T \) if \( [P] = 0 \) for all \( t \in T \). In that case we write that \( \mathcal{N}(P, \mathcal{T}, T) = 0 \).

### 4.1 Staircase

We assume now that \( T \subseteq \mathcal{T} \) is a finite set of terms. Equivalently, \( T' \) is the set of exponents of all \( t \in T \). Any BMS-style algorithm will generate minimal relations; hence when we try to establish a new relation \( u_i + d + \sum_{k \in \mathbb{N}} a_k u_k = 0 \) we must check two properties:

1. \( a_k \) are scalars \( a_k \in \mathbb{E} \) so that \( \forall i \in T', u_i + d + \sum_{k \in \mathbb{N}} a_k u_k = 0 \);
2. \( b \) are no nonzero relations \( \sum_{k \in \mathbb{N}} \beta_k u_k = 0 \) which are valid for all \( i \in T' \).

We can translate these properties in polynomial terms:

1. There is a monic polynomial \( P \in \mathbb{K}[x_1, \ldots, x_n] \) of leading term \( x^{d_1} \) s.t. \( \mathcal{N}(P, \mathcal{T}, T) = 0 \);
2. There are no nonzero relations \( \sum_{i \in \mathcal{T}} \beta_i \mathcal{N}(i)(1) = 0 \) which are valid for all \( i \in T' \).

Hence it is important to identify a set of terms for which there is no linear relations.

**Definition 6.** Let \( T \) be a finite subset of \( \mathcal{T} \). We say that a finite set \( S \subseteq \mathcal{T} \) of terms is a useful staircase \( u \) if \( \mathcal{T}', T \) and \( S \) are\( \mathcal{N}(P, \mathcal{T}, T) = 0 \).

### 4.2 Linear Algebra to find relations

We give a simple algorithm to check that a finite set \( S \subseteq \mathcal{T} \) is a useful staircase \( u \) if \( \mathcal{T}', T \) and \( S \) are. Let us start with a simple example:

**Example 4.** \( \text{Ex. 3 cont.} \). We look for a relation \( P(x, y) = a x^3 + a x y^2 + a_3 + a_1 x + a_0 y + d_0 \) that is to say we try to find \( a_0, \ldots, a_{s+1} \) s.t. \( [P] = 0 \) for all \( t \in T : a_0 u_{k+1} + a_1 u_{k+2} + a_3 u_{k+2} + a_2 u_{k+1} + a_0 u_{k+1} = 0 \).\( t \) is a useful staircase. Since 1 is not in this set, it is not stable under division. However, using the stability criterion, we recover the complete staircase \( \{x, y, x^2, y^2\} \).

Alg. 2 transforms a useful staircase into a staircase.

**Algorithm 2. Stabilize**

**Input:** \( S \) a useful staircase

**Output:** a staircase

For \( \forall s \in S \) do \( S := S \cup \{ t \mid t \in \mathcal{T} \text{ and } t \text{ divides } s \} \)

Return \( S \)
or equivalently

\[
H_T = \begin{pmatrix}
1 & y & x & y^2 & xy & x^2 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

In this example, 1 and \(y^2\) are clearly useless so that \(S' = \{y, x, yx, x^2\}\) is the useful staircase and \(\det(H_T) = 1 \neq 0\). In addition we can try to find a relation \(Q(x, y) = x^2 - ax^2 - ay(x) - a_1 x - ay_y\). Again this is equivalent to finding \(a_0 \ldots a_3\) s.t. \(\text{rel}(Q)(i_1, i_2) = 0\) for all \((i_1, i_2)\) with \(i_1 + i_2 \leq 2\). In turn, this reduces to solving the linear system:

\[
H_S \cdot [a_0 \ldots a_2] = \begin{pmatrix} a_0 \ldots a_3 \end{pmatrix} = 0.
\]

Since \(H_T\) is full rank we find \(a_0 = \ldots = a_3 = 0\) and the relation \(Q(x, y) = x^2\).

