Linear Algebra for Computing Gröbner Bases of Linear Recursive Multidimensional Sequences

Jérémy Berthomieu, Brice Boyer, Jean-Charles Faugère

Linear Algebra for Computing Gröbner Bases of Linear Recursive Multidimensional Sequences

Jérémy Berthomieu*, Brice Boyer, Jean-Charles Faugère

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

The so-called Berlekamp – Massey – Sakata algorithm computes a Gröbner basis of a 0-dimensional ideal of relations satisfied by an input table. It extends the Berlekamp – Massey algorithm to n-dimensional tables, for $n > 1$.

We investigate this problem and design several algorithms for computing such a Gröbner basis of an ideal of relations using linear algebra techniques. The first one performs a lot of table queries and is analogous to a change of variables on the ideal of relations.

As each query to the table can be expensive, we design a second algorithm requiring fewer queries, in general. This FGLM-like algorithm allows us to compute the relations of the table by extracting a full rank submatrix of a *multi-Hankel* matrix (a multivariate generalization of Hankel matrices).

Under some additional assumptions, we make a third, adaptive, algorithm and reduce further the number of table queries. Then, we relate the number of queries of this third algorithm to the *geometry* of the final staircase and we show that it is essentially linear in the size of the output when the staircase is convex. As a direct application to this, we decode n-cyclic codes, a generalization in dimension n of Reed Solomon codes.

We show that the multi-Hankel matrices are heavily structured when using the LEX ordering and that we can speed up the computations using fast algorithms for quasi-Hankel matrices. Finally, we design algorithms for computing the generating series of a linear recursive table.

*Laboratoire d'Informatique de Paris 6, Université Pierre-et-Marie-Curie, boîte courrier 169, 4 place Jussieu, F-75252 Paris Cedex 05, France.

Email addresses: `jeremy.berthomieu@lip6.fr` (Jérémy Berthomieu), `brice.boyer@lip6.fr` (Brice Boyer), `jean-charles.faugere@inria.fr` (Jean-Charles Faugère)

Keywords: The BMS algorithm, the FGLM algorithm, 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 with constant coefficients satisfied by S or the length of the shortest linear feedback shift register (LFSR) which generates it.

The Berlekamp – Massey algorithm (BM, Berlekamp (1968); Massey (1969)) guesses a solution of this problem for sequences with one parameter, i.e. in the one-dimensional case. For n-dimensional sequences, the definition of a recurrence relation with constant coefficients was proposed by several authors Chabanne and Norton (1992); Fitzpatrick and Norton (1990); Saints and Heegard (1995); Sakata (1988, 1990).

Sakata extended the BM algorithm to 2 dimensions in Sakata (1988) and then to n dimensions in Sakata (1990, 2009). The so-called Berlekamp – Massey – Sakata algorithm (BMS) computes a Gröbner basis of the ideal of relations satisfied by the first terms of the input sequence, (Sakata, 1990, Lemma 5).

Direct and important applications of this generalization can be found in Coding Theory: n-dimensional cyclic codes, a generalization of Reed Solomon codes, can be decoded using the BMS algorithm, Sakata (1991). As the ouput of the BMS algorithm is a Gröbner basis, a natural application is the computation of a Gröbner basis of an ideal for another order, in fact the latest versions of the Sparse-FGLM algorithm rely heavily on the BM and BMS algorithms.

Related work

In the 18th century, Gauß was interested in predicting the next term of a sequence. Given a discrete set $(u_i)_{i\in\mathbb{N}}$, find the best coefficients, in the least-squares sense, $(\alpha_i)_{i\in\mathbb{N}}$ that will approximate u_i by $-\sum_{k=1}^{d}\alpha_{n-k}\,u_k$.

This yields a linear system whose matrix is Toeplitz and symmetric. This problem has also been extensively used in Digital Signal Processing theory and applications. Numerically, Levinson – Durbin recursion method can be used to solve this problem. Hence, to some extent, the original Levinson – Durbin problem in Norbert Wiener's Ph.D. thesis, Levinson (1947); Wiener (1964), predates the Hankel interpretation of the Berlekamp – Massey algorithm, see for instance Jonckheere and Ma (1989).

We refer to Kaltofen and Pan (1991); Kaltofen and Yuhasz (2013a,b). A very nice classification of the BM algorithms for solving this problem, and generalization to matrix sequences, can be found in the latter two. Of particular importance for us is the solution of the underlying linear system in Toeplitz/Hankel form. Let us also recall that the BM algorithm can be considered as a simplified version of the extended Euclidean algorithm. Indeed, the BM algorithm called on table $(u_0, u_1, \ldots, u_{2d-1})$ will do the same computations as the extended Euclidean algorithm with input polynomials x^{2d} and $u_0\, x^{2d-1} + u_1\, x^{2d-2} + \cdots + u_{2d-1}$.

The BM algorithm has two classical forms: an algebraic form, handling polynomials, and a matrix form, yielding a Hankel linear system. The BMS algorithm, Sakata (1988, 1990), extends the algebraic form of the BM algorithm to n dimensions. For a 0-dimensional ideal, according to Sakata (1990), the BMS algorithm can be used for computing a Gröbner basis.

The BM and BMS algorithms only *guess* and check that the computed relations are valid for a finite number of terms of the input sequence. Then, usually, one needs to prove that they are satisfied for all the terms of the sequence. In this paper, a *table* shall denote a finite subset of terms of a sequence: it is one of the input parameter of the algorithms since one cannot handle an infinite sequence.

Contributions

First of all, we recall what are linear recursive sequences in n dimensions, following Fitzpatrick and Norton (1990); Sakata (1988) and give some characterizations thereof. 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 Section 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).

As a first step, we try to rely on the BM algorithm to solve the n dimensional case: applying a random change of variables yields a new table whose relations are simpler, see Section 3. Exploiting this property yields Theorem 1.

Theorem 1. *Let $\mathbf{u} = (u_{i_1,\ldots,i_n})_{i_1,\ldots,i_n \in \mathbb{N}}$ 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 $O(n^{2d} + n\,\mathsf{M}(d) \log d)$ operations in $\mathbb{K}$, where $\mathsf{M}(d)$ is the complexity of multiplying two polynomials of degree at most $d - 1$.*

Under genericity assumptions, this probabilistic technique essentially reduces the problem to using the efficient 1-dimensional BM algorithm.

We extend the Hankel interpretation of the BM algorithm into a *multi-Hankel* one (a multivariate generalization thereof) for this problem. This allows us to characterize the output and to give properties of our algorithms. It also helps us to simplify the proofs of our results. The objective of these algorithms is to extract a maximum full rank submatrix of such a multi-Hankel matrix. This is exactly what does our first FGLM-like algorithm, Scalar-FGLM, in Sections 4. It can be observed that if one were to build a linear recursive sequence from an ideal J, the Scalar-FGLM algorithm would not necessarily return J as the ideal of relations of the table independently from the initial conditions. The first result is that the output ideal is necessarily *Gorenstein*, see Brachat et al. (2010); Gorenstein (1952); Macaulay (1934). In particular, if the solution points of J have no multiplicities, then the output is the expected one, that is J. This linear algebra interpretation allows us to also notice that the BMS algorithm can only return Gorenstein ideals.

Sections 4 and 5 are devoted to FGLM-like algorithms for computing Gröbner bases of the ideal of relations, namely Scalar-FGLM and Adaptive Scalar-FGLM. As for the BM and BMS algorithms, the Scalar-FGLM algorithm needs an upper bound on the order of the table to compute the Gröbner basis of the ideal of relations. On the one hand, whenever the order of the table is relatively big, this algorithm is efficient. On the other hand, when the order of the table is abnormally small, we propose an output sensitive probabilistic algorithm, called Adaptive Scalar-FGLM: this time an estimate of the order of the table 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 $u_{i_1,i_2,\ldots}$ of the table; thus the number of table queries has to be minimized. In fact, the main drawback of the change of variables is the large number of queries to the original table. The FGLM application is a kind of application wherein the knowledge of a table element is expensive as each query requires a matrix-vector product to be computed. Though the number of table queries can be sharply estimated by counting the number of distinct elements in a multi-Hankel matrix. This quantity can also be linked to the geometry of the staircase of the computed Gröbner basis.

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

We show that in favorable cases such as convex ones, the complexity is essentially linear in the size of the output. However, we also exhibit pathological cases where the complexity grows quadratically.

In Sections 5.2 and 5.3, to illustrate the efficiency of the proposed algorithms, we report on experiments for two applications: the Sparse-FGLM algorithm and the decoding of n-dimensional cyclic codes. The results of the experiments are fully in line with the theory: for instance, in coding theory, when t errors are generated randomly, they can be recovered in $O(t)$ evaluation of the syndromes.

For the LEX ordering, multi-Hankel matrices are heavily structured. If d_i is the maximal degree of the polynomials in x_i, then the matrix is quasi-Hankel with displacement rank $d_2 \cdots d_n$. This allows us to use fast quasi-Hankel arithmetic, see Bostan et al. (2007), in Section 6, to solve the underlying linear system in $O((d_2 \cdots d_n)^{\omega-1}\,\mathsf{M}(d_1 \cdots d_n)\,\log(d_1 \cdots d_n))$ operations in the base field. Whenever $d_2, \ldots, d_n$ are bounded, denoting $\delta = d_2 \cdots d_n$, this complexity is quasi-linear in the size of the staircase of the Gröbner basis: $O(\delta^{\omega-1}\,\mathsf{M}(d_1\,\delta)\,\log(d_1\,\delta)) = O(\mathsf{M}(d)\,\log d)$.

In Section 7, we recall that the generating series of the recursive linear sequences is of special interest. It is a rational fraction whose denominator factors into univariate polynomials. We propose several algorithms for computing this rational fraction, a deterministic one based on the Scalar-FGLM algorithm and a fast probabilistic one using the BM algorithm. On the one hand, thanks to its factorization into univariate polynomials, the denominator can be computed in $O(n\,\mathsf{M}(d)\,\log d)$ operations in the base field through our fast probabilistic algorithm. On the other hand, though not mandatory, expanding the numerator can be done in at most $O(n\,d^{n-1}\,\mathsf{M}(d))$ operations in the base field. This is coherent with the fact that the numerator is a dense multivariate polynomial of degree $d_i \leq d$ in each x_i.

As stated above, the BMS algorithm is an extension to the algebraic form of the BM algorithm. We left as an open question whether our algorithms could be seen as a matrix version of the BMS algorithm.

Amongst the changes from the ISSAC version of the paper (Berthomieu et al. (2015)), we now mention that only Gorenstein ideals can be recovered as ideal of relations. This gives another probabilistic test for the Gorenstein property, see also Daleo and Hauenstein (2015). The Adaptive Scalar-FGLM algorithm, presented in the ISSAC paper and in Section 5, could fail on tables satisfying a relation for a while and then switching not to satisfy it anymore. In particular, it fails if the first element is 0. The Extended Adaptive Scalar-FGLM algorithm, or Algorithm 5, extends the Adaptive Scalar-FGLM algorithm through an input parameter allowing to check multiple table elements at a time. This parameter represents the trade-off between an exact computation and the output-sensitivity of the Adaptive Scalar-FGLM algorithm. Finally, Section 7 with the generating series characterization of recursive sequences and the algorithms to compute the generating series was not present in the ISSAC version of the paper.

2. Definition and Characterization of Linear Recursive Sequences

This section is devoted to characterizing linear recursive (with constant coefficients) sequences. This is done in Definition 1 and in Proposition 4. In Section 2.3, we adopt an FGLM viewpoint to describe a multidimensional linear recursive sequence.

We shall use standard notation, with bolding letters corresponding to vectors or sequences. In particular, we let $\mathbf{i} = (i_1, \ldots, i_n) \in \mathbb{N}^n$ and $\mathbf{x} = (x_1, \ldots, x_n)$. As usual, we write $\mathbf{x}^{\mathbf{i}}$ for $x_1^{i_1}\cdots x_n^{i_n}$ and $|\mathbf{i}| = i_1 + \cdots + i_n$. When needed e_i shall be the ith element of the canonical basis of $\mathbb{Z}^n$. In the remaining part of the paper, $\mathbf{u} = (u_{\mathbf{i}})_{\mathbf{i}\in\mathbb{N}^n}$ shall be a n-dimensional sequence over a field $\mathbb{K}$.

Let us recall that a one-dimensional sequence $\mathbf{u} = (u_i)_{i\in\mathbb{N}}$ over $\mathbb{K}$ is linear recursive if there exist $\alpha_0,\ldots,\alpha_{d-1} \in \mathbb{K}$ such that for all $i \in \mathbb{N}$

$$u_{i+d} = -\sum_{k=0}^{d-1} \alpha_k\, u_{i+k}.$$

Such a relation shall be called *linear recurrence relation (with constant coefficients)*.

Furthermore, if d is minimal, then $\mathbf{u}$ is said to be *linear recursive of order d*. It is quite easy to see that the knowledge of $u_0,\ldots,u_{d-1},\alpha_0,\ldots,\alpha_{d-1}$ allows one to compute any term u_i of $\mathbf{u}$. In Section 7, we also remind the reader that the generating series is a rational fraction,

$$\sum_{i\in\mathbb{N}} u_i\, x^i \in \mathbb{K}(x).$$

Extending the notion of a linear recurrence relation satisfied by a multi-dimensional sequence $\mathbf{u} = (u_{\mathbf{i}})_{\mathbf{i}\in\mathbb{N}^n}$ is straightforward, see (Saints and Heegard, 1995, Definition 21): let $\mathcal{K}$ be a finite subset of $\mathbb{N}^n$, let $\boldsymbol{\alpha} = (\alpha_{\mathbf{k}})_{\mathbf{k}\in\mathcal{K}}$ be nonzero, then $\mathbf{u}$ satisifes the linear relation definied by $\boldsymbol{\alpha}$ if for all $\mathbf{i} \in \mathbb{N}^n$,

$$\sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}}\, u_{\mathbf{i}+\mathbf{k}} = 0.$$

In (Chabanne and Norton, 1992, Definition 2), the authors propose that a sequence shall be called linear recursive if it satisfies one linear recurrence relation. On the one hand, such sequences may not allow one to compute any term from a finite number of initial terms nor may they have a generating series that is a rational fraction. For instance $\mathbf{u} = (1/i!)_{(i,j)\in\mathbb{N}^2}$ satisfies $u_{i,j+1} - u_{i,j} = 0$ for all $(i,j) \in \mathbb{N}^2$ but one needs to know infinitely many initial conditions, say $u_{i,0} = 1/i!$, to compute any term. In this instance, the generating series has the closed form $\frac{\exp x}{1-y} \notin \mathbb{K}(x,y)$.

On the other hand, there exist sequences satisfying a linear recurrence relation with a generating series that is a rational fraction, yet they do not allow one to compute any term of the sequences from a finite number of initial terms. An example of this kind is $\mathbf{b}_{i,j} = \binom{i}{j}$ which satisfies Pascal's rule $b_{i+1,j+1} - b_{i,j+1} - b_{i,j} = 0$, its generating series is

$$\sum_{(i,j)\in\mathbb{N}^2} \binom{i}{j} x^i\, y^j = \sum_{i\in\mathbb{N}} x^i\,(1+y)^i = \frac{1}{1-x-x\,y} \in \mathbb{K}(x,y),$$

yet one cannot compute all the terms of sequence without the infinitely many initial conditions $u_{i,0} = 1$ for all $i \geq 0$ and $u_{0,j} = 0$ for all $j \geq 1$.

This motivates us to follow Sakata (1988) and Fitzpatrick and Norton (1990) to define linear recursive sequences. They were called "rectilinear recurrent sequences" in the latter.

Definition 1. *Let* $\mathbf{u} = (u_{\mathbf{i}})_{\mathbf{i}\in\mathbb{N}^n}$ *be a n-dimensional sequence with coefficients in* $\mathbb{K}$*. The sequence* $\mathbf{u}$ *is* linear recursive *if from a nonzero finite number of initial terms* $u_{\mathbf{i}}, \mathbf{i} \in S$*, and a finite number of linear recurrence relations, without any contradiction, one can compute any term of the sequence.*

In (Sakata, 2009, p. 147), the following sequence $\mathbf{u} = (u_{i,j})_{(i,j)\in\mathbb{N}^2}$ is exhibited with relations contradicting themselves. The set of initial terms is $\{u_{0,0}, u_{1,0}, u_{0,1}\}$ and the set of relations is $\{u_{i+2,j} - u_{i,j} = 0, u_{i+1,j+1} - u_{i,j} = 0, u_{i,j+2} - u_{i,j} = 0\}$. It is then clear that $u_{i+1,j} - u_{i,j+1} = 0$ implying, in particular, that the set of initial conditions is only $\{u_{0,0}, u_{1,0}\}$. Therefore, if $u_{0,1}$ were initialised with a different value from $u_{1,0}$, then the sequence would be inconsistant and one could not compute any term.

In Section 7, Proposition 18, we recall that the generating series of a *n-dimensional linear recursive sequence* is a rational fraction whose denominator can be factored into univariate polynomials. This is coherent with the binomial counter-example where the denominator of the generating series, $1 - x - x\,y$, cannot be factored into univariate polynomials.

Remark 3. *A linear recursive sequence is a special case of a* P-recursive *sequence whose recurrence relations only have constant coefficients, Koutschan (2013).*

Binomial sequence $\mathbf{b} = \left(\binom{i}{j}\right)_{(i,j)\in\mathbb{N}^2}$ *is P-recursive satisfying for all* $(i, j) \in \mathbb{N}^2$ *both relations*

$$(i - j + 1)\, b_{i+1,j} - (i + 1)\, b_{i,j} = 0, \quad (j + 1)\, b_{i,j+1} - (i - j)\, b_{i,j} = 0.$$

2.1. Ideal of relations

For a one-dimensional linear recursive sequence $\mathbf{u}$ satisfying the smallest relation $R(i) = u_{i+d} + \sum_{k=0}^{d-1} \alpha_k\, u_{i+k} = 0$ for all $i \in \mathbb{N}$ and with $\alpha_0, \ldots, \alpha_k$, polynomial $\mathrm{Pol}_{\mathbf{u}}(R)(x) = x^d + \sum_{k=0}^{d-1} \alpha_k\, x^k$ is called the *characteristic polynomial* of the sequence. One can prove that the vector space of sequences satisfying R has dimension d, corresponding to the d initial conditions $u_0, \ldots, u_{d-1}$ one can choose. Another way of seeing this vector space of dimension d is recalling that if $\zeta \neq 0$ is a root of $\mathrm{Pol}_{\mathbf{u}}(R)$ with multiplicity μ, then sequences $(i^m \zeta^i)_{i\in\mathbb{N}}$, with $0 \leq m < \mu$ satisfy R, hence d independent sequences.

Definition 2. *Let $\mathcal{K}$ be a finite subset of $\mathbb{N}^n$ and let $(\alpha_{\mathbf{k}})_{\mathbf{k}\in\mathcal{K}} \in \mathbb{N}^{\#\mathcal{K}}$ be a nonzero vector. We define the function $R : \mathbb{N}^n \to \mathbb{K}$ that maps $\mathbf{i}$ to $R(\mathbf{i}) = \sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}}\, u_{\mathbf{i}+\mathbf{k}}$. If $R(\mathbf{i}) = 0$ for all $\mathbf{i} \in \mathbb{N}^n$, then we say that the sequence $\mathbf{u}$ satisfies the linear recurrence relation $R(\mathbf{i}) = 0$.*

Finally, the associated polynomial *of $(\alpha_{\mathbf{k}})_{\mathbf{k}\in\mathcal{K}}$ is*

$$\operatorname{Pol}_{\mathbf{u}}(R)(\mathbf{x}) = \sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}}\, \mathbf{x}^{\mathbf{k}}.$$

Conversely, from any polynomial $P \in \mathbb{K}[\mathbf{x}]$, $P = \sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}}\, \mathbf{x}^{\mathbf{k}}$, we define the non-instantiated associated relation $\operatorname{Rel}_{\mathbf{u}}(P)(\mathbf{i}) = \sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}}\, u_{\mathbf{i}+\mathbf{k}}$.

In the end, we can always shift any polynomial so that it is enough to evaluate such a relation in $\mathbf{i} = (0, \dots, 0)$, consequently we will use the following convention:

$$\left[x_1^{i_1} \cdots x_n^{i_n}\right]_{\mathbf{u}} = [x_1^{i_1} \cdots x_n^{i_n}] = u_{i_1,\dots,i_n},$$
$$[P]_{\mathbf{u}} = [P] = \sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}}\, u_{\mathbf{k}}.$$

Example 1. *If $P(x, y) = x\,y - x - 1 \in \mathbb{K}[x, y]$ then $[P] = u_{1,1} - u_{1,0} - u_{0,0}$ and $[x^2\, y\, P] = u_{3,2} - u_{3,1} - u_{2,1}$.*

Let us now recall classical definitions and properties of Gröbner bases and admissible monomial orders. These will be used in Proposition 4, which can be seen as another equivalent definition of linear recursive sequences.

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

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

A Gröbner basis of an ideal I for $\prec$ is a finite subset $\mathcal{G}$ of I such that for all $f \in I$, there exists a $g \in \mathcal{G}$ s.t. $\operatorname{LT}_{\prec}(g) | \operatorname{LT}_{\prec}(f)$. The set of monomials in $\mathbf{x}$ that are not divisible by any $\operatorname{LT}_{\prec}(g), g \in \mathcal{G}$ forms a canonical basis of the algebra $\mathbb{K}[\mathbf{x}]/I$. It is called the *staircase* of $\mathcal{G}$ since these are exactly the monomials below $\{\operatorname{LT}_{\prec}(g), g \in \mathcal{G}\}$. If the algebra $\mathbb{K}[\mathbf{x}]/I$ has finite dimension over $\mathbb{K}$, then equivalently all the polynomials in I have finitely many common solutions over the algebraic closure of $\mathbb{K}$ and I is said 0-*dimensional*.

For a *homogeneous ideal I*, i.e. spanned by homogeneous polynomials, a *truncated Gröbner basis of I up to degree d* for $\prec$, or *d-truncated Gröbner basis*, is a finite subset $\mathcal{G}$ of I s.t. for all $f \in I$, if $\deg f \leq d$ then there exists a $g \in \mathcal{G}$ s.t.

$\mathrm{LT}(g) | \mathrm{LT}(f)$. This can be computed using any Gröbner basis algorithm by discarding critical pairs of degree greater than d.

For an affine ideal I, an analogous definition of d-truncated Gröbner basis exists. It is the output of a Gröbner basis algorithm discarding all critical pairs (f, φ) with $\deg \mathrm{LT}(f) + \deg \mathrm{LT}(\varphi) - \deg \mathrm{lcm}(\mathrm{LT}(f), \mathrm{LT}(\varphi)) > d$, i.e. with degree higher than d. In this situation, a d-truncated Gröbner basis $\mathcal{G}$ will span the subspace of polynomials $\sum_{g\in\mathcal{G}} h_g\, g$ with $\deg h_g \leq d - \deg g$.

Proposition 4. *Let* $\mathbf{u}$ *be a n-dimensional sequence defined over a field* $\mathbb{K}$*. The set of all the polynomials* $\mathrm{Pol}_{\mathbf{u}}(R)$*, with R a linear recurrence relation sastified by* $\mathbf{u}$*, is an ideal of* $\mathbb{K}[\mathbf{x}]$ *called the* ideal of relations*.*

Furthermore, $\mathbf{u}$ *is* linear recursive *if and only if its ideal of relations has dimension* 0*.*

Proof. We shall take the convention that the void relation, associated to the zero polynomial, is always true.

Let $(\alpha_{\mathbf{k}})_{\mathbf{k}\in\mathbb{N}^n}, (\beta_{\mathbf{k}})_{\mathbf{k}\in\mathbb{N}^n}$ be two vectors with only a finite number of nonzero coefficients such that for all $\mathbf{i} \in \mathbb{N}^n$, $R_1(\mathbf{i}) = \sum_{\mathbf{k}\in\mathbb{N}^n} \alpha_{\mathbf{k}}\, u_{\mathbf{i}+\mathbf{k}} = 0$ and $R_2(\mathbf{i}) = \sum_{\mathbf{k}\in\mathbb{N}^n} \beta_{\mathbf{k}}\, u_{\mathbf{i}+\mathbf{k}} = 0$, i.e. $\mathrm{Pol}_{\mathbf{u}}(R_1)$ and $\mathrm{Pol}_{\mathbf{u}}(R_2)$ are in this set. For any $\lambda \in \mathbb{K}$, $R_1(\mathbf{i}) + \lambda R_2(\mathbf{i}) = 0 = \sum_{\mathbf{k}\in\mathbb{N}^n} (\alpha_{\mathbf{k}} + \lambda \beta_{\mathbf{k}})\, u_{\mathbf{i}+\mathbf{k}}$ for all $\mathbf{i} \in \mathbb{N}^n$, thus $\mathrm{Pol}_{\mathbf{u}}(R_1 + \lambda R_2)$ is also in this set. Finally, for any $\mathbf{j} \in \mathbb{N}^n$, let us define the shifted vector $(\alpha'_{\mathbf{k}})_{\mathbf{k}\in\mathbb{N}^n}$ by for all $\mathbf{k} \in \mathbb{N}^n$, $\alpha'_{\mathbf{k}+\mathbf{j}} = \alpha_{\mathbf{k}}$. We denote by R'_1 the maps from $\mathbb{N}^n$ to $\mathbb{K}$, it induces. Since $R_1(\mathbf{i}) = 0$ for all $\mathbf{i} \in \mathbb{N}^n$, then $R_1(\mathbf{i} + \mathbf{j}) = 0 = \sum_{\mathbf{k}\in\mathbb{N}^n} \alpha_{\mathbf{k}}\, u_{\mathbf{k}+\mathbf{j}+\mathbf{i}} = \sum_{\mathbf{k}\in\mathbb{N}^n} \alpha'_{\mathbf{k}+\mathbf{j}}\, u_{\mathbf{k}+\mathbf{j}+\mathbf{i}} = R'_1(\mathbf{i})$ and $\mathrm{Pol}_{\mathbf{u}}(R'_1) = \mathbf{x}^{\mathbf{j}}\, \mathrm{Pol}_{\mathbf{u}}(R_1)$ is in this set.

All in all, this proves that this set of polynomials is an ideal of $\mathbb{K}[\mathbf{x}]$.

Let $\mathbf{u}$ be a n-dimensional linear recursive sequence, $\mathcal{K}$ be a finite subset of $\mathbb{N}^n$ of initial conditions of $\mathbf{u}$. For any $\mathbf{i} \in \mathbb{N}^n \setminus \mathcal{K}$, there exists a $\boldsymbol{\alpha}^{(\mathbf{i})} = \left(\alpha_{\mathbf{k}}^{(\mathbf{i})}\right)_{\mathbf{k}\in\mathcal{K}}$ s.t.

$$\forall \mathbf{j} \in \mathbb{N}^n, \quad u_{\mathbf{i}+\mathbf{j}} = \sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}}^{(\mathbf{i})}\, u_{\mathbf{k}+\mathbf{j}},$$

hence $\left(\mathbf{x}^{\mathbf{i}} - \sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}}^{(\mathbf{i})}\, \mathbf{x}^{\mathbf{k}}\right) \in I$. Therefore, in $\mathbb{K}[\mathbf{x}]/I$, all monomials $\mathbf{x}^{\mathbf{i}}$ with $\mathbf{i} \in \mathbb{N}^n \setminus \mathcal{K}$ reduce to a linear combination of the $\mathbf{x}^{\mathbf{k}}, \mathbf{k} \in \mathcal{K}$, hence $\mathbb{K}[\mathbf{x}]/I$ has finite dimension and I is 0-dimensional.

Conversely, let $\mathcal{G} = \{g_1, \ldots, g_m\}$ be a minimal reduced Gröbner basis of I for a monomial order $\prec$. There exists a finite subset S of $\mathbb{N}^n$ s.t. for all j, $1 \leq j \leq m$, $g_j = \mathbf{x}^{\mathbf{i}_j} - \sum_{\mathbf{k}\in S} \gamma_{\mathbf{i}_j,\mathbf{k}}\, \mathbf{x}^{\mathbf{k}}$ with $\gamma_{\mathbf{i}_j,\mathbf{k}} \in \mathbb{K}$. Let us prove we can set a finite number of terms of $\mathbf{u}$ and then compute any term. Let $u_{\mathbf{i}}$ be any term of the sequence. If $\mathbf{x}^{\mathbf{i}}$ is in the staircase of $\mathcal{G}$, then we set $u_{\mathbf{i}}$. Otherwise there exist j and $\mathbf{i}'$ s.t. $\mathbf{x}^{\mathbf{i}} = \mathbf{x}^{\mathbf{i}'}\mathbf{x}^{\mathbf{i}_j}$, hence $\mathbf{x}^{\mathbf{i}'} g_j = \mathbf{x}^{\mathbf{i}} - \sum_{\mathbf{k}\in S} \gamma_{\mathbf{i}_j,\mathbf{k}}\, \mathbf{x}^{\mathbf{i}'+\mathbf{k}} \in I$. By recurrence on the $\mathbf{x}^{\mathbf{i}'+\mathbf{k}} \prec \mathbf{x}^{\mathbf{i}}$, there exist $\alpha_{\mathbf{i},\mathbf{k}} \in \mathbb{K}$ s.t. $\mathbf{x}^{\mathbf{i}} - \sum_{\mathbf{k}\in S} \alpha_{\mathbf{i},\mathbf{k}} \mathbf{x}^{\mathbf{k}} \in I$. Therefore $u_{\mathbf{i}} - \sum_{\mathbf{k}\in S} \alpha_{\mathbf{i},\mathbf{k}}\, u_{\mathbf{k}} = 0$ and one can compute any $u_{\mathbf{i}}$ from a finite number of initial terms. □

In other words, $\mathbf{u}$ is linear recursive if and only if $\mathbb{K}[\mathbf{x}]/I$ is a finite dimensional $\mathbb{K}$-vector space. The dimension shall be called the *order* of $\mathbf{u}$.

This is related to the definition of a *holonomic* function in an Ore algebra $\mathcal{A} = \mathbb{K}(\mathbf{z})\langle\partial_{\mathbf{z}}\rangle$, see (Koutschan, 2013, Definition 1). If $\operatorname{Ann}(f)$ is the left ideal of polynomials vanishing on $f = \sum_{\mathbf{i}\in\mathbb{N}^n} u_{\mathbf{i}}\,\mathbf{z}^{\mathbf{i}}$, then $\mathcal{A}/\operatorname{Ann}(f)$ is a finite dimensional vector space over $\mathcal{A}$.

Example 2. *Consider the following sequences:*

1. $\mathbf{u} = \left((2^i + 3^i)\,7^j\right)_{(i,j)\in\mathbb{N}^n}$ *is linear recursive of order* 2 *satisfying the relations*

$$\forall\,(i,j)\in\mathbb{N}^2,\; u_{i+2,j} - 5\,u_{i+1,j} + 6\,u_{i,j} = 0,\quad u_{i,j+1} - 7\,u_{i,j} = 0.$$

Its ideal of relations is $\langle (x-2)\,(x-3), y-7\rangle$. *The polynomial system spanned by these equations has two solutions with multiplicity* 1.

2. $\mathbf{v} = \left((i+1)\,2^i\,7^j\right)_{(i,j)\in\mathbb{N}^2}$ *is linear recursive of order* 2 *satisfying the relations*

$$\forall\,(i,j)\in\mathbb{N}^2,\; v_{i+2,j} - 4\,v_{i+1,j} + 4\,v_{i,j} = 0,\quad v_{i,j+1} - 7\,v_{i,j} = 0.$$

Its ideal of relations is $\langle (x-2)^2, y-7\rangle$. *The polynomial system spanned by these equations has one solution with multiplicity* 2.

3. $\mathbf{b} = \left(\binom{i}{j}\right)_{(i,j)\in\mathbb{N}^2}$ *is* not *linear recursive, however it is* holonomic. *Calling the* BMS *algorithm or Algorithms 3 and 4 on this table for all* $b_{i,j}$, $i + j \le 2d$, *one obtains relations*

$$b_{i+1,j+1} - b_{i,j+1} - b_{i,j} = 0,$$

the famous Pascal's rule, together with

$$\sum_{k=0}^{d} \binom{d}{k} (-1)^{d-k}\, b_{i+k,j} = 0,\quad b_{i,j+d} = 0.$$

From the polynomial point of view, they form a d-truncated Gröbner basis of $I = \langle (x-1)^d, x\,y - x - y, y^d\rangle = \langle 1\rangle$. *Let us notice that* 1 *is reached by these polynomials only as a linear combinations of degree* $d+1$ *and the ideal* $\langle 1\rangle$ *has not dimension* 0 *but* -1. *From the sequence point of view, one needs to add infinitely many initial conditions to compute the whole sequence.*

2.2. Gorenstein ideals

In the proof of Proposition 4, we showed that a Gröbner basis $\mathcal{G}$ of an ideal J and the initial conditions $\{u_{\mathbf{i}}, \mathbf{i}\in\mathcal{K}\}$, with $\mathcal{K}$ the staircase of $\mathcal{G}$ define uniquely a sequence as they allow one to compute any term thereof. However, the ideal of relations of this sequence need not be J.

Let $\mathbb{K}$ be an effective field and let $a, b \in \mathbb{K}$ with $a \neq 0$. Let us consider the sequence $\mathbf{u}$ built from $J = \langle x^2 \rangle$ and initial conditions $\{u_0 = a, u_1 = b\}$. It is quite clear that the ideal of relations of $\mathbf{u}$ is $I = \langle x \rangle \supset J$ if and only if $b = 0$ and that it is J otherwise.

More generally, let $J = \langle Q \rangle$ be the ideal used to build the sequence $\mathbf{u} = (u_i)_{i \in \mathbb{N}}$ and let $I = \langle P \rangle$ be the ideal of relations of $\mathbf{u}$. Since $[Q]_{\mathbf{u}} = 0$, then $Q \in I$ and $J \subseteq I$. Therefore, if the ideal of relations is bigger then there are relations induced by the recurrence system between the initial conditions.

For multidimensional sequences, the situation is more complex. The ideal of relations can intrinsically be bigger than the ideal used to build the sequence whatever the initial conditions are. Let $a, b, c \in \mathbb{K}$ with $a \neq 0$. Let $J = \langle x^2, x\,y, y^2 \rangle$ with initial conditions $\{u_{0,0} = a, u_{1,0} = b, u_{0,1} = c\}$. It is easy to check that for all $i, j \in \mathbb{N}$, $c\,u_{i+1,j} - b\,u_{i,j+1} = 0$ meaning that $c\,x - b\,y$ must be in I. Hence, whenever b and c are not 0, the ideal is in fact $I = \langle x - b\,y/c, y^2 \rangle \supset J$. If $b = 0, c \neq 0$ then $I = \langle x, y^2 \rangle$, else if $b \neq 0, c = 0$ then $I = \langle x^2, y \rangle$ else $b = c = 0$ and $I = \langle x, y \rangle$. All these cases are generalizations of the dimension 1 situation.

Proposition 3.3 in Brachat et al. (2010) proves that the ideal of relations of a linear recursive sequence is necessarily *Gorenstein*, Gorenstein (1952); Macaulay (1934), and problems occur only if J has a zero of multiplicity at least 2. The following theorem can also be found in (Elkadi and Mourrain, 2007, Theorem 8.3).

Theorem 5. *Let $I \subseteq \mathbb{K}[\mathbf{x}]$ be a 0-dimensional ideal and let $R = \mathbb{K}[\mathbf{x}]/I$. The ideal I (resp. ring R) is* Gorenstein *if equivalently*

1. *R and its dual are isomorphic as R-modules;*
2. *there exists a $\mathbb{K}$-linear form τ on R such that the following bilinear form is non degenerate*

$$\begin{aligned} R \times R &\to \mathbb{K} \\ (a, b) &\mapsto \tau(a\,b). \end{aligned}$$

On the one hand, this result is important for the Sparse-FGLM application. If the input ideal is not Gorenstein, the output ideal will be bigger. However, this can be easily tested by comparing the degrees of the input and output ideals. On the other hand, this yields a probabilistic test for the Gorenstein property of an ideal J. Pick at random initial conditions, construct a sequence thanks to these initial conditions and J and then compute the ideal I of relations of the sequence. If $I = J$, then J is Gorenstein. We refer to Daleo and Hauenstein (2015) for another test on the Gorenstein property of an ideal.

As stated above, a non Gorenstein zero-dimensional ideal induces a polynomial system with a zero of multiplicity at least 2. However, this does not mean that ideals with multiplicities are out of reach. The ideal of relations of the sequence $\mathbf{u} = (u_{i,j})_{(i,j)\in\mathbb{N}^2}$ defined by $u_{i,j} = 1$ if $i, j \leq 1$ and $u_{i,j} = 0$ otherwise is exactly $J = \langle x^2, y^2\rangle$. Yet, the vanishing variety of J is the origin with multiplicity 4.

2.3. Matrix multiplications in the quotient ring point of view

This section introduces a key point for the adaptive Algorithm 4, designed in Section 5.