In the general case we can now define the structured matrix associated to two lists of terms:

**Definition 7.** Let \(T\) and \(S\) be two finite subsets of \(\mathcal{T}\). We consider the polynomial \(P_S(x) = \sum_{\tau \in S} \tau a_{\tau}\) and the linear equations \(H_T S = 0\) from the previous linear system of equations in the unknown variables \(a_{\tau}\) for \(\tau \in S\): \[\vdots \quad \vdots \quad \vdots \quad \vdots \]

\[
H_T S = \begin{pmatrix} t \vdots \vdots \vdots \end{pmatrix}_u = \begin{pmatrix} 0 \end{pmatrix}.
\]

When \(T = S\) we simply write \(H_T\) for the multi-Hankel matrix \(H_T S\). These two crucial operations involve linear algebra:

(a) Checking that a finite set of terms \(S \subseteq \mathcal{T}\) is a useful staircase wrt. \(T\) (we assume that \#\(T\) \(\geq\) \#\(S\)) is equivalent to checking that the matrix \(H_T S\) has full rank;

(b) Finding a monic polynomial \(P \in \mathbb{K}[x_1, \ldots, x_n]\) of given support s.t. \(\text{NF}(P, u, T) = 0\) is equivalent to solving a linear system \(H_T S \cdot a = \text{NF}(P, u, T) = 0\)

where \(S = \mathcal{T}(P - \text{LT}(P))\) is the support of the polynomial \(P\) except the leading term. If \(a\) is a solution then \(P = \text{LT}(P) + \sum_{\tau \in S} \tau a_{\tau}\) is a polynomial s.t. \(\text{NF}(P, u, T) = 0\).

**Proposition 6.** Let \(T\) be a finite subset of \(\mathcal{T}\). If the finite set of terms \(S \subseteq \mathcal{T}\) is a useful staircase wrt. \(u, T\) and \(<\) then:

\[
\text{det}(H_S) \neq 0 \text{ and rank } H_S = \text{rank } H_T = \text{rank } H_T S.
\]

**Proof.** This is another wording of Def. 6.

The two statements of Prop. 7 are easy to prove but they are the basis of the algorithm: according to them we know that we can proceed degree by degree.

**Proposition 7.** If \(S' \subseteq S\) are finite subsets of terms, then \(\text{rank } H_S \geq \text{rank } H_{S'}\). Moreover, assume \(S\) is a useful staircase s.t. \(\det(H_T S) \neq 0\) and \(t \in \mathcal{T} \setminus S\) then we have: \(\text{rank } H_{S \cup \{t\}} = \text{ColRank } H_{S \cup \{t\}} = \text{rowRank } H_{S \cup \{t\}}\).

**Proof.** \(H_S\) is a submatrix of \(H_T\) and \(H_T S\) is symmetric.

We rely on the following naive strategy to extract a full rank matrix of a multi-Hankel matrix. We start with \(H_{\{1\}}\) and we proceed by induction, assuming that at some point we have found a useful \(S\) s.t. \(H_S\) is full rank. Then we select the minimal \(t \in \mathcal{T} \setminus S\). If \(\text{rank } H_{S \cup \tau} = \text{rank } H_T - 1\) then we update \(S := S \cup \{t\}\); else we have \(\text{rank } H_{S \cup \{t\}} = \text{rank } H_T\) and we consider \(\mathcal{T}(S \cup \{t\})\).

Given a useful staircase \(S\), it is important that \(2S = \{s, s, t \in S\}\) be not too big compared to \(S\), when counting the number of table queries. We will see how to bound the cardinality of \(2S\) in Sec. 5.1.

### 4.3 An FGLM-like Algorithm

Since the input table \(u\) is infinite we need a bound given by the user: \(d \geq 0\) and \(T\) will be the set of all monomials of degree less than \(d\). Accordingly, we will assume that the monomial ordering \(<\) is an admissible ordering refined by the total degree. Since the output of the BMS algorithm is a (truncated) Gröbner basis, it is a natural idea to try to adapt existing Gröbner basis algorithms to obtain the same result. To this end, we can try to slightly modify the FGLM algorithm. However, in the scalar case, a fundamental difference is that the structure of the quotient ring (and in particular the staircase) is not known. Hence, to clarify our intention we will split the algorithm in two parts: first we derive the staircase wrt. the monomial ordering and the given bound; in a second step we compute a truncated Gröbner basis. In a real implementation, the two steps can be combined to increase the efficiency of the algorithm.

**Algorithm 3.** FGLM for scalars.

**Input:** \(<\) a monomial ordering, \(u = (\beta_i)_{i \in \mathcal{N}} \) a sequence with coefficients in \(\mathbb{K}\), a given bound.