In Proposition 4, we showed that the initial terms of a sequence and the ideal of relations are enough to allow one to compute any term of the sequence. In fact, adopting a FGLM viewpoint, we will show that any term is a scalar product between two vectors, one of them being the vector of the initial terms.

Let $\mathbf{u} = (u_{\mathbf{i}})_{\mathbf{i}\in\mathbb{N}^n}$ be a linear recursive sequence over $\mathbb{K}$. Let $\mathcal{K}$ be the staircase of a Gröbner basis of its ideal of relations I, then $\mathbb{K}[\mathbf{x}]/I$ is a $\mathbb{K}$-algebra whose canonical basis as a vector space is $(\mathbf{x}^{\mathbf{k}})_{\mathbf{k}\in\mathcal{K}}$. Let us define T_j to be the multiplication matrices by x_j in $\mathbb{K}[\mathbf{x}]/I$ for all j, $1 \leq j \leq n$. The identity element $1 \in \mathbb{K}[x]/I$ is represented by the first vector of the canonical basis $\mathbf{1} = (1, 0, \dots, 0)^{\mathrm{T}}$. For any $\mathbf{i} = (i_1, \dots, i_n) \in \mathbb{N}^n$, the vector representing $\mathbf{x}^{\mathbf{i}}$ in $\mathbb{K}[x]/I$ is $T_1^{i_1}\cdots T_n^{i_n}\cdot\mathbf{1}$. Now, to express $u_{\mathbf{i}}$ in terms of the $u_{\mathbf{k}}, \mathbf{k} \in \mathcal{K}$, it suffices to perform the scalar product $u_{\mathbf{i}} = \langle \mathbf{r}, T_1^{i_1}\cdots T_n^{i_n}\cdot\mathbf{1}\rangle$ where $\mathbf{r} = (u_{\mathbf{0}}, \dots) = (u_{\mathbf{k}})_{\mathbf{k}\in\mathcal{K}}$.

Example 3. *The set $G = \{x - y, y^2 - y - 1\}$ is a Gröbner basis of the ideal $I = \langle G\rangle$ for the* DRL *ordering with $y \prec x$ whose staircase $\mathcal{K}$ is $\{1, y\}$. Let $\mathbf{u} = (u_{i,j})_{(i,j)\in\mathbb{N}^2}$ be a two-dimensional sequence such that for all $(i, j) \in \mathbb{N}^2$, $u_{i,j} = F_{i+j}$ where $(F_i)_{i\in\mathbb{N}}$ is the Fibonacci sequence. Clearly for all $(i, j) \in \mathbb{N}^2$, $[(x - y)\, x^i\, y^j] = u_{i+1,j} - u_{i,j+1} = F_{i+j+1} - F_{i+j+1} = 0$ and $[(y^2 - y - 1)\, x^i\, y^j] = u_{i,j+2} - u_{i,j+1} - u_{i,j} = F_{i+j+2} - F_{i+j+1} - F_{i+j} = 0$. There are no smaller relations so that I is the ideal of relations of* $\mathbf{u}$.

We let $\mathbf{r}$ be the vector of initial conditions $\mathbf{r} = (u_{0,0}, u_{0,1}) = (0, 1)$ and T_1, T_2 be the multiplication matrices by x and y in $\mathbb{K}[x, y]/I$ for the canonical basis $(1, y)$. Thus, $T_1 = T_2 = \left(\begin{smallmatrix}0 & 1\\ 1 & 1\end{smallmatrix}\right)$ and for all $(i, j) \in \mathbb{N}^2$, $(i, j) \neq (0, 0)$,

$$\langle \mathbf{r}, T_1^i \cdot T_2^j \cdot \mathbf{1}\rangle = \begin{pmatrix}0 & 1\end{pmatrix}\begin{pmatrix}0 & 1\\ 1 & 1\end{pmatrix}^{i+j}\begin{pmatrix}1\\ 0\end{pmatrix} = \begin{pmatrix}0 & 1\end{pmatrix}\begin{pmatrix}F_{i+j-1} & F_{i+j}\\ F_{i+j} & F_{i+j+1}\end{pmatrix}\begin{pmatrix}1\\ 0\end{pmatrix} = F_{i+j} = u_{i,j}.$$

3. Randomized Reduction: from the Berlekamp – Massey – Sakata Algorithm to the Berlekamp – Massey Algorithm

When solving algebraic systems coming from applications, computational difficulties might appear because of the choices of the variables. In general, a random linear change of variables is first applied on the system so that it has better chances to behave as a generic system. After the computations, the inverse map is applied to the results.

In this section, we introduce an action of $\mathrm{GL}_n(\mathbb{K})$, the group of invertible matrices of size n, on a n-dimensional sequence $\mathbf{u} = (u_{\mathbf{i}})_{\mathbf{i}\in\mathbb{N}^n}$. With good probability, this randomized preprocessing will yield a new sequence that will have one linear recurrence relation of the form $u_{\mathbf{i}+d\,e_1} - \sum_{k=0}^{d-1} \alpha_k\, u_{\mathbf{i}+k\,e_1} = 0$, for all $\mathbf{i}$ and other relations of the type $u_{\mathbf{i}+e_j} - \sum_{k=0}^{d-1} \beta_{j,k}\, u_{\mathbf{i}+k\,e_1} = 0$. These recurrence relations means that the first one should be computed by calling the BM algorithm on the subsequence $(u_{i\,e_1})_{i\in\mathbb{N}}$ while each other one is found by solving a special linear system.

Therefore, the bottleneck of the execution of the BMS algorithm on this table would be the computation of the first relation.

3.1. Linear Transformation of the Table

In this section, we describe the action of $\mathrm{GL}_n(\mathbb{K})$ over the set $\mathbb{K}^{\mathbb{N}^n}$ of n-dimensional sequences over $\mathbb{K}$. For a matrix $A = (a_{i,j})_{1\le i,j\le n} \in \mathrm{GL}_n(\mathbb{K})$, we denote

$$A\,\mathbf{x} = \boldsymbol{\xi} = (\xi_1,\dots,\xi_n) = \left(\sum_{k=1}^{n} a_{1,k}\,x_k,\dots,\sum_{k=1}^{n} a_{n,k}\,x_k\right).$$

Then, by extension, for all $\mathbf{i} = (i_1,\dots,i_n) \in \mathbb{N}^n$, we let

$$(A\,\mathbf{x})^{\mathbf{i}} = \boldsymbol{\xi}^{\mathbf{i}} = \prod_{k=1}^{n} \xi_k^{i_k}.$$

We define the action of A on an n-dimensional sequence as follows.

Definition 3. *Let* $\mathbf{u} = (u_{\mathbf{i}})_{\mathbf{i}\in\mathbb{N}^n}$ *be a n-dimensional sequence. We define the* action *of an invertible matrix* $A \in \mathrm{GL}_n(\mathbb{K})$ *on* $\mathbf{u}$ *as* $A\cdot\mathbf{u} = \left(\left(\left[(A\,\mathbf{x})^{\mathbf{i}}\right]_{\mathbf{u}}\right)_{\mathbf{i}}\right)_{\mathbf{i}\in\mathbb{N}^n}$.

For $\mathbf{u} = (u_{i,j})_{(i,j)\in\mathbb{N}^2}$, $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{GL}_2(\mathbb{K})$ and $\mathbf{v} = A\cdot\mathbf{u}$, we have

$$\begin{aligned}
v_{0,0} &= u_{0,0}, & & \\
v_{1,0} &= a\,u_{1,0} + b\,u_{0,1}, & v_{0,1} &= c\,u_{1,0} + d\,u_{0,1},\\
v_{2,0} &= a^2\,u_{2,0} + 2\,a\,b\,u_{1,1} + b^2\,u_{0,2}, & v_{1,1} &= a\,c\,u_{2,0} + (a\,d + b\,c)\,u_{1,1} + b\,d\,u_{0,2},\dots
\end{aligned}$$

The following proposition shows that the action of A on a sequence $\mathbf{u}$ extends to polynomials naturally, i.e. as the classical action of $\mathrm{GL}_n(\mathbb{K})$ on polynomials in $\mathbb{K}[\mathbf{x}]$.

Proposition 6. *Let* $\mathbf{u}$ *be a n-dimensional sequence. Let P be a polynomial associated to a linear recurrence relation of* $\mathbf{u}$. *Let A be an invertible matrix of size n and let* $\mathbf{v} = A \cdot \mathbf{u}$. *Then the polynomial* $P(A^{-1}\,\mathbf{x})$ *is associated to a linear recurrence relation of* $\mathbf{v}$.

Proof. Let $\mathbf{i} \in \mathbb{N}^n$. Since $v_{\mathbf{i}}$ is merely the polynomial $(A\,\mathbf{x})^{\mathbf{i}}$ evaluated in $\mathbf{u}$, any polynomial $P(A^{-1}\,\mathbf{x})$ evaluated in $\mathbf{v}$ will yield $P(\mathbf{x})$ evaluated in $\mathbf{u}$. Therefore, $\left[P(A^{-1}\,\mathbf{x})\right]_{\mathbf{v}} = 0$ if and only if $[P(\mathbf{x})]_{\mathbf{u}} = 0$. □

Therefore, the preprocessing of applying a randomized invertible linear map on the table can be seen as applying – the same – randomized invertible linear transformation on the variables appearing in the ideal of relations of the table.

3.2. Essential reduction to the Berlekamp – Massey algorithm

We design an algorithm that computes the ideal of relations of a table $\mathbf{u}$ using a randomized linear transformation of the table and running the BM algorithm on the new table.

Let $\mathbf{u} = (u_{\mathbf{i}})_{\mathbf{i}\in\mathbb{N}^n}$ be a n-dimensional linear recursive sequence and let $I \in \mathbb{K}[\mathbf{x}]$ be its ideal of relation. We shall introduce a new variable t and a new ideal $J = I + (t-1)$ in $\mathbb{K}[\mathbf{x}, t]$. The ideal J is consequently the ideal of relation of the $(n+1)$-dimensional sequence $\mathbf{v} = (v_{\mathbf{i},j})_{(\mathbf{i},j)\in\mathbb{N}^n\times\mathbb{N}}$ defined by

$$\forall\, \mathbf{i} \in \mathbb{N}^n,\ \forall\, j \in \mathbb{N},\ v_{\mathbf{i},j} = u_{\mathbf{i}},$$

where t represents the last coordinate.

Generically, when applying a change of coordinates fixing each x_i and mapping t onto $t + \sum_{k=1}^{n} c_k\, x_k$ for some $c_1, \ldots, c_n \in \mathbb{K}$, the minimal reduced Gröbner basis of the new ideal J' obtained from J for the LEX order with $t < x_1 < \cdots < x_n$ is in *shape position*. This means the Gröbner basis is $\langle f(t), x_1 - f_1(t), \ldots, x_n - f_n(t)\rangle$, with $\forall\, k \in \{1, \ldots, n\}$, $\deg f_k < \deg f$, see Gianni and Mora (1989); Lakshman (1990).

As $f(t) = \sum_{k=0}^{d} \alpha_k\, t^k$ depends only on t, it is found by running the BM algorithm on the first terms of the subsequence $(\tilde{v}_{\mathbf{0},j})_{j\in\mathbb{N}}$, with $\mathbf{0} = (0, \ldots, 0) \in \mathbb{N}^n$, of $\tilde{\mathbf{v}}$ obtained from $\mathbf{v}$ and the aforementioned change of coordinates. Each polynomial $x_k - f_k(t) = x_k - \sum_{\ell=0}^{d-1} \beta_{k,\ell}\, t^\ell$, for $1 \le k \le n$, is then found by solving the linear system

$$\begin{cases} \tilde{v}_{e_k,0} &= \sum_{\ell=0}^{d-1} \beta_{k,\ell}\, \tilde{v}_{\mathbf{0},\ell} \\ \quad\vdots & \qquad\vdots \\ \tilde{v}_{e_k,d-1} &= \sum_{\ell=0}^{d-1} \beta_{k,\ell}\, \tilde{v}_{\mathbf{0},\ell+d-1} \end{cases}$$

whose matrix is Hankel.

Therefore, after applying a linear transformation on the table, finding the first relation only in t essentially comes down to running the BM algorithm on a 1-dimensional subtable. Then, the other relations are found by solving Hankel systems with the exact same matrix as in the call to the BM algorithm but with different right-hand side vectors. This is summed up in Algorithm 1.

Algorithm 1: The BM algorithm for n-dimensional tables.

Input: A n-dimensional table $\mathbf{u} = (u_{\mathbf{i}})_{\mathbf{i}\in\mathbb{N}^n}$.
Output: An equivalent basis of relations of $\mathbf{u}$.
Pick at random $c_1, \ldots, c_n \in \mathbb{K}$ and let $\xi = 1 + c_1\,x_1 + \ldots + c_n\,x_n$.
Compute $\tilde{\mathbf{v}} = (\tilde{v}_{\mathbf{i},j})_{(\mathbf{i},j)\in\mathbb{N}^n\times\mathbb{N}} = \left(\left([\mathbf{x}^{\mathbf{i}}\,\xi^j]_{\mathbf{u}}\right)_{\mathbf{i},j}\right)_{(\mathbf{i},j)\in\mathbb{N}^n\times\mathbb{N}}$.
Compute f of degree d by running the BM algorithm on $\left(\tilde{v}_{\mathbf{0},j}\right)_{j\in\mathbb{N}}$.
For k **from** 1 **to** n **do**
  Solve the Hankel linear system

$$\begin{pmatrix} \tilde{v}_{\mathbf{0},0} & \cdots & \tilde{v}_{\mathbf{0},d-1} \\ \vdots & \iddots & \vdots \\ \tilde{v}_{\mathbf{0},d-1} & \cdots & \tilde{v}_{\mathbf{0},2\,d-2} \end{pmatrix} \begin{pmatrix} \beta_{k,0} \\ \vdots \\ \beta_{k,d-1} \end{pmatrix} = \begin{pmatrix} \tilde{v}_{e_k,0} \\ \vdots \\ \tilde{v}_{e_k,d-1} \end{pmatrix}.$$

Return $f, x_1 - \sum_{\ell=0}^{d-1} \beta_{1,\ell}\,t^\ell, \ldots, x_n - \sum_{\ell=0}^{d-1} \beta_{n,\ell}\,t^\ell$.

Let us illustrate this algorithm with the following example.

Example 4. *We consider the sequence* $\mathbf{u} = ((-1)^{i_1\,i_2})_{(i_1,i_2)\in\mathbb{N}^2}$. *We set* $c_1 = -1$ *and* $c_2 = 2$ *and get the sequence* $\tilde{\mathbf{v}} = (\tilde{v}_{i_1,i_2,j})_{(i_1,i_2,j)\in\mathbb{N}^3}$. *The first terms of the subsequence* $\left(\tilde{v}_{0,0,j}\right)_{j\in\mathbb{N}}$ *are* $(1, 2, 12, 32, 144, 512, 2112, 8192, 33024, 131072)$.

Calling the BM *algorithm on this table yields* $f(t) = t^4 - 4\,t^3 - 4\,t^2 + 16\,t$. *Then solving*

$$\begin{pmatrix} \tilde{v}_{0,0,0} & \tilde{v}_{0,0,1} & \tilde{v}_{0,0,2} & \tilde{v}_{0,0,3} \\ \tilde{v}_{0,0,1} & \tilde{v}_{0,0,2} & \tilde{v}_{0,0,3} & \tilde{v}_{0,0,4} \\ \tilde{v}_{0,0,2} & \tilde{v}_{0,0,3} & \tilde{v}_{0,0,4} & \tilde{v}_{0,0,5} \\ \tilde{v}_{0,0,3} & \tilde{v}_{0,0,4} & \tilde{v}_{0,0,5} & \tilde{v}_{0,0,6} \end{pmatrix} \begin{pmatrix} \beta_{1,0} \\ \beta_{1,1} \\ \beta_{1,2} \\ \beta_{1,3} \end{pmatrix} = \begin{pmatrix} \tilde{v}_{1,0,0} \\ \tilde{v}_{1,0,1} \\ \tilde{v}_{1,0,2} \\ \tilde{v}_{1,0,3} \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 12 & 32 \\ 2 & 12 & 32 & 144 \\ 12 & 32 & 144 & 512 \\ 32 & 144 & 512 & 2112 \end{pmatrix} \begin{pmatrix} \beta_{1,0} \\ \beta_{1,1} \\ \beta_{1,2} \\ \beta_{1,3} \end{pmatrix} = \begin{pmatrix} 1 \\ -2 \\ -4 \\ -32 \end{pmatrix}$$

yields $\beta_{1,0} = -1, \beta_{1,1} = 2/3, \beta_{1,2} = 1/2, \beta_{1,3} = -1/6$ *so that we find the polynomial of relation* $x_1 + \frac{1}{6}\,t^3 - \frac{1}{2}\,t^2 - \frac{2}{3}\,t + 1$.

Likewise, we find $x_2 + \frac{1}{12}\,t^3 - \frac{1}{4}\,t^2 - \frac{5}{6}\,t + 1$ *when solving the same linear system but with the right-hand side vector* $\begin{pmatrix}\tilde{v}_{0,1,0}\\ \tilde{v}_{0,1,1}\\ \tilde{v}_{0,1,2}\\ \tilde{v}_{0,1,3}\end{pmatrix} = \begin{pmatrix}1\\ 4\\ 8\\ 40\end{pmatrix}$.

We mention that the degrees of I, J and J' coincide and are all d and that one can also obtain a set of generator of $I \subset \mathbb{K}[\mathbf{x}]$, the ideal of relations of the original sequence, i.e. the polynomials not in t, by computing a new Gröbner basis of the ideal for an order eliminating t, i.e. any monomial ordering wherein a monomial divisible by t is greater than any monomial only in $x_1, \dots, x_n$, for instance LEX with $x_1 \prec \cdots \prec x_n < t$. This can be done with the FGLM algorithm for instance and we refer the reader to Example 9 for a continuation of Example 4 with a computation of a Gröbner basis for an order eliminating t.

For instance, one can use Poteaux and Schost (2013)'s Las Vegas algorithm to change the order of a triangular set with complexity essentially that of modular composition $O(\mathsf{C}(d)) \subseteq O(d^{(\omega+1)/2})$ operations.

3.3. Complexity results

Proposition 7. *Let* $d \in \mathbb{N}$. *Computing terms* $v_{\mathbf{i}}$ *for all* $\mathbf{i} \in \mathbb{N}^n$ *such that* $|\mathbf{i}| \le d$ *can be done in* $O(n^{2\,d})$ *operations in* $\mathbb{K}$, $O(n^{2\,d})$ *memory space and* $O(n^d)$ *queries to the table elements.*

Proof. According to Section 3.1, for any $\mathbf{i} \in \mathbb{N}^n$ with $|\mathbf{i}| \le d$, one needs to compute $\boldsymbol{\xi}^{\mathbf{i}} = \xi^{i_1} \cdots \xi^{i_n}$, where $\boldsymbol{\xi} = A\,\mathbf{x}$.