**Output:** a reduced \((d + 1)\)-truncated Gröbner basis wrt. \(<\) of \(u\).

**Build the matrix \(H_{\mathcal{T}}\):**

**Find \(S\) the useful staircase s.t.** \(\text{rank } H_{\mathcal{T} S} = \text{rank } H_{\mathcal{T}}\). // as in Sec. 4.2

**Sort \(L\) by increasing order (wrt. \(<\)) and remove multiples of \(\text{LT}(G)\).**

**Return \(G\)**

**Example 5 (CONT. OF EX. 2.C).** We consider the table \(u\) generated by \(u_{i,j} = 0\) and we fix \(d = 2\). We consider the matrix

\[
H = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

It is easy to check that the column (resp. row) \(x\) is the sum of the first two columns (resp. rows). Hence, the useful staircase is \(S' = \{1, x, x^2, x^3\}\). Since \(S'\) is stable under division, \(S' = S\). We initialize \(L = [x, x^2, y, x^2 y, y^3]\) so that \(\mathcal{T} = \{s\}\). The next step is to solve the system \(H_T \cdot x = [1, 0, 2, 0, 3]^T = 0\) and we find \(x = [-1, -1, 0, 0, 0]^T\) so that \(G = [y, x - y]^T\). We can update \(L = [x^2, y^3]\) and by solving two other linear systems, we find \(G = [x y - y, -1, x^3 - 3 x^2 + 3 x - 1 = (x - 1)^3, y^3]\). Clearly this is not a full Gröbner basis but a 3-truncated Gröbner basis.

**Theorem 8.** The output of Alg. 3 is a \((d + 1)\)-truncated Gröbner basis. Moreover, if \(u\) is a recursive linear sequence of order \(D\), taking \(d = D\) suffices to recover a full Gröbner basis of the sequence.

**Proof.** Alg. 3 clearly terminates since the size of \(L\) decreases.

Taking the basis monomials in increasing order ensures that the set \(S'\) contains monomials of smallest degrees. Let us first show that the staircase of the ideal of \(u\) contains the useful staircase. In the following, we shall see polynomials as linear combinations of columns of \(H_{\mathcal{T} S}\). Any polynomial with leading term outside the staircase of the ideal of \(u\) reduces to polynomials in the staircase. Suppose that one element \(e\) of the useful staircase lies outside the staircase. Seen as linear combinations of columns of \(H_{\mathcal{T} S}\), it is then a linear combination of elements in the staircase, some of which are not in the useful staircase by linear independence. Let \(f\) be one of these particular elements: \(f\) is a linear combination of smaller elements in the useful staircase. Then \(e\) is actually a linear combination of elements in the useful staircase, which is contradictory.

Conversely, if \(S'\) does not contain a maximal (for the natural order on the table) element of the staircase for \(u\), then this element can be written as a linear combination of smaller terms, which contradicts the fact it belongs to the staircase. The stabilization of these maximal elements is therefore the full staircase of the ideal of \(u\).
The set $G$ contains elements with leading terms that do not divide each other. Let us consider $f$ and $g$ in $G$ (with leading terms in $\mathbb{S}_{d+1}$) and their $S$-polynomials $S(f,g)$. Then either the leading term of $S(f,g)$ is in $\mathbb{S}_{d+1}\setminus S'$ or it is in $S'$. In the latter case, it means there is a relation in $HS$, so it cannot be a new relation. In the former case, the relation was already found by the main loop. So no $S(f,g)$ produces a new relation.

5. ADAPTIVE ALGORITHM

So far we have seen two algorithms to recover the relations from a table: these algorithms are efficient when the degree of the elements in the Gröbner basis $G$ is small compared with the order of the recurrence: $\max \deg G \approx (\text{order of the recurrence})^{1/2}

Unfortunately, this is not always the case, especially if the monomial ordering is a lexicographical order. Using the previous algorithms on these examples would increase too much the complexity (by complexity we mean the number of accesses to the table $u$). The goal of this section is to describe an adaptive algorithm to take into account the shape of the final Gröbner basis.