Introducing a new set of variables $z_0, z_1, \dots, z_n$, we can get all the $\boldsymbol{\xi}^{\mathbf{i}}$ with $\mathbf{i} \le d$ by expanding

$$(z_0 + \xi_1\, z_1 + \ldots + \xi_n\, z_n)^d\,.$$

This allows us to directly determine all the polynomials we need to compute $v_{\mathbf{i}}$, $|\mathbf{i}| \le d$.

The linear form $z_0 + \xi_1\, z_1 + \ldots + \xi_n\, z_n$ has $n^2 + 1$ monomials, therefore its dth power has $\binom{n^2+d}{d} \in O(n^{2\,d})$ monomials, that must be all stored, and one needs to perform $O(n^{2\,d})$ operations in $\mathbb{K}$ to compute them.

To obtain each $v_{\mathbf{i}}$, we evaluate the polynomial by replacing $\mathbf{x}^{\mathbf{i}}$ by $u_{\mathbf{i}}$. For a given $\mathbf{i} \in \mathbb{N}^n$, $|\mathbf{i}| \le d$, all the replacements of an $\mathbf{x}^{\mathbf{i}}$ by $u_{\mathbf{i}}$ requires a lonely query to the table, hence a global total of $O(n^d)$ queries.

For $\delta \in \{0, .., d\}$, each of the $O(n^\delta)$ different polynomials of degree δ has $O(n^\delta)$ coefficients. Thus $O\left(\sum_{\delta=0}^{d} n^{2\,\delta}\right) = O(n^{2\,d})$ multiplications must be performed. □

Proof of Theorem 1. Besides the change of basis in $O(n^{2\,d})$, we need to run the BM algorithm once in $O(\mathsf{M}(d)\log d)$ operations in $\mathbb{K}$. Then, each Hankel system can be solved in $O(\mathsf{M}(d)\log d)$ operations, according to Bostan et al. (2007). □

4. Multi-Hankel Solver

This section is devoted to the design of an FGLM-like algorithm for computing the Gröbner basis of the ideal of relations of a table $\mathbf{u} = (u_{\mathbf{i}})_{\mathbf{i}\in\mathbb{N}^n}$, with coefficients in $\mathbb{K}$.

We fix $\prec$ to be an admissible monomial ordering on $x_1 \ldots, x_n$, then $\mathcal{T}$ is the ordered set of terms that we can make from $x_1, \ldots, x_n$. For $P \in \mathbb{K}[x_1, \ldots, x_n]$, we let $\mathcal{T}(P)$ denote the set of terms appearing in P and $\mathrm{LT}(P) = \mathrm{LT}_{\prec}(P)$ is the maximum of $\mathcal{T}(P)$. For any $D \in \mathbb{N}$, $\mathcal{T}_D$ is the ordered set of all terms of degree less than or equal to D sorted by increasing order (wrt. $\prec$).

From the algorithm point of view, it is impossible for us to check that a relation is valid for all $\mathbf{i} \in \mathbb{N}^n$. Indeed, at some point of the algorithm we will have a finite subset of indices $T \subset \mathbb{N}^n$ and we will try to find relations that are valid for those indices:

$$\forall\, \mathbf{i} \in T,\ u_{\mathbf{i}+\mathbf{d}} + \sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}}\, u_{\mathbf{i}+\mathbf{k}} = 0.$$

Definition 4. *Let T be a finite subset of $\mathbb{N}^n$. We say that a polynomial $P \in \mathbb{K}[x_1, \ldots, x_n]$ is valid up to T if $\mathrm{Rel}_{\mathbf{u}}(P)(\mathbf{i}) = \left[\mathbf{x}^{\mathbf{i}}\, P\right]_{\mathbf{u}} = 0$ for all $\mathbf{i} \in T$. In that case we write that $\mathrm{NF}(P, \mathbf{u}, T) = 0$.*

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 $[t\, P] = 0$ for all $t \in T$. In that case we write that $\mathrm{NF}(P, \mathbf{u}, T) = 0$.

From the relations $u_{\mathbf{i}+\mathbf{d}} + \sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}}\, u_{\mathbf{i}+\mathbf{k}} = 0$, valid for all $\mathbf{i} \in T$, we can make the polynomial $P = \mathbf{x}^{\mathbf{d}} + \sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}}\, \mathbf{x}^{\mathbf{k}}$. This polynomial satisfies $[\mathbf{x}^{\mathbf{i}}\, P]_{\mathbf{u}} = 0$ for all $\mathbf{i} \in T$. In other words, we have $\mathrm{NF}(P, \mathbf{u}, T) = 0$.

Example 5. *Let $T = \{(0,0),(0,1),(1,0),(0,2)\}$ be a finite ordered subset of $\mathbb{N}^2$ and let $\mathbf{u} = ((-1)^{i\,j})_{(i,j)\in\mathbb{N}^2}$. Finding $\alpha_{0,0}, \alpha_{1,0}, \alpha_{0,1}$ such that*

$$\begin{cases} u_{0,2} + \alpha_{0,0}\, u_{0,0} + \alpha_{0,1}\, u_{0,1} + \alpha_{1,0}\, u_{1,0} & = 1 + \alpha_{0,0} + \alpha_{0,1} + \alpha_{1,0} = 0\\ u_{0,3} + \alpha_{0,0}\, u_{0,1} + \alpha_{0,1}\, u_{0,2} + \alpha_{1,0}\, u_{1,1} & = 1 + \alpha_{0,0} + \alpha_{0,1} - \alpha_{1,0} = 0\\ u_{1,2} + \alpha_{0,0}\, u_{1,0} + \alpha_{0,1}\, u_{1,1} + \alpha_{1,0}\, u_{2,0} & = 1 + \alpha_{0,0} - \alpha_{0,1} + \alpha_{1,0} = 0\\ u_{0,4} + \alpha_{0,0}\, u_{0,2} + \alpha_{0,1}\, u_{0,3} + \alpha_{1,0}\, u_{1,2} & = 1 + \alpha_{0,0} + \alpha_{0,1} + \alpha_{1,0} = 0 \end{cases}$$

yields $\alpha_{0,0} = -1$ and $\alpha_{1,0} = \alpha_{0,1}$. Hence, the polynomial $P = y^2 - 1$ is such that for all $(i, j) \in T$, $[x^i\, y^j\, P]_{\mathbf{u}} = 0$, i.e. $\mathrm{NF}(P, \mathbf{u}, T) = 0$.

4.1. Staircase

We assume now that $T \subset \mathcal{T}$ 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 a new relation can be found if for a finite subset $\mathcal{K}$ of $\mathbb{N}^n$, the following two properties are satisfied:

(a) There are scalars $\alpha_{\mathbf{k}} \in \mathbb{K}, \mathbf{k} \in \mathcal{K}$ such that $\forall\, \mathbf{i} \in T', u_{\mathbf{i}+\mathbf{d}} + \sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}}\, u_{\mathbf{i}+\mathbf{k}} = 0$;
(b) There are no nonzero relations $\sum_{\mathbf{k}\in\mathcal{K}} \beta_{\mathbf{k}}\, u_{\mathbf{i}+\mathbf{k}} = 0$ which are valid for all $\mathbf{i} \in T'$.

This would lead to the relation $u_{\mathbf{i}+\mathbf{d}} + \sum_{\mathbf{k}\in\mathcal{K}} \alpha_{\mathbf{k}}\, u_{\mathbf{i}+\mathbf{k}} = 0$.

We can translate these properties in polynomial terms:

(a) There is a monic polynomial $P \in \mathbb{K}[\mathbf{x}]$ of leading term $\mathbf{x}^{\mathbf{d}}$ s.t. $\mathrm{NF}(P, \mathbf{u}, T) = 0$;
(b) There are no nonzero relations $\sum_{t\in\, \mathcal{T}(P-\mathbf{x}^{\mathbf{d}})} \beta_t\, \mathrm{Rel}_{\mathbf{u}}(t)(\mathbf{i}) = 0$ which are valid for all $\mathbf{i} \in T'$. Equivalently, there are no nonzero relations $\sum_{t\in\, \mathcal{T}(P-\mathbf{x}^{\mathbf{d}})} \beta_t\, [m\, t] = 0$ which are valid for all $m \in T$.

In other words, we need to identify a set of terms for which there is *no* linear relations.

Definition 5. *Let T be a finite subset of $\mathcal{T}$. We say that a finite set $S \subset T$ of terms is a useful staircase wrt.* $\mathbf{u}$*, T and $\prec$ if*

$$\sum_{t\in S} \beta_t\, [m\, t] = 0, \quad \forall\, m \in S$$

implies that $\beta_t = 0$ for all $t \in S$, S is maximal for the inclusion and minimal for $\prec$. We compare two ordered sets for $\prec$ by seeing them as tuples of their elements and then comparing them lexicographically.

Let us mention that these "useful staircases" are not necessarily staircases in the sense of Gröbner bases since they are not always stable under division.

Example 6. *In dimension* 1*, consider the set $T = \{1, x, x^2\}$ of monomials of degree less than or equal to* 2 *and the sequence $\mathbf{u} = (u_i)_{i\in\mathbb{N}}$ defined by*

$$u_i = \begin{cases} 0, & \text{if } i \leq 2, \\ i - 2, & \text{otherwise.} \end{cases}$$

For $m \in T$, we have the following potential relations

$$\begin{cases} \alpha_1\, [1] + \alpha_x\, [x] + \alpha_{x^2}\, [x^2] & = 0 \\ \alpha_1\, [x] + \alpha_x\, [x^2] + \alpha_{x^2}\, [x^3] & = \alpha_{x^2} = 0 \\ \alpha_1\, [x^2] + \alpha_x\, [x^3] + \alpha_{x^2}\, [x^4] & = \alpha_x + 2\, \alpha_{x^2} = 0. \end{cases}$$

Therefore, the useful staircase is $\{x, x^2\}$ *and this set is not stable under division because* 1 *is not inside. Using the stability criterion, we can recover the complete staircase* $\{1, x, x^2\}$.

Notice though, that if T *is big enough, the useful staircase shall coincide with the classical staircase: let us take this time* $T = \{1, x, x^2, x^3\}$ *and the same sequence. The potential relations become*

$$\begin{cases} \alpha_1\,[1] + \alpha_x\,[x] + \alpha_{x^2}\,[x^2] + \alpha_{x^3}\,[x^3] & = \alpha_{x^3} = 0 \\ \alpha_1\,[x] + \alpha_x\,[x^2] + \alpha_{x^2}\,[x^3] + \alpha_3\,[x^4] & = \alpha_{x^2} + 2\,\alpha_{x^3} = 0 \\ \alpha_1\,[x^2] + \alpha_x\,[x^3] + \alpha_{x^2}\,[x^4] + \alpha_{x^3}\,[x^5] & = \alpha_x + 2\,\alpha_{x^2} + 3\,\alpha_{x^3} = 0 \\ \alpha_1\,[x^3] + \alpha_x\,[x^4] + \alpha_{x^2}\,[x^5] + \alpha_{x^3}\,[x^6] & = \alpha_1 + 2\,\alpha_x + 3\,\alpha_{x^2} + 4\,\alpha_{x^3} = 0, \end{cases}$$

and the useful staircase is $\{1, x, x^2, x^3\}$.

Algorithm 2 transforms a useful staircase into a staircase.

Algorithm 2: STABILIZE
Input: A useful staircase S
Output: A staircase
$S' := \varnothing$
For $s \in S$ **do** $S' := S' \cup \{t \mid t \in \mathcal{T} \text{ and } t|s\}$.
Return S'.

4.2. Linear Algebra to find relations

We design a simple algorithm for checking that a finite set $S \subseteq T$ is a useful staircase wrt. $\mathbf{u}$, T and $\prec$. Let us first start with an example:

Example 7. *We consider the sequence* $\mathbf{u} = (u_{i,j})_{(i,j)\in\mathbb{N}^2}$ *defined by*

$$u_{i,j} = \begin{cases} 1, & \text{if } i = 1, j = 2, \\ 0, & \text{otherwise}. \end{cases}$$

and we look for a relation $P(x, y) = \alpha_{x^2}\,x^2 + \alpha_{x\,y}\,x\,y + \alpha_{y^2}\,y^2 + \alpha_x\,x + \alpha_y\,y + \alpha_1$. *That is, we try to find* $(\alpha_s)_{\deg s \leq 2}$ *s.t.* $[t\,P] = 0$ *for all* $t \in T$:

$$\forall (i, j) \in \mathbb{N}^2,\ i+j \leq 2,\ \alpha_1\,u_{i,j} + \alpha_y\,u_{i,j+1} + \alpha_x\,u_{i+1,j} + \alpha_{y^2}\,u_{i,j+2} + \alpha_{x\,y}\,u_{i+1,j+1} + \alpha_{x^2}\,u_{i+2,j} = 0.$$

Finding a useful staircase S is equivalent to extracting a full rank matrix with as many low-labeled columns as possible in the following multi-Hankel matrix:

$$H_T = \begin{array}{c} \\ 1 \\ y \\ x \\ y^2 \\ xy \\ x^2 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & y & x & y^2 & xy & x^2 \end{array} \\ \begin{pmatrix} u_{0,0} & u_{0,1} & u_{1,0} & u_{0,2} & u_{1,1} & u_{2,0} \\ u_{0,1} & u_{0,2} & u_{1,1} & u_{0,3} & u_{1,2} & u_{2,1} \\ u_{1,0} & u_{1,1} & u_{2,0} & u_{1,2} & u_{2,1} & u_{3,0} \\ u_{0,2} & u_{0,3} & u_{1,2} & u_{0,4} & u_{1,3} & u_{2,2} \\ u_{1,1} & u_{1,2} & u_{2,1} & u_{1,3} & u_{2,2} & u_{3,1} \\ u_{2,0} & u_{2,1} & u_{3,0} & u_{2,2} & u_{3,1} & u_{4,0} \end{pmatrix} \end{array} = \begin{array}{c} \\ 1 \\ y \\ x \\ y^2 \\ xy \\ x^2 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & y & x & y^2 & xy & x^2 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

In this example, the columns labeled with $1, x^2$ are clearly linearly dependent from the columns labeled with lower terms so that $S = \{y, x, y^2, xy\}$ is the useful staircase and $\det(H_S) = 1 \neq 0$. Furthermore, we can now try to find a relation $Q(x,y) = x^2 - \alpha_{xy}\,xy - \alpha_{y^2}\,y^2 - \alpha_x\,x - \alpha_y\,y$. Again this is equivalent to finding $\alpha_y, \ldots, \alpha_{xy}$ s.t. $\mathrm{Rel}(Q)(i,j) = 0$ for all $(i,j) \in \mathbb{N}^2$ with $i + j \leq 2$. Finally, this comes down to solving the linear system:

$$H_S \begin{pmatrix} \alpha_y \\ \alpha_x \\ \alpha_{y^2} \\ \alpha_{xy} \end{pmatrix} = \begin{pmatrix} u_{2,1} \\ u_{3,0} \\ u_{2,2} \\ u_{3,1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Since H_S is full rank we find $\alpha_y = \alpha_x = \alpha_{x^2} = \alpha_{xy} = 0$ and the relation $Q(x,y) = x^2$.

For the more general case of two sets of terms, we are now in a position to define the structured matrix associated to these sets.

Definition 6. *Let T and S be two finite subsets of $\mathcal{T}$. We consider the polynomial $P_S(\mathbf{x}) = \sum_{s \in S} a_s\, s$ and the linear equations $[t\, P_S] = 0$ for all $t \in T$. Then, we generate the coefficient matrix $H_{T,S}$ from the previous linear system of equations in the unknown variables a_s for $s \in S$:*

$$H_{T,S} = \begin{array}{c} \\ \vdots \\ t \in T \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccc} \cdots & s \in S & \cdots \end{array} \\ \begin{pmatrix} \ddots & \vdots & \iddots \\ \cdots & [s\,t]_{\mathbf{u}} & \cdots \\ \iddots & \vdots & \ddots \end{pmatrix} \end{array}.$$

When $T = S$ we simply write H_T for the multi-Hankel matrix $H_{T,S}$.

The following two computations are the key of our algorithms and make use of linear algebra techniques:

(a) Assuming two finite sets S and T are such that $\#T \geq \#S$, then checking that $S \subset \mathcal{T}$ is a useful staircase wrt. T is equivalent to checking that the matrix $H_{T,S}$ has full rank;

(b) Given a finite set of terms T, computing a monic polynomial $P \in \mathbb{K}[\mathbf{x}]$ of support $\mathcal{T}(P)$ s.t. $\mathrm{NF}(P, \mathbf{u}, T) = 0$ is equivalent to solving a linear system

$$H_{T,S} \times \mathbf{a} + H_{T,\{\mathrm{LT}(P)\}} = 0,$$

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

Proposition 8. *Let T be a finite subset of $\mathcal{T}$. Let S be a subset of T, then* $\operatorname{rank} H_T \geq \operatorname{rank} H_S$.

Furthermore, if S is a useful staircase wrt. $\mathbf{u}$*, T and $\prec$ then:*

$$\det(H_S) \neq 0 \textit{ and } \operatorname{rank} H_S = \operatorname{rank} H_{T,S} = \operatorname{rank} H_T.$$

Proof. The first statement is clear for H_S is a submatrix of H_T.

The second statement is another wording of Definition 5. □

4.3. An FGLM*-like Algorithm*

The algorithm we want to design cannot check all the infinitely many elements of the sequence $\mathbf{u}$. That is why we need a bound $d \geq 0$ given by the user on the order of the sequence. Let us recall that the *order* of a linear recursive sequence is the size of the staircase of a Gröbner basis of its ideal of relations. This also means that if the order is d, no monomials of degree greater or equal to d appear in the staircase. We let T be the set of all monomials of degree less than d and $\prec$ be an admissible monomial ordering.

Calling the BMS algorithm on table $\mathbf{u}$ and bound $d \geq$ returns a d-truncated Gröbner basis of the ideal of relations of $\mathbf{u}$. We shall try to adapt existing Gröbner basis algorithms to obtain the same result. To this end, we can try to slightly modify the FGLM algorithm, Faugère et al. (1993). Let us notice that when running the FGLM algorithm, the user already knows the quotient ring structure thanks to the input Gröbner basis and its staircase. In this scalar case, we aim to determine the structure of the quotient ring and therefore cannot rely on this knowledge during the computation. This is a fundamental difference between both situations.

To ease the presentation, we shall split our algorithm in two steps. The first step is devoted to the staircase computation wrt. to $\prec$, the monomial ordering, and to d, the bound on the degree. The second step concernes the d-truncated Gröbner basis computation. It is worth mentioning that in a real implementation, both steps should be combined to increase the efficiency of the algorithm.

Algorithm 3: SCALAR-FGLM.

Input: A table $\mathbf{u} = (u_{\mathbf{i}})_{\mathbf{i}\in\mathbb{N}^n}$ with coefficients in $\mathbb{K}$, d a given bound and $\prec$ a monomial ordering.

Output: A reduced $(d+1)$-truncated Gröbner basis wrt. $\prec$ of the ideal of relations of $\mathbf{u}$.

Build the matrix $H_{\mathcal{T}_d}$.
Extract a submatrix of maximal rank. `// as in Section 4.2`
Find S the useful staircase s.t. $\operatorname{rank} H_{\mathcal{T}_d} = \operatorname{rank} H_S$.
$S' := \operatorname{Stabilize}(S)$. `// the staircase (stable under division)`
$L := \mathcal{T}_{d+1} \setminus S'$. `// set of next terms to study`
$G := \varnothing$. `// the future Gröbner basis`
While $L \neq \varnothing$ **do**
  $t := \min_{\prec}(L)$ and remove t from L.
  Find $\boldsymbol{\alpha}$ s.t. $H_S\, \boldsymbol{\alpha} + H_{S,\{t\}} = 0$.
  $G := G \cup \{t + \sum_{s\in S} \alpha_s\, s\}$.
  Sort L by increasing order (wrt. $\prec$) and remove multiples of $\operatorname{LT}(G)$.
Return G.

Example 8. *Let us trace Algorithm 3 on Example 2, item 3.*

We consider the sequence $\mathbf{b} = (b_{i,j})_{(i,j)\in\mathbb{N}^2}$ *defined by* $b_{i,j} = \binom{i}{j}$ *and we fix* $d = 2$, $\prec$ *the* DRL *ordering with* $y \prec x$, *i.e.* $x^i\, y^j \prec x^k\, y^\ell$ *if and only if*

$$(i + j < k + \ell) \vee ((i + j = k + l) \wedge (j > \ell)).$$

Then $\mathcal{T}_2 = \{1, y, x, y^2, x\,y, x^2\}$ *and the matrix* $H_{\mathcal{T}_2}$ *is as follows*

$$H_{\mathcal{T}_2} = \begin{array}{c} \\ 1 \\ y \\ x \\ y^2 \\ x\,y \\ x^2 \end{array}\begin{array}{c} \begin{array}{cccccc} 1 & y & x & y^2 & x\,y & x^2 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 2 \\ 1 & 1 & 1 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 2 & 0 & 1 & 3 \\ 1 & 2 & 1 & 1 & 3 & 1 \end{pmatrix} \end{array}.$$

We can verify easily that the column (resp. row) labeled with $x\,y$ *is the sum of the first and second columns (resp. rows) labeled with* 1 *and* y *and that the other columns (resp. rows) are all linearly independent. Therefore, the useful staircase is* $S = \{1, y, x, y^2, x^2\}$. *As* S *is stable under division,* $S' = \operatorname{Stabilize}(S) = S$ *and we initialize* L *to the ordered set* $\{x\,y, y^3, x\,y^2, x^2\,y, x^3\}$.

*In the **while** loop, we start with $t = xy$ and we need to solve the linear system*

$$H_S\,\boldsymbol{\alpha} + \begin{pmatrix}1\\0\\2\\0\\3\end{pmatrix} = 0 \quad\Rightarrow\quad \boldsymbol{\alpha} = \begin{pmatrix}-1\\-1\\0\\0\\0\end{pmatrix}.$$

Hence $G = \{xy - y - 1\}$. We update $L = \{y^3, x^3\}$. Then we take $t = y^3$, find $G = \{xy - x - y, y^3\}$ and update $L = \{x^3\}$.

Finally, setting $t = x^3$, we find $G = \{xy - x - y, y^3, x^3 - 3x^2 + 3x - 1\}$ and update $L = \varnothing$.

We showed in Example 2 that G is not a Gröbner basis of the ideal of relations but only a 3-truncated Gröbner basis of the ideal. The reason is twofold: the Gröbner basis of $\langle G\rangle$ is 1 and the true ideal of relations is only $\langle xy - y - 1\rangle$.

Theorem 9. *Algorithm 3 is correct, terminates and its output 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 Gröbner basis of the ideal of relations of the sequence.*

Proof. Algorithm 3 clearly terminates since the size of L decreases at each step.

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: any polynomial with leading term outside the staircase of the ideal of **u** reduces to a polynomial with support in the staircase.

Let us denote $\mathbb{K}$ the field of coefficients of **u** and $I = \langle G\rangle \subseteq \mathbb{K}[x]$ where G is computed by Algorithm 3. Suppose that one element e of the useful staircase lies outside the staircase. On the one hand as e is not in the staircase S, one can find $\alpha_s \in \mathbb{K}, s \in S$ such that $e - \sum_{s\in S} \alpha_s\, s = 0$ in $\mathbb{K}[\mathbf{x}]/I$. Therefore, the column labeled with e is a linear combination of columns labeled with lower terms $s, s \in S$ with coefficients α_s. On the other hand, as e is in the useful staircase, column labeled with e should be linearly independent from the previous ones, which are all labeled with terms lower than e by construction of the matrix. This is a contradiction.

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 that 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 $\mathcal{T}_{d+1}$) and their S-polynomials $S(f,g)$. Then either the leading term of $S(f,g)$ is in $\mathcal{T}_{d+1} \setminus S$ or it is in S. In the latter case, it means there is a relation in H_S, 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. When d is the order of the linear recursive sequence, then it is also the size of the staircase and the result is a Gröbner basis. □

Given a useful staircase S, when counting the number of table queries, $2S = \{s\,t \mid s,t \in S\}$ must not be too big compared to S. We shall see how to bound the cardinality of $2S$ in Section 5.1.

5. Adaptive Algorithm

Sections 3 and 4 were devoted to the designs of two algorithms to recover the relations of a table.

The main drawback of the Algorithm 1 is the number of table queries. Indeed, in the generic case where the new ideal is in shape position, running the BM algorithm on the new table means accessing all elements of index $(0,\dots,0,i_{n+1})$ with $0 \leq i_{n+1} \leq 2\,d-1$. These elements are obtain from all the elements $u_{\mathbf{i}}$, $\mathbf{i} \in \mathbb{N}^n$ of index degree $|\mathbf{i}| \leq 2\,d-1$ in the original table $\mathbf{u}$. Hence, we need to access all the $\binom{n+2\,d}{2\,d-1}$ terms $u_{\mathbf{i}}$, $\mathbf{i} \in \mathbb{N}^n$ with $|\mathbf{i}| \leq 2\,d-1$.

Likewise, Algorithm 3 is efficient when the given bound on the degree of the polynomials is small compared to the order of the sequence: otherwise we need to access once again all the elements of order the order of the sequence by construction of the matrix. In other word, if d is the order of the recurrence of $\mathbf{u}$ then both algorithms are efficient whenever

$$\max\deg G \approx d^{1/n}.$$

Unfortunately, this is not always the case, especially if the monomial ordering is a LEX ordering and using Algorithms 1 or 3 on these examples would increase too much the number of accesses to the table. In this section, we design an adaptive algorithm taking into account the shape of the final Gröbner basis.

As the strategy we shall use is similar to the one of the FGLM algorithm, we shall start by recalling how it works. Given a zero-dimensional ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ and a Gröbner basis $\mathcal{G}_1$ of I for the monomial ordering $\prec_1$, the FGLM algorithm computes a Gröbner basis $\mathcal{G}_2$ of I for the ordering $\prec_2$. To this end, the algorithm has an ordered (for $\prec_2$) set of monomials L to check and a staircase S. At the beginning, $L = \{1\}$ and $S = \mathcal{G}_2 = \varnothing$. At each step, the algorithm selects m the smallest monomial of L and removes it from L: if m reduces modulo I (thanks to $\mathcal{G}_1$) to a linear combination of monomials in S, then a new polynomial is found and added to $\mathcal{G}_2$, all multiples of m are removed from L. Otherwise, m is added to S

and all the monomials $x_1\, m, \dots, x_n\, m$, that are not multiples of leading monomials of $\mathcal{G}_2$, are added to L. The algorithm stops when L is empty.

Example 9. *This example is a continuation of Example 4.*

Let us consider the Gröbner basis $\mathcal{G}_1 = \{t^4 - 4\,t^3 - 4\,t^2 + 16\,t, x_1 + \frac{1}{6}\,t^3 - \frac{1}{2}\,t^2 - \frac{2}{3}\,t + 1, x_2 - \frac{1}{12}\,t^3 - \frac{1}{4}\,t^2 - \frac{5}{6}\,t + 1\}$ *of the ideal* $I = \langle \mathcal{G}_1 \rangle$ *for the monomial ordering* LEX *with* $t \prec_1 x_1 \prec_1 x_2$. *We aim to compute* $\mathcal{G}_2$ *a Gröbner basis of* I *for the* LEX *ordering with* $x_1 \prec_2 x_2 \prec_2 t$. *We let* $L = \{1\}, S = \mathcal{G}_2 = \varnothing$.

We start with 1 *which does not reduce modulo* $\mathcal{G}_1$. *Then* $L = \{x_1, x_2, t\}$ *and* $S = \{1\}$.

Next, x_1 *reduces modulo* $\mathcal{G}_1$ *to* $-\frac{1}{6}\,t^3 + \frac{1}{2}\,t^2 + \frac{2}{3}\,t - 1$, *but no nonzero linear combination of* $1, x_1$ *can reduce to* 0. *Therefore, we update* $L = \{x_1^2, x_2, x_1\,x_2, t, x_1\,t\}$, $S = \{1, x_1\}$.

Then, x_1^2 *reduces modulo* $\mathcal{G}_1$ *to* 1. *Fortunately, now the polynomial* $x_1^2 - 1$ *reduces to* 0 *modulo* $\mathcal{G}_1$. *Hence* $\mathcal{G}_2 = \{x_1^2 - 1\}$ *and* $L = \{x_2, x_1\,x_2, t, x_1\,t\}$.

Next, x_2 *is selected and reduces modulo* $\mathcal{G}_1$ *to* $-\frac{1}{12}\,t^3 + \frac{1}{4}\,t^2 + \frac{5}{6}\,t - 1$. *Likewise, no nonzero linear combination of* $1, x_1, x_2$ *can reduce to* 0. *We update* $L = \{x_1\,x_2, x_2^2, t, x_1\,t, x_2\,t\}$ *and* $S = \{1, x_1, x_2\}$.

Then, $x_1\,x_2$ *reduces modulo* $\mathcal{G}_1$ *to* $-\frac{1}{4}\,t^2 + \frac{1}{2}\,t + 1$, *but no nonzero linear combination of* $1, x_1, x_2, x_1\,x_2$ *can reduce to* 0. *Therefore, we update* $L = \{x_2^2, x_1\,x_2^2, t, x_1\,t, x_2\,t, x_1\,x_2\,t\}$, $S = \{1, x_1, x_2, x_1\,x_2\}$.

Next, x_2^2 *is selected and reduces modulo* $\mathcal{G}_1$ *to* 1. *Thus the polynomial* $x_2^2 - 1$ *reduces to* 0. *Therefore,* $\mathcal{G}_2$ *is updated to* $\{x_1^2 - 1, x_2^2 - 1\}$ *and* L *to* $\{t, x_1\,t, x_2\,t, x_1\,x_2\,t\}$.

Finally, t *does not reduce modulo* $\mathcal{G}_1$ *but* $t + x_1 - 2\,x_2 - 1$ *reduces to* 0. *Thus,* $\mathcal{G}_2$ *is updated to* $\{x_1^2 - 1, x_2^2 - 1, t + x_1 - 2\,x_2 - 1\}$ *and* L *to* $\varnothing$.

The algorithms returns $\mathcal{G}_2$ *and the ideal of relations of the sequence* $((-1)^{i_1\,i_2})_{(i_1,i_2)\in\mathbb{N}^2}$ *is the ideal spanned by the polynomials in* $\mathcal{G}_2$ *that are not in* t, *that is* $\langle x_1^2 - 1, x_2^2 - 1\rangle$.

In the FGLM algorithm, when we discover a relation

$$f = t + \sum_{s \in S} \alpha_s\, s$$

then we *know* for all $m \in \mathcal{T}$, $m\,f$ is still a valid relation. In constrast, in Algorithm 3, finding a relation

$$[f]_{\mathbf{u}} = [t]_{\mathbf{u}} + \sum_{s \in S} \alpha_s\, [s]_{\mathbf{u}} = 0 \tag{1}$$

need not imply $[m\,f]_{\mathbf{u}} = 0$. In fact, in the 1-dimensional BM algorithm, this is the reason why the relation must be updated.

Reconsidering Section 2.3, we can see that any n-dimensional linear recursive sequence $\mathbf{u}$ can be written as

$$\forall\, i \in \mathbb{N}^n,\ u_{\mathbf{i}} = \langle \mathbf{r}, T_1^{i_1} \cdots T_n^{i_n} \cdot \mathbf{1} \rangle,$$

where $\mathbf{r}$ is a vector depending on the initial conditions and T_i are multiplication matrices associated to the Gröbner basis G. Therefore, relation (1) can be rewritten as

$$\langle \mathbf{r}, \mathrm{NormalForm}(f, G) \rangle = 0. \tag{2}$$

Therefore, if $\mathbf{r}$ is random enough, we deduce that $\mathrm{NormalForm}(f, G) = 0$ so that $\mathrm{NormalForm}(m\, f, G) = 0$ for all $m \in \mathcal{T}$ which implies that $[m\, f]_{\mathbf{u}} = 0$.

We would like to point out that in some applications, it is possible to check afterwards that the relation is correct, we refer for instance to Remark 15 in Section 5.2. Accordingly, we design Algorithm 4 which is an FGLM-like algorithm using this property. Let us also recall that, from Section 2.2, we know that only Gorenstein ideals of relations can be recovered in this framework. This gives another probabilistic test for the Gorenstein property, see Daleo and Hauenstein (2015).

We proceed term by term to discover the new staircase. This is equivalent to increasing the rank of the multi-Hankel matrix by 1. We start with matrix $H_\varnothing$ and proceed by induction. Assuming we found a subset of term $S \subseteq \mathcal{T}$ such that H_S has full rank, we set t to be the minimum of $\mathcal{T} \setminus S$. If $\operatorname{rank} H_{S \cup \{t\}} = \operatorname{rank} H_S + 1$, then S is updated to $S \cup \{t\}$, otherwise we consider t' the minimum of $\mathcal{T} \setminus (S \cup \{t\})$.

Instead of taking a bound on the degrees of the polynomials, this algorithm takes a lower bound on the size of the staircase. It also ensures to return a truncated Gröbner basis whose staircase has size at least this lower bound.

We shall 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 set of terms G, $\mathrm{MinGBasis}(G)$ is the corresponding minimal Gröbner basis.

Remark 10. *Depending on the origin of the sequence* $\mathbf{u}$*, it is possible that* $u_{\mathbf{i}}$ *has no meaning whatsoever for certain* $\mathbf{i}$*. For instance, in error correcting codes, see e.g. Section 5.3, there exists a bound* $B \in \mathbb{N}$ *such that* $u_{\mathbf{i}}$ *cannot be computed for* $\mathbf{i} = (i_1, \ldots, i_n)$ *with* $i_k > B$*. It suffices to change two lines: the first one is* $L := L \cup \{x_i\, t \mid i = 1, \ldots, n\} \setminus \{t\}$ *where only monomials* $x_i\, t$*, with* $[x_i\, t]_{\mathbf{u}}$ *computable, should be added to L. The second one is solving the system* $H_S\, \boldsymbol{\alpha} + H_{S,\{t'\}} = 0$ *and updating* $G := G \cup \{t' + \sum_{s \in S} \alpha_s\, s\}$ *which should be skipped as soon as a term* $[s\, t']_{\mathbf{u}}$ *cannot be computed for* $s \in S$*.*

Algorithm 4: Adaptive Scalar-FGLM (simple version).

Input: A table $\mathbf{u} = (u_{\mathbf{i}})_{\mathbf{i}\in\mathbb{N}^n}$ with coefficients in $\mathbb{K}$, d a given bound and $\prec$ a monomial ordering.

Output: A reduced Gröbner basis of a zero-dimensional ideal of degree $\geq d$.

$L := \{1\}$. `// set of next terms to study`

$S := \varnothing$. `// the useful staircase wrt. the new ordering ≺`

$G' := \varnothing$. `// leading terms of the final Gröbner basis`

While $L \neq \varnothing$ **do**

- $t := \min_{\prec}(L)$.
- **If** $H_{S\cup\{t\}}$ *is full rank* **then**
  - $S := S \cup \{t\}$ and $L := L \cup \{x_i\, t \mid i = 1, \dots, n\} \setminus \{t\}$.
  - Remove multiples of elements of G' in L.
  - **If** $\# S \geq d$ **then** `// early termination`
    - $G := \varnothing$ and $G' := \text{MinGBasis}(G' \cup L \cup \mathcal{T}_{\deg t+1} \setminus S)$.
    - **For all** $t' \in G'$ **do**
      - Find $\boldsymbol{\alpha}$ s.t. $H_S\, \boldsymbol{\alpha} + H_{S,\{t'\}} = 0$.
      - $G := G \cup \{t' + \sum_{s\in S} \alpha_s\, s\}$.
    - **Return** S and G.
- **Else**
  - $G' := G' \cup \{t\}$ and remove multiples of t in L.

Error “Run Algorithm 3”.

Proposition 11. *Let S and G be the output of Algorithm 4. Then S is a staircase of size $\geq d$ and G is a set of valid relations, that is to say $\mathrm{NF}(f, \mathbf{u}, S) = 0$ for all $f \in G$.*

Example 10. *We give the trace of the algorithm on Example 2, item 1, i.e., for the table $\mathbf{u} = ((2^i + 3^i)\, 7^j)_{(i,j)\in\mathbb{N}^2}$, with bound $d = 2$ and $\prec$ the DRL ordering with $y \prec x$.*

We start with $L = \{1\}, S = \varnothing, G' = \varnothing$. Setting $t = 1$ yields the matrix $H = (\,2\,)$ with max rank. Thus $S = \{1\}$ and we update $L = \{y, x\}$.

Now, we set $t = y$ and get the matrix $\begin{pmatrix} 2 & 14 \\ 14 & 98 \end{pmatrix}$ of rank 1, hence $G' = \{y\}$ and $L = \{x\}$.