The main difference between Alg. 3 and the original FGLM is the following: with polynomials, when we discover a relation $f = t + \sum_{s \in G} G_s$ we know that $m|f$ is still a valid relation for any $m \in T$. In contrast, when we find a relation

$$[f]u = [m]u + \sum_{s \in G} G_s(x) u = 0 \quad \text{(1)}$$

it is not in general that $[m]u = 0$. However we know from Sec. 2.2 that any $n$-dimensional linear recurrence can be written as

$$u_0 = (r, T_1, \ldots, T_n \cdot 1)$$

where $r$ is a vector depending on the initial conditions and $T_i$ are multiplication matrices associated to the Gröbner basis $G$. Therefore we can write $(r, \text{NormalForm}(f, G)) = 0$ for relation (1). Hence, if $r$ is sufficiently random we know that $\text{NF}(f,G) = 0$ so that $\text{NF}(m,f,G) = 0$ for all $m \in T$ which implies that $[m]u = 0$. Note that in some applications (see for instance Sec. 5.2) it is possible to check afterwards that the relation is correct. Accordingly, we propose an FGLM algorithm to take advantage of this property. We proceed term by term and try to discover the new staircase which is equivalent to increasing the rank of the multi-Hankel matrix by 1. We do not give any bounds on the degree of the output polynomial but we assume that an estimate of the size of order of the recurrence is given by the user. We will see later that for many applications the complexity can be reduced drastically; depending on the shape (for instance the convexity) of the final staircase, the number of queries to the table can often be linear in the order of the recurrence, similarly to the one-dimensional case. In the following algorithm, for any list of terms $G$, $\text{MinBase}(G)$ is the corresponding minimal Gröbner basis.

**Algorithm 4. Adaptive FGLM for scalars (simple version)**

**Input:** $\prec$ a monomial ordering, $u = (u_0, \ldots, u_n) \in \mathbb{C}^{n+1}$ a sequence with coefficients in $\mathbb{C}$, $d$ is a given bound.

$L_0 := \{1\} \quad $// list of next terms to study

$S := \{\} \quad $// the useful staircase wrt. the new ordering $\prec$

$G := \{\} \quad $// leading terms of the final Gröbner basis

**While** $L \neq \emptyset$

$t := \text{first}(L)$

If $H_{G(t)}$ is full rank then

$S := S \cup \{t\}$ and $L := L \cup \{x_i \mid i = 1, \ldots, n\} \setminus \{t\}$

Sort $L$ (wrt. $\prec$) and remove duplicates and multiples of $G$

If $\#S \geq d$ then $\quad$ // early termination

$G := \bigcup S$ and $G := \text{MinBase}(G \cup L \cup \mathbb{S}_{d+1}(\text{deg}))$

For $r \in G$

$G := G \cup [S_{r_{1}}, S_{r_{2}}, \ldots, S_{r_{n}}]$ where $x$ s.t. $H_{k}x + H_{G(t)}x = 0$

Return $S$ and $G$

Else $G := G \cup [\{\} \setminus \{t\}]$ and remove multiples of $tL$

**Error** "Run Alg 3."

**Remark.** In some applications (see e.g. Sec. 5.3 on error correcting codes), the input table is bounded: $u_{i_{1}}, \ldots, u_{i_{k}}$ cannot be computed or has no meaning when $i_{j} > B$ for some bound $B$. One can easily modify the algorithm to take this constraint into consideration.

**Proposition 10.** Let $S$ and $G$ be the output of Alg. 4. Then $S$ is a staircase of size $\geq d$ and $G$ is a list of valid relations, that is to say $\text{NF}(f, u, S) = 0$ for all $f \in G$.

**Example 6.** As explained in Sec. 5.2, we consider the ideal of the vanishing ideal of the points $[0.0, [0.1, [1.1, [1.2]]]$. We compute a total degree Gröbner basis in $\mathbb{T}_1[x_1, x_2]$ and we apply the Sparse-FGLM algorithm with a random vector $r = [10, 3, 5]$. Hence, we ran Alg. 4 with $d = 3$ on the table $u = \left(\begin{array}{c}10 & 13 & \cdots \\ 3 & 5 & 5 \end{array}\right)$. Step 1: $t = 1$: we check $H_{[1]} = (10)$ has rank 1 so that $L = [x_2, x_1]$. Step 2: $t = x_2$: we compute the rank of $H_{[1, x_2]} = \left(\begin{array}{c}10 & 3 & 3 \\ 5 & 5 & 5 \end{array}\right)$ which is equal to 2. We update $L = [x_2, x_1, x_1, x_2]$ and $S = [x_1, x_2]$. Step 3: $t = x_2^2$ but obviously the rank of $H_{[1, x_2, x_2]} = \left(\begin{array}{c}10 & 3 & 3 \\ 3 & 3 & 3 \end{array}\right)$ is always 2. We set $G' = [x_2]$. Step 4: $t = x_1$: we compute the rank of $H_3 = H_{[1, x_2, x_1]} = \left(\begin{array}{c}10 & 3 & 5 \\ 3 & 5 & 5 \end{array}\right)$ which is equal to 3. Now $S = [x_2, x_1, x_2]$ and $L = [x_1, x_2, x_2]$. Since $\#S \geq d$ we can stop the algorithm so that $G = \text{MinBase}([x_2, x_1, x_2, x_2]) = [x_2, x_1, x_2, x_2, x_1]$. Lastly, we solve the 3 linear systems $H_3x = u$ with

$$u = H_{[1]} = \left(\begin{array}{c}1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{array}\right)$$

and we obtain $G_{\text{lex}} = [x_2^3 - x_2 x_1 x_2 x_1 - x_1 x_2 - x_2^3 - x_1 x_2^2]$. It is easy to compute the solutions $\{[0, 0], [0, 1], [1, 1]\}$.

5.1 Relation between the number of table queries and the geometry of the final basis

To estimate the complexity of the algorithms we have to bound the number of table queries and the complexity of the linear algebra part (this issue is addressed in Sec. 6). Indeed, in some applications computing one element $u_{i_1,i_2,\ldots}$ of the table is very costly (see for instance Sec. 5.2) and it is important to minimize the number of queries. Estimating this number is equivalent to counting the number of distinct elements in $H_3$ where $S$ can be any value of the variable in Alg. 4. We denote by $S$ the value at the end of the algorithm. Similarly to the original FGLM algorithm we can bound the number of monomials $t$ that we have to consider using $\#L \leq n\#S$. Hence it is crucial to bound the number of elements in $H_3$ where $S$ is the final staircase. Restating Th. 2, the necessary number of queries to $u$ to build $H_3$ is the cardinal of $2S = \{(u,v) \mid (u,v) \in S^2\}$ the dilated set of $S$.

It is clear that $\#(2S) = \#S(\#S - 1)/2 \leq (\#S)^2/2$ in the worst case: in many applications we have $\#(2S) \leq c\#S$ for some constant $c$. According to [16, Th. 1.1], $S$ verifying this condition must be included in a bigger set whose elements are in arithmetical progression of dimension $d$ and of size $C\#S$ for some constant $C$. In other words, $S$ must be included in a $d$-dimensional parallelepiped where the number of points is $C\#S$.

**Proposition 11.** Depending on the shape of the final Gröbner basis $G$, we estimate $\#(2S)$ when $d \rightarrow \infty$:

a. (1-dimensional case – BM) $n = 1$, $S_d = \{1, x, x^2, \ldots, x^{d-1}\}$ then $\#(2S_d) = 2d + 1$ and $\#(2S_d)/\#S_d \approx 2$;

b. (Worst 1-dimensional case) $n = 1$, $S_2 = \{1, x^2, x^4, x^{2d+1}\}$ then $\#(2S_2) = \frac{2}{d} d + 1$ and $\#(2S_2)/\#S_2 \approx \frac{d}{2}$;

c. (Alg. 3) $S_d = \{x \in \mathbb{T} \mid \deg (x) \leq d\}$ then $\#(2S_d) = \frac{d}{2} d + 1$ and $\#(2S_d)/\#S_d \approx 2^{d}$;

d. (Shape position) When $G = [x_1 - h_{i}, \ldots, x_n]$ then $S_d = \{1, x_1, \ldots, x_n\}$ where $d = \deg (h)$. Again $\#(2S_d)/\#S_d \approx 2$;
e. (Worst case in dimension $n$) $S_d = \bigcup_{i=1}^{n} \{x_1, x_2, \ldots, x_d | x_i \in \mathbb{R} \}$ then
\[
\frac{\#(2S)}{\#S_d} \approx \frac{1}{2} \frac{n}{\#S_d}.
\]

Figure 1: Behavior of $\#(2S)$ wrt. $D = \#S$ (the area in blue)

PROOF. (a) Clearly $2S_0 = S_{2d−1}$. (b) $S_{2d} = S_d \cup \{x^{2d−1} \mid i \neq j \} \cup \{x^{2i−1} \}$. Note that $S_d$ is not stable under division in that case.