Letting $t = x$ yields the matrix $\begin{pmatrix} 2 & 5 \\ 5 & 13 \end{pmatrix}$ of rank 2 which let us update $S = \{1, x\}$ and $L = \{x^2\}$. As $\# S \geq d$, we reach the early termination part of the algorithm.

We have $G' = \mathrm{MinGBasis}(\{y\} \cup \{x^2\} \cup \{y^2, x\,y, x^2\} \setminus \{1, x\}) = \{y, x^2\}$. Solving the two linear systems yields $G = \{y - 7, x^2 - 5\,x + 6\}$.

Remark 12. *It is worth mentioning that on the one hand if d is set too small, since Algorithm 3 does not check whether a relation is valid on the following terms or not, it might return clearly a wrong result. On the other hand, for greater d, it might return an error coming from the wrong relations setting $L = \varnothing$ but yielding a staircase of size less than d.*

For instance, on table $((-1)^{i\,j})_{(i,j)}$, Algorithm 4 shall produce $\{y - 1, x - 1\}$ when d is set to 1 and an error if d is set to 2 or more. Yet, the ideal of relations is $\langle y^2 - 1, x^2 - 1\rangle$.

This remark motivates an extension of Algorithm 4. As said before, its drawback is that given a computed staircase S, it only checks elements at distance 1 of S, i.e. elements of form $x_i\, s, s \in S$ to find new relations. Following algorithm extends this behavior to check elements at distance e, where e is an input parameter, i.e. elements $\mathbf{x}^{\mathbf{i}}\, s, s \in S$ with $|\mathbf{i}| \leq e$.

Example 11. *Let us detail how Algorithm 5 works on $\mathbf{u} = ((-1)^{i\,j})_{(i,j)\in\mathbb{N}^2}$ with parameters $d = 4$ and $e = 2$.*

The sets L, S, G' are initialized with $L = \{1, y, x\}$, $S = G' = \varnothing$ while $t = \{1, y\}$ and the matrix $H = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ has rank 1. Hence $S = \{1\}$, $L = \{y, x, y^2, x\,y, x^2\}$.

Now $t = \{y, x\}$ and the matrix $H = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix}$ has rank 3. Hence $S = \{1, y, x\}$, $L = \{y^2, x\,y, x^2, y^3, x\,y^2, x^2\,y, x^3\}$.

Taking $t = \{y^2, x\,y\}$, the matrix $H = \begin{pmatrix} 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \end{pmatrix}$ has rank 4 with clearly the 4th column, labeled with y^2, linearly dependent from the previous ones, thus $S = \{1, y, x, x\,y\}$ and $L = \{y^2, x^2, y^3, x\,y^2, x^2\,y, x^3, x\,y^3, x^2\,y^2, x^3\,y\}$.

Algorithm 5: EXTENDED ADAPTIVE SCALAR-FGLM (simple version).

Input: A table $\mathbf{u} = (u_{\mathbf{i}})_{\mathbf{i}\in\mathbb{N}^n}$ with coefficients in $\mathbb{K}$, d a given bound, e the maximal distance for the allowed neighbors and $\prec$ a monomial ordering.

Output: A reduced Gröbner basis of a zero-dimensional ideal of degree $\geq d$.

$L := \{\mathbf{x}^{\mathbf{i}} \mid |\mathbf{i}| \leq e\}$. `// set of next terms to study`

$S := \varnothing$. `// the useful staircase wrt. the new ordering ≺`

$G' := \varnothing$. `// leading terms of the final Gröbner basis`

While $L \neq \varnothing$ **do**

- $e' := \min(e, \# L)$.
- $t := \{t_0, \ldots, t_{e'-1} \mid t_0 \prec \cdots \prec t_{e'-1},\ \forall u \in L \setminus \{t_0, \ldots, t_{e'-1}\}, t_{e'-1} \prec u\}$.
- **If** $\operatorname{rank} H_{S \cup \{t_0,\ldots,t_{e'-1}\}} > \operatorname{rank} H_S$ **then**
  - $S' := S$, $S :=$ usefulStaircase$(S \cup \{t_0, \ldots, t_{e'-1}\})$.
  - $L := L \cup \left\{\mathbf{x}^{\mathbf{i}}\, u \,\middle|\, |\mathbf{i}| \leq e, u \in t\right\} \setminus S$.
  - Remove multiples of elements of G' in L.
  - **If** $\# S \geq d$ **then** `// early termination`
    - $G := \varnothing$ and $G' :=$ MinGBasis$\left(\left(G' \cup L \cup \bigcup_{t' \in S \setminus S'} \mathcal{T}_{\deg t'+1}\right) \setminus \text{Stabilize}(S)\right)$.
    - **For all** $t' \in G'$ **do**
      - Find α s.t. $H_S\, \alpha + H_{S,\{t'\}} = 0$.
      - $G := G \cup \{t' + \sum_{s \in S} \alpha_s\, s\}$.
    - **Return** S and G.
- **Else**
  - $G' := G' \cup \{t_0\}$ and remove multiples of t_0 of degree at least $\deg t_0 + e$ in L.

Error "Run Algorithm 3".

The staircase size being greater than 4*, we find* $G' = \{y^2, x^2\}$*. Solving the linear systems yields* $G = \{y^2 - 1, x^2 - 1\}$*.*

Remark 13. *The only difference between the* Extended Adaptive Scalar-FGLM *and* Adaptive Scalar-FGLM *algorithms is the parameter e allowing one to check elements at distance e from the uncovered staircase. Should this parameter be as big as the order of the table, the algorithm would behave as the* Scalar-FGLM *algorithm. Therefore, this parameter represents the trade-off between an exact computation and the output-sensitivity of the* Adaptive Scalar-FGLM *algorithm.*

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

The complexity of Algorithms 3, 4 and 5 depend on two main parameters: the number of table queries and the linear algebra part. Section 6 deals with the latter while we shall focus on the former in this section.

In the *black-box* model, it may be possible that computing *a single* element $u_{\mathbf{i}}$ of the table is very costly, we refer for instance to Section 5.2. Hence, it is important to minimize the number of queries.

Estimating this number is equivalent to counting the number of distinct elements in H_S where S can be any state of the variable in Algorithm 4. We denote by S the set 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_S where S is the final staircase. Restating Theorem 2, the necessary number of queries to $\mathbf{u}$ to build H_S is the cardinal of $2\,S = \{u\,v \mid (u, v) \in S^2\}$ the dilated set of S.

Obviously $\#(2\,S) \leq \# S(\# S - 1)/2 \leq (\# S)^2/2$ in the worst case; however in many applications we have $\#(2\,S) \leq c\,\# S$ for some constant c. (Ruzsa, 1994, Theorem 1.1) states that sets S satisfying this condition are exactly those 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 is included in a d-dimensional parallelotope with $C\,\# S$ points.

Proposition 14. *We give several estimations on* $\#(2\,S)$ *depending on the shape of the final staircase for* $d \to \infty$*.*

(a) *(Dimension* 1 *– the* BM *algorithm)* $n = 1$*,* $S_d = \{1, x, \ldots, x^{d-1}\}$ *then* $2\,S_d = S_{2d-1}$ $\#(2\,S_d) = 2\,d - 1$ *and*

$$\frac{\#(2\,S_d)}{\# S_d} = \frac{2\,d - 1}{d} \approx 2;$$

(b) *(Dimension 1 – worst case)* $n = 1$, $S_d = \{1, x^2, x^4, \ldots, x^{2^{d-1}}\}$ *then* $\#(2\,S_d) = \binom{d+1}{2} + 1$ *and*

$$\frac{\#(2\,S_d)}{\#S_d} = \frac{\binom{d+1}{2}+1}{d} \approx \frac{d}{2};$$

(c) *(Algorithm 3)* $S_d = \{t \in \mathcal{T} \mid \deg t < d\}$ *then* $2\,S_d = S_{2\,d-1}$ *and* $\#(2\,S_d) = \binom{n+2\,d-2}{n}$ *and*

$$\frac{\#(2\,S_d)}{\#S_d} = \frac{\binom{n+2\,d-2}{n}}{\binom{n+d-1}{n}} \approx \frac{(n+2\,d)^n}{(n+d)^n} \approx 2^n;$$

(d) *(Shape position) When* $G = \{x_1 - h_1(x_n), \ldots, x_{n-1} - h_{n-1}(x_n), h_n(x_n)\}$, *with* $\deg h_n = d$, *then* $S_d = \{1, x_n, \ldots, x_n^{d-1}\}$. *Again*

$$\frac{\#(2\,S_d)}{\#S_d} \approx 2;$$

(e) *(Dimension n – worst case)* $S_d = \bigcup_{i=1}^{n}\{1, x_i, x_i^2, \ldots, x_i^{d/n}\}$ *then*

$$\frac{\#(2\,S_d)}{\#S_d} \approx \frac{1}{2}\frac{n-1}{n}\#S_d.$$

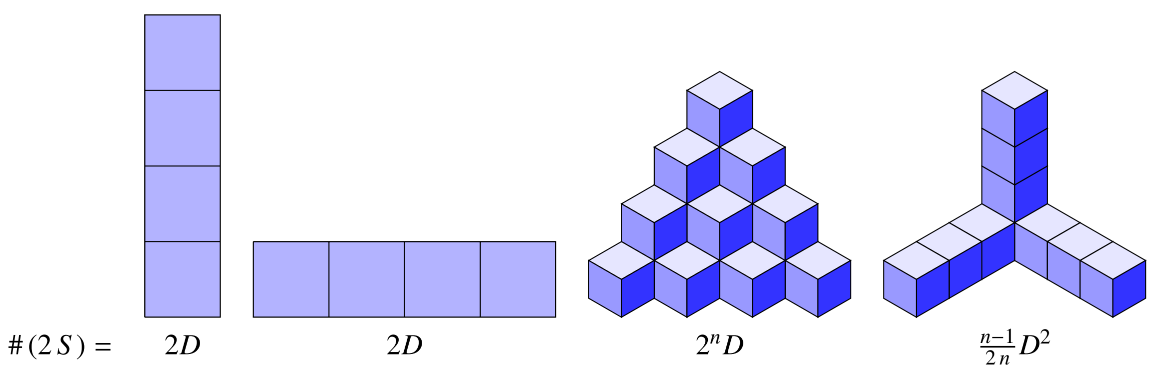

Figure 1: Behavior of $\#(2\,S)$ wrt. $D = \#S$ (the area in blue)

Proof.

a. Clearly $2\,S_d = S_{2\,d-1}$.

b. $S_{2\,d} = S_d \cup \{x^{2^i+2^j} \mid 1 \leq i < j \leq d-1\} \cup \{x^{2^d}\}$. Note that S_d is *not stable* under division in that case.

c. Noticing $2\,S_d = S_{2\,d-1}$ and $\#\,S_d = \binom{n+d-1}{n}$, we have

$$\frac{\#\,S_{2\,d-1}}{\#\,S_d} = 2^n - 2^{n-1}\binom{n+1}{2}\frac{1}{d-1} + O\left(\frac{1}{d^2}\right).$$

d. Same as item a.

e. We define $S'(n,d) = \bigcup_{i=1}^{n}\{x_i^j \mid j = 0,\ldots,d\}$ and it easy to show that $\#\,(2\,S'(n,d)) = n\,(n-1)\,d^2 + 2\,n\,d + 1$. Hence $S_d = S'(n, d/n)$ and

$$\frac{\#\,(2\,S_d)}{\#\,S_d} = \frac{\frac{n-1}{2n}\,d^2 + 2\,d + 1}{d+1} \approx \frac{1}{2}\,\frac{n-1}{n}\,d. \qquad \square$$

5.2. Application to the Sparse-FGLM *algorithm*

The Sparse-FGLM algorithm, Faugère and Mou (2011), is a natural application of the previous algorithm: for a 0-dimensional polynomial system we compute a first 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 $u_{\mathbf{i}} = \langle \mathbf{r}, T_1^{i_1} \cdots T_n^{i_n} \cdot \mathbf{1}\rangle$ where $\mathbf{r}$ is a random vector and $\mathbf{1} = [1, 0, \ldots]^{\mathrm{T}}$. The computation of one element of the table from the previous ones can be reduced to one matrix-vector multiplication.

Remark 15. *Assuming that we store the vectors* $\mathbf{V}_{\mathbf{i}} = T_1^{i_1} \cdots T_n^{i_n} \cdot \mathbf{1}$ *for the visited indices* $\mathbf{i}$*, any relation* $g = \sum_{s\in S} \alpha_s\, s \in G$ *computed by the algorithm can be easily checked: if* $\sum_{s\in S} \alpha_s\, \mathbf{V}_s = 0$ *then we have a proof that* $g \in I$*. 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 Algorithm 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 Faugère and Mou (2011), when the lexicographical basis is in shape position, the Gröbner basis can be computed very efficiently; in particular, the number of table queries is $2\,D$, in this situation we can also use the change of variables designed in Section 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 ith equation is of degree i and is invariant by the action of the nth Cyclic group; since there is a linear equation, the actual number of variables is $n-1$. We report in Table 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}.

Example	n	D	Nb Ranks	# Queries/D
Cyclic-5	5	70	76	7.4
Cyclic-6	6	156	167	9.4
Cyclic-7	7	924	953	21.7

Figure 2: Number of rank computations and table queries

Ideal of points

Given a set $P \subset \mathbb{K}^n$ of t distinct points, we define the ideal $I_P = \{f \in \mathbb{K}[x_1, \dots, x_n] \mid f(\mathbf{p}) = 0 \ \forall\, \mathbf{p} \in P\}$. We consider two such sets.

a. (Random) For any integer B, we generate exactly t points in $P_B \subset \mathbb{K}^n$ with coordinates randomly chosen in $\{0, \dots, B-1\}$. 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) $P_t = \{i\, e_j,\ 1 \leq i \leq n, 1 \leq j \leq t/n\}$.

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 Proposition 14. In Figure 3, 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 ℓ be an integer and $a \in \mathbb{F}_p$ such that $a^j \neq 1$ for $0 < j < p-1$. We work with polynomials in $R = \mathbb{F}_p[\mathbf{x}]/\langle x_1^{p-1} - 1, \dots, x_n^{p-1} - 1\rangle$. Then we define the generating polynomials $g_i(\mathbf{x}) = \prod_{j=0}^{\ell-1}(x_i - a^j)$. When we send a message $\mathcal{M}$ we split this message into n blocks $\mathcal{M}^{(k)} = (c_1^{(k)}, c_2^{(k)}, \dots)$ where $c_i \in \mathbb{F}_p$ and we generate n multivariate polynomials $U_k(\mathbf{x}) = c_1^{(k)} + c_2^{(k)} x_1 + c_3^{(k)} x_2 + \cdots$. The transmitter sends the encoded message $M(\mathbf{x}) = \sum_{k=1}^{n} g_k(\mathbf{x}) U_k(\mathbf{x})$. The receiver interprets the received word as a multivariate polynomial $N(\mathbf{x}) = M(\mathbf{x}) + e(\mathbf{x})$ where $e(\mathbf{x}) \in R$ is the error polynomial. If the length of $e(\mathbf{x})$ is less than $t = \frac{\ell}{2}$ the goal is to recover it. To this end, we build the table $u_{i_1,\dots,i_n} := N(a^{i_1}, \dots, a^{i_n}) \equiv e(a^{i_1}, \dots, a^{i_n})$ in R for $0 \leq i_j < t$ and we apply Algorithm 4 to obtain a LEX Gröbner basis G. It is easy to recover all the solutions in the finite field $\mathbb{F}_q$; next, by computing the discrete logarithm wrt. a of all the components we recover the position of the nonzero monomials in $e(\mathbf{x})$. Lastly, we solve a linear system to find the coefficients of $e(\mathbf{x})$.

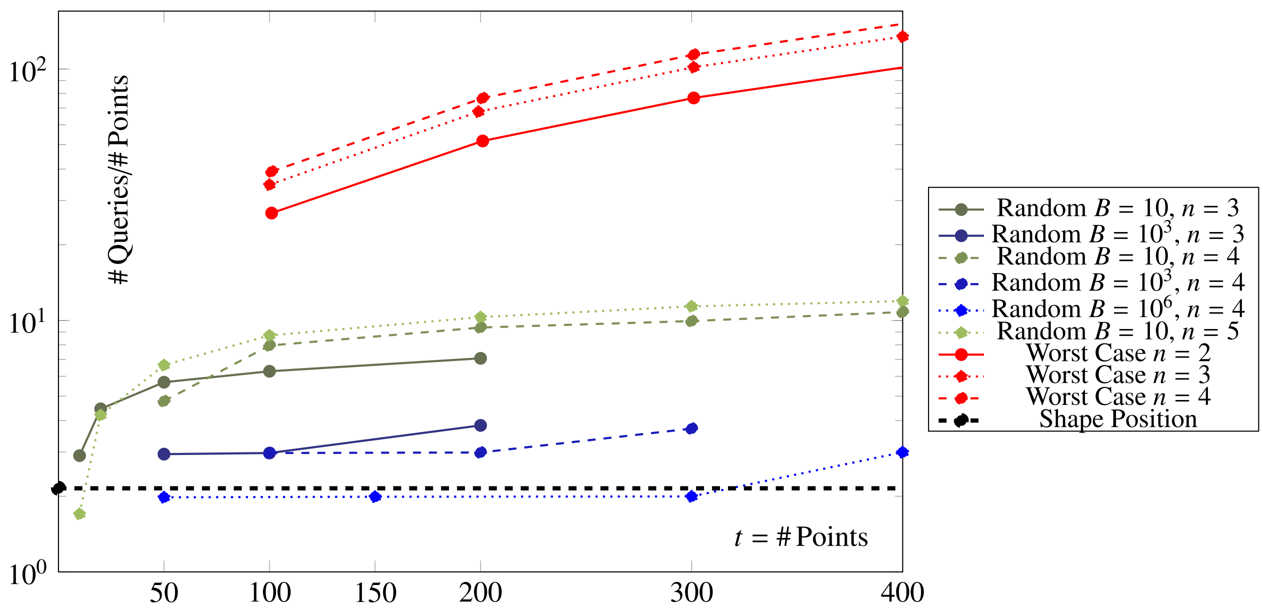

Figure 3: Number of table queries divided by number of points.

In the experiments of Table 4, we consider two cases: (random case) we randomly generate the support and the coefficients of the error polynomial $e(\mathbf{x})$; (worst case) we take $e(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=0}^{t/n} c_{i,j}\, x_i^j$.

We can see that whenever the errors are picked at random, the number of table queries grows linearly in the number of errors, i.e. the size of the staircase of the Gröbner basis. This illustrates that generically, in the error correcting codes application, the decoding complexity should be linear in the number of errors. In accordance with Proposition 14, we observe a quadratic behaviour for the worst case scenario.