(c) Noticing $S_{2d} = S_d$ and $\#S_d = \left(\binom{d+n}{n} \right)$, we have $\frac{\#(2S)}{\#S_d} = 2^n - 2^{n−1} \left(\frac{n+1}{2}\right)$.

(d) Same as item a.

5.2 Application to the Sparse-FGLM algorithm

The Sparse-FGLM [5] is a natural application of the previous algorithm: for a 0-dimensional polynomial system we compute a Gröbner basis (most of the time wrt. a total degree ordering). Then, we compute the $D \times D$ multiplication matrices $T_i$ wrt. the variable $x_i$ for all $i \in \{1, \ldots, n\}$. We consider the table $\mathbf{u}_i = (r, T_{i1} \cdots T_{in})$ where $r$ is a random vector and $1 = [1, 0, \ldots, 0]^T$. The computation of one element of the table from the previous ones can be reduced to one matrix-vector multiplication.

REMARK 12. Assuming that we store the vectors $V_i = T_i \cdots T_{n1}$ for the visited indices $i$, any relation $g = \sum_{i,j}s_{ij}V_i \in G$ computed by the algorithm can be easily checked if $\sum_{i} s_{ij} \mathbf{u}_i = 0$ then we have a proof that $g \in 1$. Note, that in addition, we know precisely the bound $d$ since it is the number of solutions (with multiplicities). Hence it is always possible to check the correctness of Alg. 4.

Even if the sparsity of the multiplication matrices can be used to speed up the computation, it is important not to precompute all the elements of the table in advance. Hence a black-box representation is recommended. As shown in [5], when the lexicographical basis is in shape position, the Gröbner basis can be computed very efficiently; in particular, the number of table queries is $2D$, in this situation we can also use the change of variables designed in Sec. 3 to compute the Gröbner basis. This is why, in the experiments of the following paragraphs, we consider examples which are far from the shape position and we compute the LEX basis.

Cyclic-$n$ problem. This is a well known benchmark; there are $n$ equations in $n$ variables, the $i$th equation is of degree $i$ and is invariant by the action of the $n$th Cyclic group; since there is a linear equation, the actual number of variables is $n − 1$. We report in Tab. 5.2, the number of rank computations and the normalized number of table queries (the number divided by the number of solutions). This number is always less than $2^n−1$.

<table>
<thead>
<tr>
<th>Example</th>
<th>$n$</th>
<th>$D$</th>
<th>Nb. Ranks</th>
<th>#Queries/D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclic 5</td>
<td>5</td>
<td>70</td>
<td>70</td>
<td>9.4</td>
</tr>
<tr>
<td>Cyclic 6</td>
<td>6</td>
<td>156</td>
<td>167</td>
<td>9.4</td>
</tr>
<tr>
<td>Cyclic 7</td>
<td>7</td>
<td>924</td>
<td>953</td>
<td>21.7</td>
</tr>
</tbody>
</table>

Ideal of points. Given a set $P \subseteq \mathbb{C}$ of $t$ distinct points, we define the ideal $I_P = \{ f \in \mathbb{R}[X_1, \ldots, X_n] \mid f(p) = 0 \forall p \in P \}$. We consider two such sets.

a. (Random) For any integer $B$, we generate exactly $t$ points in $P_B \subseteq \mathbb{C}$ with coordinates randomly chosen in $\{0, \ldots, B−1\}$. Since $B$ is a bound on the degree of the univariate polynomial in the LEX Gröbner basis, this basis is far from the shape position when $t \gg B$.

b. (Worst Case) $R = \{x_j, 1 \leq i \leq n, 1 \leq j \leq t/n\}$. In both cases we report the ratio between the number of queries and the number of points. As expected in the first case, this ratio is a constant $c \in [2, 2^n]$ depending on the value of $B$. In the second case, we expect a linear behavior, from Prop. 11. In Fig. 5.2 the points below the thick dashed black line correspond to Gröbner bases in shape positions.

5.3 Application to error correcting codes

In Coding Theory, $n$-dimensional cyclic codes with $n−1$ are generalizations of Reed Solomon codes. We give a simplified description of such codes. Let $f$ be an integer and $a \in \mathbb{F}_p$ such that $a^f \neq 0$ for $0 < j < p−1$. We work with polynomials in $R = \mathbb{F}_p[x]/(x^p−1−1, x^{p^{n−1}}−1)$. Then we define the generating polynomials $g(x) = \prod_{j=0}^{p−1} (x−a^j)$. When we send a message $\mathbf{m}$ we split this message into $n$ blocks $\mathbf{m}(k) = (c_0^{(k)}, c_1^{(k)}, \ldots)$ where $c_j \in \mathbb{F}_p$ and we generate $n$ multivariate polynomials $U_j(x) = c_0^{(k)} + c_1^{(k)}x + c_2^{(k)}x^2 + \cdots$. The transmitter sends the encoded message $\mathbf{M}(x) = \sum_{j=0}^{p−1} \mathbf{U}(x)$. The receiver interprets the received word as a multivariate polynomial $\mathbf{N}(x) = \mathbf{M}(x) + e(x)$ where $e(x) \in R$ is the error polynomial. If the length of $e(x)$ is less than $t = \frac{q}{f}$ the goal is to recover it. To do this, we build the table $\mathbf{u}_{i−j} := N(x^i, \ldots, x^j) = e(x^i, \ldots, x^j) \in R$ for $0 \leq i < j < t$ and we apply Alg. 4 to obtain a LEX Gröbner basis $G$. It is easy to recover all the solutions in the finite field $\mathbb{F}_p$, next, by computing the discrete logarithm wrt. $a$ of all the components we recover the position of the nonzero monomials in $e(x)$. Lastly, we solve a linear system to find the coefficients of $e(x)$.

In the experiments of Tab. 5.3, we consider two cases: (random case) we randomly generate the support and the coefficients of the error polynomial $e(x)$; (worst case) we take $e(x) = \sum_{j=0}^{p−1} \mathbf{u}_{i−j} x^j$.

6. MULTIBLOCK HANKEL ARITHMETIC

In Alg 3 and 4, linear systems must be solved. In this Section, we show that in fact, if $f$ is a LEX order, then the matrices are heavily structured as they are Hankel matrices. Let’s recall that a Gröbner basis of a 0-dimensional ideal for LEX order on $x_1, \ldots, x_n$ with $x_1 < \cdots < x_n$ is given in terms of non-constant polynomials $P_{i,j}(x_j)$ and $P_{i,j}(x_j, \ldots, x_n)$ for $i > 1$, with deg$_x P_{i,j} \leq d_i$.

When computing $P_{i,j}$, one will only consider sets of monomials $S = \{x_{n−1}\}$. Therefore, the matrix $\mathbf{H}_S$ is Hankel. When looking for $P_{i,j}, \ldots, \mathbf{P}_{j,m}$, one needs to consider sets of monomials $S' = S \cup x_{j+1} \cup \cdots \cup x_{n−1}$ with $x_j, \ldots, x_{n−1} \subseteq S$. This yields
the following matrix

\[ H^r = \begin{pmatrix}
H_{i} & H_{i+S_{1}} & \cdots & H_{i+S_{d-1}} \\
H_{i+S_{1}} & H_{i+2S_{1}} & \cdots & H_{i+S_{d-1}} \\
\vdots & \vdots & \ddots & \vdots \\
H_{i+S_{d-1}} & H_{i+2S_{d-1}} & \cdots & H_{i}
\end{pmatrix}
\]

where each \( H_{i+S_{j}} \) is Hankel rectangular. Completing \( H^r \) so that each block is square, i.e. replacing each \( H_{i+S_{j}} \) by \( H_{i,S_{j}} \) makes it block Hankel with Hankel blocks. We shall say that \( H^r \) is multi-block Hankel of depth 2. That is, they will have the same shape as \( H \). The matrix

\[ H \]

by row or column of size \( d \) can once again apply the displacement operator of Hankel matrices of size of the blocks. Because our blocks are themselves Hankel, we shall be called multiblock Hankel of depth 2. This definition extends to all \( n \in \mathbb{N} \), with depth 1 being classical Hankel matrices.