6. Multilevel Block Hankel Arithmetic

This section is devoted to the complexity of Algorithms 3 and 4. Multi-Hankel matrices are structured but exploiting this structure might not be so easy at a first glance. When the chosen monomial ordering $\prec$ is a LEX order, then the matrices are multilevel block Hankel, see Fasino and Tilli (2000) and Serra-Capizzano (2002) for results on multilevel block Hankel or multilevel block Toeplitz matrices.

We recall that a Gröbner basis of a 0-dimensional ideal I for LEX order on $x_1, \ldots, x_n$ with $x_1 \prec \cdots \prec x_n$ is of special form and interest. The least polynomial

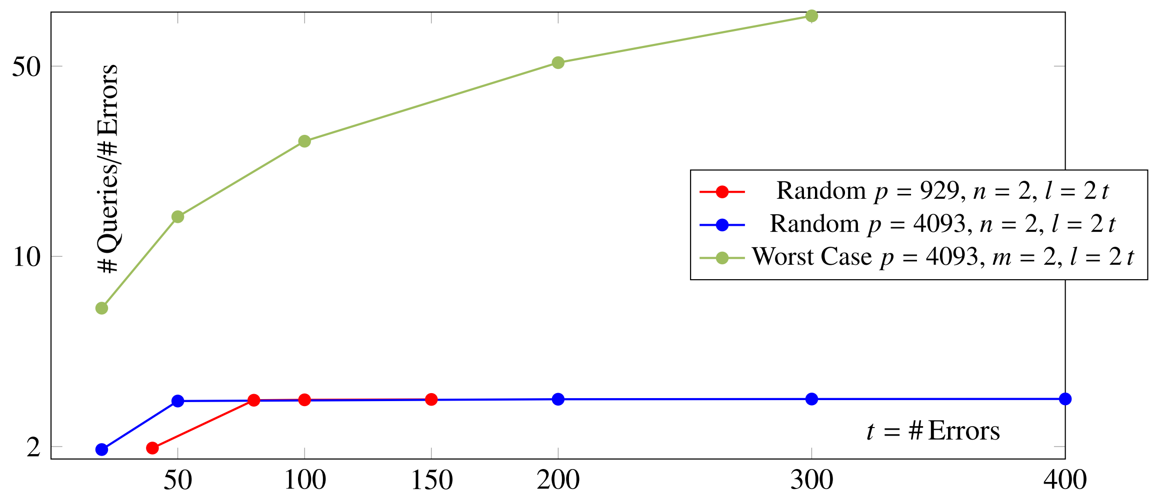

Figure 4: Number of table queries divided by number of points.

is univariate in x_1 and shall be called $P_{1,1}$. Then, for each $k > 1$, there are polynomials $P_{k,1}, \ldots, P_{k,m_k} \in \mathbb{K}[x_1, \ldots, x_k]$ with $\deg_{x_k} P_{k,\ell} \leq d_k$, for all ℓ, $1 \leq \ell \leq m_k$. Furthermore, $\mathrm{LT}(P_{k,m_k})$ is a pure power of x_k. We also remind the reader that

$$\forall\, k,\ 1 \leq k \leq n,\ I \cap \mathbb{K}[x_1, \ldots, x_k] = \langle P_{1,1}, \ldots, P_{k,1}, \ldots, P_{k,m_k} \rangle.$$

Definition 7. *A* multilevel block Hankel matrix of depth 0 *is a scalar while a* multilevel block Hankel matrix of depth 1 *is a Hankel matrix.*

For any $n \in \mathbb{N}$, a multilevel block Hankel matrix of depth $n + 1$ *is a block Hankel matrix whose blocks are multilevel block Hankel matrices of depth n.*

In the remaining part of the section, a multilevel block Hankel matrix of depth n shall be called n-multiblock Hankel.

Following Algorithms 3 and 4, a multi-Hankel matrix is built from an increasing set of monomials. For the LEX ordering with $x_1 \prec \cdots \prec x_n$, the set of monomials is at first $S^1 = \{1, x_1, \ldots, x_1^{d_1-1}\}$ and the matrix H_S is indeed Hankel, as in the BM algorithm.

Because of the relation induced by $P_{1,1}$ no more pure powers of x_1 are needed and we shall introduce x_2. The set of monomials of the useful staircase, see Definition 5, is now $S^2 = S^1_0 \cup x_2\, S^1_1 \cup \cdots \cup x_2^{d_2-1}\, S^1_{d_2-1}$ with $S^1_1, \ldots, S^1_{d_2-1} \subseteq S^1_0 = S^1$. For any i, j, we let $H_{x_2^i S_i, x_2^j S_j}$ be Hankel rectangular. This construction yields the following matrix

$$H_{S^2} = \begin{pmatrix} H_{S_0^1} & H_{S_0^1, x_2 S_1^1} & \cdots & H_{S_0^1, x_2^{d_2-1} S_{d_2-1}^1} \\ H_{x_2 S_1^1, S_0^1} & H_{x_2 S_1^1} & \cdots & H_{x_2 S_1^1, x_2^{d_2-1} S_{d_2-1}^1} \\ \vdots & \vdots & \ddots & \vdots \\ H_{x_2^{d_2-1} S_{d_2-1}^1, S_0^1} & H_{x_2^{d_2-1} S_{d_2-1}^1, x_2 S_1^1} & \cdots & H_{x_2^{d_2-1} S_{d_2-1}^1} \end{pmatrix}$$

We can extend H_{S^2} so that each block is square, by replacing each rectangular block $H_{x_2^i S_i^1, x_2^j S_j^1}$ by $H_{x_2^i S^1, x_2^j S^1}$. The extended matrix

$$\begin{pmatrix} H_{S^1} & H_{S^1, x_2 S^1} & \cdots & H_{S^1, x_2^{d_2-1} S^1} \\ H_{x_2 S^1, S^1} & H_{x_2 S^1} & \cdots & H_{x_2 S^1, x_2^{d_2-1} S^1} \\ \vdots & \vdots & \ddots & \vdots \\ H_{x_2^{d_2-1} S^1, S^1} & H_{x_2^{d_2-1} S^1, x_2 S^1} & \cdots & H_{x_2^{d_2-1} S^1} \end{pmatrix}$$

is 2-*multiblock Hankel* and has the same rank has the original H_{S^2} by construction and the definition of a useful staircase.

More generaly, assuming, when reaching variable x_k, $k \in \mathbb{N}$, the constructed matrix H_{S^k} can be embedded in a k-multiblock Hankel matrix. Then for variable x_{k+1}, we shall consider the matrix $H_{S^{k+1}}$ that is block Hankel with blocks $H_{x_{k+1}^i S_i^k, x_{k+1}^j S_j^k}$, $0 \leq i, j \leq d_{k+1} - 1$ such that $S_1^k, \ldots, S_{d_{k+1}-1}^k \subseteq S_0^k = S^k$. That is, they will have the same shape as H_{S^k} and thus can be embedded in a k-multiblock Hankel. When replacing each block $H_{x_{k+1}^i S_i^k, x_{k+1}^j S_j^k}$ by $H_{x_{k+1}^i S^k, x_{k+1}^j S^k}$, the matrix is $(k+1)$-*multiblock Hankel*.

Example 12. *We consider the following table* $\mathbf{u} = (u_{i,j,k})_{(i,j,k)\in\mathbb{N}^3}$ *defined as follows*

$$\forall\, (i, j, k) \in \mathbb{N}^3,\ u_{i,j,k} = 2^i + (1 + j)\,(1 + k).$$

The Gröbner basis of the ideal of relations I returned by Algorithm 3 for LEX *with* $x \prec y \prec z$ *is*

$$\begin{aligned} I &= \langle x^2 - 3\,x + 2, x\,y - y - x + 1, y^2 - 2\,y + 1, x\,z - z - x + 1, z^2 - 2\,z + 1\rangle \\ &= \langle (x-1)\,(x-2), (x-1)\,(y-1), (y-1)^2, (x-1)\,(z-1), (z-1)^2\rangle. \end{aligned}$$

The set of monomials S^1 *is thus* $\{1, x\}$ *and matrix* H_{S^1} *is* $\begin{pmatrix} 2 & 3 \\ 3 & 5 \end{pmatrix}$, *which is Hankel.*

The set of monomials S^2 is $\{1, x, y\}$ and extending it yields $S_e^2 = S^1 \cup yS^1 = \{1, x, y, xy\}$. The matrix H_{S^2} and 2-multiblock Hankel matrix $H_{S_e^2}$ are as follows

$$H_{S^2} = \left(\begin{array}{cc|c} 2 & 3 & 3 \\ 3 & 5 & 4 \\ \hline 3 & 4 & 4 \end{array}\right), \quad H_{S^2} = \left(\begin{array}{cc|cc} 2 & 3 & 3 & 4 \\ 3 & 5 & 4 & 6 \\ \hline 3 & 4 & 4 & 5 \\ 4 & 6 & 5 & 7 \end{array}\right).$$

Finally, the set of monomials S^3 is $\{1, x, y, z, yz\}$ while the extended set is $S_e^3 = S_e^2 \cup zS_e^2 = \{1, x, y, xy, z, xz, yz, xyz\}$. Then the matrices are

$$H_{S^3} = \left(\begin{array}{cc|c||c|c} 2 & 3 & 3 & 3 & 5 \\ 3 & 5 & 4 & 4 & 6 \\ \hline 3 & 4 & 4 & 5 & 7 \\ \hline\hline 3 & 4 & 5 & 4 & 7 \\ \hline 5 & 6 & 7 & 7 & 10 \end{array}\right), \quad H_{S_e^3} = \left(\begin{array}{cc|cc||cc|cc} 2 & 3 & 3 & 4 & 3 & 4 & 5 & 6 \\ 3 & 5 & 4 & 6 & 4 & 6 & 6 & 7 \\ \hline 3 & 4 & 4 & 5 & 5 & 6 & 7 & 8 \\ 4 & 6 & 5 & 7 & 6 & 7 & 8 & 10 \\ \hline\hline 3 & 4 & 5 & 6 & 4 & 5 & 7 & 8 \\ 4 & 6 & 6 & 7 & 5 & 7 & 8 & 10 \\ \hline 5 & 6 & 7 & 8 & 7 & 8 & 10 & 11 \\ 6 & 7 & 8 & 10 & 8 & 10 & 11 & 13 \end{array}\right).$$

A *displacement operator* φ for a matrix H is a linear operator acting on matrices s.t. $\varphi(H)$ has small rank.

It is well known that the best displacement operator for Hankel matrices comes from a "shift of the coefficients along the anti-diagonals". That is, denoting

$$Z = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ 1 & \ddots & & \vdots \\ & \ddots & \ddots & \vdots \\ & & 1 & 0 \end{pmatrix},$$

then for H Hankel, $\varphi(H) = H - ZHZ$ has rank at most 2.

Solving linear systems with a Hankel matrix can be done in $O(\mathsf{M}(d) \log d)$ operations in the base field, where $\mathsf{M}(d)$ is the complexity for multiplying two polynomials of degree at most $d - 1$.

Hankel-like matrices, which are small sums of Hankel matrices, or equivalently which are matrices sent on low-rank matrices by the aforementioned displacement operator φ, can also be solved fast. For H of size d, and $\alpha = \operatorname{rank} \varphi(H)$, the linear system can be solved in $O(\alpha^{\omega-1} \mathsf{M}(d) \log d)$ operations in the base. We refer the reader to Bostan et al. (2007).

Block Hankel matrices are also of interest by themselves. Deflating the operator φ by replacing 1's by identity matrices of size the blocks' size gives a new displacement operator. Sending block Hankel matrices to matrices of rank twice the size of the blocks. But the 2-multiblock Hankel matrix has Hankel blocks, therefore, it seems natural to apply φ to each blocks. It is quite easy to show that the rank of the resulting matrix is at most $2\,\min(d_1, d_2)$ where d_2 is the number of blocks and d_1 the size of the blocks.

For a $(n+1)$-multiblock Hankel matrix, this can be generalized: we deflate the displacement operator for n-multiblock Hankel matrices, apply it and then apply the displacement operator for n-multiblock Hankel matrices to each block.

Consequently, for n-multiblock Hankel matrices and embedding blocks of sizes $d_1, \ldots, d_n$, one can find a displacement operator, different from the Hankel matrices displacement operator, s.t. the displacement rank is at most $2\prod_{i=1}^{n} d_i/\max_{i=1}^{n} d_i$.

We leave it open if one could use this structure to improve the complexity estimate of solving such a system.

6.1. Complexity comparisons

In this section, we estimate the complexity of Algorithm 3 and compare it with the BMS algorithm. This complexity shall depend on two parameters: the dimension n of the table and the size d of the staircase of the (truncated) Gröbner basis output by the algorithm.

In Sakata (2009), the complexity of the BMS algorithm is given in terms of another parameter: μ, the cardinality of the Gröbner basis. The complexity of the BMS algorithm is then $O(\mu\, d^2)$. While n, d do not depend on $\prec$, we recall that μ does. Sakata uses the approximation $\mu \in O(d)$ to give an upper bound on the BMS algorithm, that is $O(d^3)$.

To this day, (Faugère et al., 1993, Cor. 2.1) gives the only known upper bound on μ in terms of n and d: $\mu \leq n\,d$. This allows us to estimate the complexity of the BMS algorithm as $O(n\,d^3)$ operations in $\mathbb{K}$.

Our multiblock Hankel point of view seems to overestimate the complexity of Algorithm 3, in particular because the n-multiblock Hankel matrix is bigger than the actual computed matrix, see Example 12.

We give a situation where we can estimate the complexity of Algorithm 3.

Proposition 16. *Let μ be the number of polynomials in the output Gröbner basis. Then the number of operations in the base field to compute the Gröbner basis is no more than*

$$O\left(\mu\, d_2^{\omega-1}\cdots d_n^{\omega-1}\mathsf{M}(d_1\cdots d_n)\,\log(d_1\cdots d_n)\right).$$

In particular, in the shape position *case, $d_2 = \cdots = d_n = 1$, $\mu = n$ and the complexity comes down to $O(n\,\mathsf{M}(d)\,\log d)$.*

Proof. Let $\delta = d_2\cdots d_n$. For the LEX order, the extended multiblock Hankel matrix H is in fact quasi-Hankel. As H is a block matrix with δ blocks on each row and column, with blocks of size d_1, then $\varphi(H)$ contains δ column vectors of size $\delta\, d_1$ and $\delta\,(d_1 - 1)$ column vectors of size δ deflated into vectors of size $\delta\, d_1$. Thus, at most $2\,\delta$ columns of $\varphi(H)$ are independent and the displacement rank of H is $2\,\delta$. By a result of Bostan et al. (2007), one can solve a linear system with H as the matrix in $O\left(\delta^{\omega-1}\,\mathsf{M}(d_1\,\delta)\,\log(d_1\,\delta)\right)$. One such system must be solved for each polynomial in the Gröbner basis, hence a factor μ in the complexity estimate. The *shape position* case is then straightforward. □

Corollary 17. *Assume $d_2, \ldots, d_n$ are bounded and d_1 is not. Then, μ is also necessarily bounded and the number of operations in the base field to compute the Gröbner basis is no more than*

$$O\left(\mu\, d_2^{\omega-1} \cdots d_n^{\omega-1} \mathsf{M}(d_1 \cdots d_n)\,\log(d_1 \cdots d_n)\right) = O(\mathsf{M}(d)\,\log d).$$

Proof. Let $x_1^{i_1} \cdots x_n^{i_n}$ be the leading monomial of a polynomial of the Gröbner basis. Since $i_2, \ldots, i_n$ are respectively bounded by $d_2, \ldots, d_n$, we have a bounded number of choices for $i_2, \ldots, i_n$, independent from d_1. Thanks to the divisibility property, only one i_1 can then be choosen. Therefore, μ is bounded. □

Let us notice that, whenever n is fixed, the complexity of the BMS algorithm is cubic in d while Algorithm 3 is quasi-linear in d, under the hypotheses given in Corollary 17.

It is classical that the BM algorithm is equivalent to solving a Hankel linear system, the special case $n = 1$ of Algorithm 3, therefore both have the same complexity. However, we are not able to say if either of our algorithms can be seen as a matrix version of the BMS algorithm. Should Algorithm 3 or 4 be equivalent to the BMS algorithm, we could improve the complexity of the BMS algorithm. On the one hand, finding loop invariants in these algorithms could also be the key to reach this goal and make their complexities sharper. On the other hand, it could also help us finding optimal termination criteria, in order to reduce the number of table queries.

7. Computing the Generating Series

Given a sequence $\mathbf{u} = (u_{\mathbf{i}})_{\mathbf{i}\in\mathbb{N}^n}$ defined over $\mathbb{K}$, its generating series $S \in \mathbb{K}[[\mathbf{x}]]$ is defined as

$$S = \sum_{\mathbf{i}\in\mathbb{N}^n} u_{\mathbf{i}}\, \mathbf{x}^{\mathbf{i}}.$$

Whenever $n = 1$, it is classical that $S \in \mathbb{K}(x)$ if and only if $\mathbf{u}$ is linear recursive. For multivariate sequences, only the if part of the statement remains true. Indeed, as shown in Section 2, Example 2, $\mathbf{b} = \left(\binom{i}{j}\right)_{(i,j)\in\mathbb{N}^2}$ is not linear recursive, yet P-recursive, and its generating series is

$$S = \sum_{(i,j)\in\mathbb{N}^2} \binom{i}{j} x^i\, y^j = \frac{1}{1 - x - x\,y} \in \mathbb{K}(x, y).$$

On the other hand, the P-recursive condition cannot be sufficient to have a generating series as a rational fraction. The unidimensional sequence $\mathbf{u} = (1/i!)_{i\in\mathbb{N}}$ is P-recursive and its generating sequence, the exponential, is not in $\mathbb{K}(x)$.

If $\mathbf{u}$ is linear recursive, then S can be computed from the ideal of relations and finitely many terms of $\mathbf{u}$. As the ideal of relations of $\mathbf{u}$ is zero-dimensional, let us all recall that for all $k \in \{1, \ldots, n\}$, $I \cap \mathbb{K}[x_k] = (P_k(x_k))$ with P_k nonzero and monic. In other words, for any x_k, there exist univariate polynomials in x_k in I and P_k is their monic greatest common divisor.

For a given polynomial $P \in \mathbb{K}[x]$ of degree d, we denote $Q = x^d\, P(1/x)$ its *reciprocal polynomial*, i.e. if $P = \sum_{i=0}^{d} p_i\, x^i$, then $Q = \sum_{i=0}^{d} p_i\, x^{d-i}$. Notice, that such a Q cannot be a multiple of x.