6.1 Displacement rank

We recall that for a matrix \( H \), a displacement operator \( \varphi \) is an operator s.t. \( \varphi(H) \) has small rank. One can classically take, for \( H \) Hankel, \( \varphi(H) = H - ZH^tZ \) with \( Z = (\delta_{i,j})_{1 \leq i,j \leq n} \), where \( \delta_{i,j} \) denote Kronecker’s delta function. Indeed, \( \varphi(H) \) has rank at most 2.

This nice structure allows us to solve a linear system with a Hankel-like matrix, i.e. a small sum of Hankel matrices, in \( O(\alpha^{d-1} \cdot M(d) \log d) \) operations if \( \alpha \) = rank \( \varphi(H) \) and if \( d \) is the size of \( H \), see [2].

On block Hankel matrices, one can take the deflated operator, in which each 1 of \( Z \) is replaced by an identity matrix of the right size. However, the expected matrix should have rank twice as much as the size of the blocks. Because our blocks are themselves Hankel, we can once again apply the displacement operator of Hankel matrices on all remaining blocks. If \( H \) is Hankel block Hankel with \( d_2 \) blocks by row or column of size \( d_1 \), then the obtained matrix has rank at most \( 2 \min\{d_1, d_2\} \).

Consequently, with multiblock Hankel matrices of depth \( n \) and embedding blocks of sizes \( d_1, \ldots, d_n \), one can find a displacement operator s.t. the displacement rank is at most \( 2 \sum_{i=1}^{n} (d_i / \gcd_{j=1}^{n} d_j) \).

These displacement ranks are not too small, unless e.g. all the \( d_i \)’s stay constant but one that grows to infinity.

6.2 Polynomial interpretation

Fast algorithms on solving Hankel linear systems are coming from the fact that multiplying a Hankel matrix of size \( d \times d \) with a vector can be seen as computing the middle product of univariate polynomials of sizes \( 2d \) and \( d \). Solving such a system comes down to dividing a polynomial of sizes \( 2d \) by a polynomial of size \( d \), which can be done in \( O(M(d^2)) \) operations in the base field. For Hankel block Hankel linear systems with \( d_2 \) blocks of size \( d_1 \), the matrix-vector product can be seen as a generalization of the middle product of two bivariate polynomials, both of degree \( d_1 - 1 \) in the first variable and one of degree \( d_2 - 1 \) and the other \( 2d_2 - 1 \) in the second variable. By Kronecker’s trick, the complexity of solving such a system is \( O(M(2d_1d_2)) \). For multiblock Hankel of depth \( n \) system, this strategy yields a complexity in \( O(M(2^{n-1}d_1 \cdots d_n)) \).

6.3 Complexity comparisons

Let us recall that \( d \) is the order of the recursive sequence \( u \): it is the size of the staircase of any Gröbner basis of its ideal of relations. Let also \( \mu \) denote the size of the computed Gröbner basis. In [21], the complexity of \( \text{BMS} \) is given as \( O(\alpha d^2) \) and estimated as \( O(d^3) \) with the approximation \( \alpha \sim O(d) \). Let us remark that the only proven bound is \( \mu \leq nd \) [4, Cor. 2.1] making the complexity of \( \text{BMS} \) in \( O(n^d) \) operations in \( \mathbb{K} \).

In the shape position situation, \( d_1 = d, d_2 = \cdots = d_n = 1 \); in the worst-case scenario, the staircase is a simplex with \( d_1 = \cdots = d_n \) and \( d_1 \cdots d_n = n! d \). Our complexity estimate becomes resp. \( O(M(2^n d)) \) or \( O(M(2^n n^d)) \). Both are quasi-linear in \( d \) if \( n \) is fixed.

We cannot say if one of our two algorithms could be seen as a matrix version of \( \text{BMS} \) – in which case, we would improve the complexity estimate of \( \text{BMS} \). Finding loop invariants in these algorithms could also help make their complexities sharper and find optimal termination criteria, hence reduce the number of table queries.

Acknowledgements

We thank the anonymous referees for their careful reading and their helpful comments and Erich L. Kaltofen for valuable discussions. This work has been partly supported by the French National Research Agency ANR-11-BST02-0013 HPAC project.

7. REFERENCES