Proposition 18. *Let* $\mathbf{u} = (u_{\mathbf{i}})_{\mathbf{i}\in\mathbb{N}^n}$ *a linear recursive sequence and let* I *be its ideal of relations. For* k, $1 \le k \le n$, *let* $P_k(x_k)$ *be the monic polynomial spanning* $I \cap \mathbb{K}[x_k]$, d_k *denote its degree and* $Q_k(x_k)$ *be* P_k*'s reciprocal. Then,*

$$S = \frac{\left(Q_1(x_1)\cdots Q_n(x_n) \sum_{\mathbf{i}=(0,\ldots,0)}^{(d_1-1,\ldots,d_n-1)} u_{\mathbf{i}}\, \mathbf{x}^{\mathbf{i}}\right) \bmod (x_1^{d_1}, \ldots, x_n^{d_n})}{Q_1(x_1)\cdots Q_n(x_n)}.$$

Conversely, if S *is a rational fraction of* $\mathbb{K}(x_1, \ldots, x_n)$ *whose denominator can be factored as* $Q_1(x_1)\cdots Q_n(x_n)$*, then* $\mathbf{u}$ *is linear recursive.*

Proof. We shall prove this by induction on n, the number of variables.

Assuming $n = 1$, then by definition $I = (P(x))$ with P nonzero, monic and of degree d. Let $P = p_0 + \cdots + p_d\, x^d$, with $p_d = 1$, then $Q(x) = p_0\, x^d + \cdots + 1$ and

$$\begin{aligned}
Q(x)\, S &= \sum_{k=0}^{d} p_k \sum_{i=0}^{\infty} u_i\, x^{i+d-k} = \sum_{k=0}^{d} \sum_{i=-k}^{\infty} p_k\, u_{i+k}\, x^{i+d}\\
&= (p_d\, u_0) + (p_{d-1}\, u_0 + p_d\, u_1)\, x + \cdots + (p_1\, u_0 + \cdots + p_d\, u_{d-1})\, x^{d-1}\\
&\quad + \sum_{i=0}^{\infty} \left(\sum_{k=0}^{d} p_k\, u_{i+k}\right) x^{i+d}\\
&= (p_d\, u_0) + (p_{d-1}\, u_0 + p_d\, u_1)\, x + \cdots + (p_1\, u_0 + \cdots + p_d\, u_{d-1})\, x^{d-1}\\
&= Q(x) \sum_{i=0}^{d-1} u_i\, x^i \bmod x^d.
\end{aligned}$$

Conversely, if $S = (a_0 + \cdots + a_{d-1}\, x^{d-1})/Q(x)$ with $Q = q_0 + \cdots + q_d\, x^d$, $q_0 = 1$ and $q_1, \ldots, q_d \in \mathbb{K}$. Then it is clear that S satisfies

$$S = a_0 + \cdots + a_{d-1}\, x^{d-1} - (q_1\, x + \cdots + q_d\, x^d)\, S.$$

On the left-hand-side of the equation, for $i \geq d$, the coefficient of x^i is merely u_i, while on the right-hand-side it is $-q_1\, u_{i-1} - \cdots - q_d\, u_{i-d}$, hence $\mathbf{u}$ is linear recursive.

Let $n \in \mathbb{N}^*$, let us assume the statement holds for all k with $1 \leq k \leq n$ and let us prove the statement still holds for $n+1$. Let $\mathbf{u} = (u_{\mathbf{i},j})_{(\mathbf{i},j)\in\mathbb{N}^n\times\mathbb{N}}$ be a linear recursive sequence. Let I be the ideal of relations of $\mathbf{u}$, for all k, $1 \leq k \leq n+1$, let $(P_k(x_k)) = I \cap \mathbb{K}[x_k]$ and let Q_k be the reciprocal of P_k.

For $\mathbf{i} \in \mathbb{N}^n$, each sequence $\mathbf{v}^{(\mathbf{i})} = (v_j^{(\mathbf{i})})_{j\in\mathbb{N}} = (u_{\mathbf{i},j})_{j\in\mathbb{N}}$ is one-dimensional linear recursive satisfying the recurrence relation

$$v_{j+d_{n+1}}^{(\mathbf{i})} + p_{n+1,d_{n+1}-1}\, v_{j+d_{n+1}-1}^{(\mathbf{i})} + \cdots + p_{0,d_{n+1}-1}\, v_j^{(\mathbf{i})} = 0,$$

where $P_{n+1}(x_{n+1}) = x_{n+1}^{d_{n+1}} + p_{n+1,d_{n+1}-1}\, x^{d_{n+1}-1} + \cdots + p_{0,d_{n+1}-1}$. Thus, one has

$$\begin{aligned} S &= \sum_{\mathbf{i}\in\mathbb{N}^n} \sum_{j=0}^{\infty} u_{\mathbf{i},j}\, x_{n+1}^j\, \mathbf{x}^{\mathbf{i}} = \sum_{\mathbf{i}\in\mathbb{N}^n} \sum_{j=0}^{\infty} v_j^{(\mathbf{i})}\, x_{n+1}^j\, \mathbf{x}^{\mathbf{i}} \\ &= \sum_{\mathbf{i}\in\mathbb{N}^n} \frac{Q_{n+1}(x_{n+1}) \sum_{j=0}^{d_{n+1}-1} u_{\mathbf{i},j}\, x_{n+1}^j \bmod x_{n+1}^{d_{n+1}}}{Q_{n+1}(x_{n+1})}\, \mathbf{x}^{\mathbf{i}} \\ &= \sum_{\mathbf{i}\in\mathbb{N}^n} S_{\mathbf{i}}(x_{n+1})\, \mathbf{x}^{\mathbf{i}}. \end{aligned}$$

Clearly, $(S_{\mathbf{i}}(x_{n+1}))_{\mathbf{i}\in\mathbb{N}^n}$ is a n-dimensional linear recursive sequence over $\mathbb{K}(x_{n+1})$ satisfying the relations associated to $P_1(x_1), \ldots, P_n(x_n)$. Hence

$$S = \frac{\left(Q_1(x_1)\cdots Q_n(x_n)\, \sum_{\mathbf{i}=(0,\ldots,0)}^{(d_1-1,\ldots,d_n-1)} S_{\mathbf{i}}(x_{n+1})\right) \bmod (x_1^{d_1}, \ldots, x_n^{d_n})}{Q_1(x_1)\cdots Q_n(x_n)}$$

$$S = \frac{\left(\left(\prod_{k=1}^{n+1} Q_k(x_k)\right) \sum_{(\mathbf{i},j)=(0,\ldots,0)}^{(d_1-1,\ldots,d_{n+1}-1)} u_{\mathbf{i},j}\, \mathbf{x}^{\mathbf{i}}\, x_{n+1}^j\right) \bmod (x_1^{d_1}, \ldots, x_{n+1}^{d_{n+1}})}{Q_1(x_1)\cdots Q_{n+1}(x_{n+1})}$$

Conversely, let us assume $S = A(x_1, \ldots, x_{n+1})/(Q_1(x_1)\cdots Q_{n+1}(x_{n+1}))$. Then, S can be seen as a rational fraction in $x_1, \ldots, x_n$ over $\mathbb{K}(x_{n+1})$. By the induction hypothesis, this means it is the generating series of some linear recursive sequence $(S_{\mathbf{i}}(x_{n+1}))_{\mathbf{i}\in\mathbb{N}^n} = (B_{\mathbf{i}}(x_{n+1})/Q_{n+1}(x_{n+1}))_{\mathbf{i}\in\mathbb{N}^n}$ satisfying the relations associated to $P_1(x_1), \ldots, P_n(x_n)$. Hence, each $S_{\mathbf{i}}$ is a linear recursive sequence satisfying the relation associated to $P_{n+1}(x_{n+1})$. Finally, $\mathbf{u}$ satisfies also those $n+1$ relations which makes its ideal of relations zero-dimensional. By Definition 4, $\mathbf{u}$ is linear recursive. □

Remark 19. *The generating series of a multidimensional sequence satisfying* one *linear recurrence relation with constant coefficients is given in Bousquet-Mélou and Petkovšek (2000). The authors show in (Bousquet-Mélou and Petkovšek, 2000, Section 4.2) that whenever the initial conditions have rational generating functions, then the generating series is rational. This could give another proof of Proposition 18 by recurrence on the number of variables.*

Their result is illustrated with the binomial sequence $\mathbf{b} = \left(\binom{i}{j}\right)_{(i,j)\in\mathbb{N}^2}$ *whose generating series* $\frac{1}{1-x-xy}$ *is rational. The sequence satisfies the relation* $b_{i+1,j+1} - b_{i,j+1} - b_{i,j}$ *for all* $(i,j) \in \mathbb{N}^2$ *and its set of initial terms* $\left\{\binom{i}{0}, \binom{0}{j} \middle| i \in \mathbb{N}, j \in \mathbb{N}^*\right\}$ *has the rational generating series*

$$\sum_{i\in\mathbb{N}} \binom{i}{0} x^i + \sum_{j\in\mathbb{N}^*} \binom{0}{j} y^j = \sum_{i\in\mathbb{N}} x^i = \frac{1}{1-x}.$$

7.1. Algorithms for computing the generating series

Based on Proposition 18, this section is devoted to the design of several algorithms for computing the generating series. The first one is a deterministic algorithm using Algorithms 3 and 4. The second algorithm uses our probabilistic essential reduction to one call to the BM algorithm introduced in Section 3.2 while the last one is another probabilistic algorithm using n calls to the BM algorithm. These algorithms differ on the method to obtain all the n univariate polynomials $P_1 \in \mathbb{K}[x_1], \dots, P_n \in \mathbb{K}[x_n]$, needed to compute the generating series, see Proposition 18.

If needed, their common last step is expanding the numerator. For each k, $1 \leq k \leq n$, we need to multiply a univariate polynomial of degree less than d_k by $d_1 \cdots d_n / d_k$ univariate polynomials of degree less than d_k. Since $d_1, \dots, d_n \leq d$, this is can be done in $O(n\, d^{n-1} \mathsf{M}(d))$ operations in the base field.

Let us now describe Algorithm 6 for computing the generating series. Calling the BMS algorithm, Algorithm 3 or Algorithm 4 on a n-dimensional table with a LEX ordering with $x_1 \prec \cdots \prec x_n$ yields a Gröbner basis with P_1 but not $P_2, \dots, P_n$ which are also needed. For each k, $2 \leq k \leq n$, we apply a change of ordering on the output ideal to obtain a LEX Gröbner basis with $x_k \prec x_1 \prec \cdots \prec x_{k-1} \prec x_{k+1} \prec \cdots \prec x_n$ yielding $P_k \in \mathbb{K}[x_k]$.

Following Algorithm 7 uses Algorithm 1 and resultants computations to determine the n univariate polynomials. It works if the output of Algorithm 1 is in shape position.

Proposition 20. *Given a* $(n+1)$*-dimensional linear recursive table over* $\mathbb{K}$ *whose ideal of relations is in shape position, Algorithm 7 computes its* $n+1$ *univariate polynomials in* $O(n\, d\, \mathsf{M}(d) \log d)$ *operations in* $\mathbb{K}$.

Algorithm 6: Deterministic computation of the generating series.

Input: A table $\mathbf{u} = (u_{\mathbf{i}})_{\mathbf{i}\in\mathbb{N}^n}$.
Output: The n univariate polynomials needed for the generating series.
Compute $G_1 := \{P_1, P_{2,1}, \dots, P_{n,m_n}\}$ as returned by Algorithm 3 for LEX $x_1 \prec \cdots \prec x_n$.
For k **from** 2 **to** n **do**
  Compute $G_k := \{P_k\} \cup \{P_{i,j}^{(k)} | 1 \leq i \leq n, i \neq k, 1 \leq j \leq m_j^{(k)}\}$ as returned by the FGLM algorithm called on G_1 with LEX $x_k \prec x_1 \prec \cdots \prec x_{k-1} \prec x_{k+1} \prec \cdots \prec x_n$.
Return $P_1, \dots, P_n$.

Algorithm 7: Probabilistic computation of the generating series.

Input: A table $\mathbf{u} = (u_{\mathbf{i}})_{\mathbf{i}\in\mathbb{N}^n}$.
Output: The n univariate polynomials needed for the generating series.
Compute $f_0 \in \mathbb{K}[t], f_1 \in \mathbb{K}[x_1, t], \dots, f_n \in \mathbb{K}[x_n, t]$ as returned by Algorithm 1.
For k **from** 1 **to** n **do**
  Compute P_k the resultant of f_0 and f_k in t.
Return $P_1, \dots, P_n$.

Proof. Calling Algorithm 1 counts for $O(n\,\mathsf{M}(d)\log d)$ operations in $\mathbb{K}$. Then, each resultant computation is the resultant of two polynomials of degree at most d, one of them being bivariate and of degree 1 is the second variable. By evaluating the second variable and interpoling the results, each resultant can be computed in $O(d\,\mathsf{M}(d)\log d)$ operations, hence a global complexity in $O(n\,d\,\mathsf{M}(d)\log d)$ operations in $\mathbb{K}$. □

For all $k \in \{1, \dots, n\}$ and $N_1, \dots, N_{k-1}, N_{k+1}, \dots, N_n \in \mathbb{N}$, the sequence $\mathbf{u}^{(k)} = (u_i^{(k)})_{i\in\mathbb{N}} = (u_{N_1,\dots,N_{k-1},i,N_{k+1},\dots,N_n})_{i\in\mathbb{N}}$ is linear recursive of dimension 1 satisfying the relation associated to P_k. Therefore, running Berlekamp – Massey on this table yields a factor of P_k. Assuming P_k has degree d, with good probability however, table

$$\left(\sum_{\ell=1}^{d} \alpha_\ell\, u_{N_{\ell,1},\dots,N_{\ell,k-1},i,N_{\ell,k+1},\dots,N_{\ell,n}}\right)_{i\in\mathbb{N}}$$

should not satisfy any relation associated to a strict factor of P_k and running Berlekamp – Massey on it should yield P_k.

Example 13. *The sequence* $\mathbf{u} = ((-1)^{ij})_{(i,j)\in\mathbb{N}^2}$ *is linear recursive of order* 4 *whose ideal of relations is* $\langle x^2 - 1, y^2 - 1\rangle$.

For any even N, the sequence $(u_{i,N})_{i\in\mathbb{N}} = (1)_{i\in\mathbb{N}}$ satisfies the relation associated to $x-1$, while for any odd N, the sequence $(u_{i,N})_{i\in\mathbb{N}} = ((-1)^i)_{i\in\mathbb{N}}$ satisfies the relation associated to $x+1$. Therefore x^2-1 cannot be found by a single run of Berlekamp – Massey on the first coordinate.

Let $N_1, N_2 \in \mathbb{N}, \alpha_1, \alpha_2 \in \mathbb{K}$,

$$\mathbf{v} = (\alpha_1\, u_{N_1,i} + \alpha_2\, u_{i,N_2})_{i\in\mathbb{N}} = \left(\alpha_1\,(-1)^{N_1\,i} + \alpha_2\,(-1)^{N_2\,i}\right)_{i\in\mathbb{N}}.$$

For random N_1, N_2, with probability $1/2$, N_1 and N_2 are not both even nor odd. Assuming wlog. that N_1 is even and N_2 is odd, the sequence is $\left(\alpha_1 + \alpha_2\,(-1)^i\right)_{i\in\mathbb{N}}$ which satisfies at best x^2-1 as long as $\alpha_1 \neq 0, \alpha_2 \neq 0$.

Algorithm 8: Fast probabilistic computation of the generating series.

Input: A table $\mathbf{u} = (u_{\mathbf{i}})_{\mathbf{i}\in\mathbb{N}^n}$.
Output: The n univariate polynomials needed for the generating series.
For k **from** 1 **to** n **do**
  For ℓ **from** 1 **to** d **do**
    Pick at random $\alpha_\ell \in \mathbb{K}$.
    Pick at random $N_{\ell,1}, \dots, N_{\ell,k-1}, N_{\ell,k+1}, \dots, N_{\ell,n}$ in $\{0, \dots, d-1\}$.
  Compute P_k as returned by the BM algorithm on table $\left(\sum_{\ell=1}^{d} \alpha_\ell\, u_{N_{\ell,1},\dots,N_{\ell,k-1},i,N_{\ell,k+1},\dots,N_{\ell,n}}\right)_{i\in\mathbb{N}}$
Return $P_1, \dots, P_n$.

Proposition 21. *Given a n-dimensional linear recursive table over $\mathbb{K}$, Algorithm 8 computes its n univariate polynomials in $O(n\,\mathsf{M}(d)\log d)$ operations in $\mathbb{K}$ and $2\,n\,d^2$ queries to the table.*

Proof. Each table is made by combining d subsequences. Each call to the BM algorithm amounts for $O(\mathsf{M}(d)\log d)$ operations in $\mathbb{K}$ and $2\,d$ queries to the table elements, hence $2\,n\,d^2$ queries to table. □

Acknowledgements

We would like to thank the anonymous referees for their helpful comments and A. Bostan, F. Chyzak, C.-P. Jeannerod, E. Kaltofen, E. Mayr, T. Mora, B. Mourrain and B. Salvy for valuable discussions.

This work has been partly supported by the French National Research Agency ANR-11-BS02-0013 HPAC project.

References

Berlekamp, E., 1968. Nonbinary BCH decoding. IEEE Trans. Inform. Theory 14 (2), 242–242.

Berthomieu, J., Boyer, B., Faugère, J.-Ch., 2015. Linear Algebra for Computing Grbner Bases of Linear Recursive Multidimensional Sequences. In: 40th International Symposium on Symbolic and Algebraic Computation. Proceedings of the 40th International Symposium on Symbolic and Algebraic Computation. Bath, United Kingdom, pp. 61–68.
URL `https://hal.inria.fr/hal-01237861`

Bostan, A., Jeannerod, C.-P., Schost, É., 2007. Solving Toeplitz- and Vandermonde-like Linear Systems with Large Displacement Rank. In: Brown, C. W. (Ed.), ISSAC'07. ACM Press, pp. 33–40.

Bousquet-Mélou, M., Petkovšek, M., 2000. Linear recurrences with constant coefficients: the multivariate case. Discrete Math. 225 (13), 51 – 75, FPSAC'98.
URL `http://www.sciencedirect.com/science/article/pii/S0012365X00001473`

Brachat, J., Comon, P., Mourrain, B., Tsigaridas, E. P. P., 2010. Symmetric tensor decomposition. Linear Algebra Appl. 433 (11-12), 1851–1872.
URL `https://hal.inria.fr/inria-00355713`

Chabanne, H., Norton, G. H., 1992. On the key equation for n-dimensional cyclic codes: applications to decoding. Tech. report INRIA RR-1796.
URL `http://opac.inria.fr/record=b1029178`

Daleo, N. S., Hauenstein, J. D., 2015. Numerically testing generically reduced projective schemes for the arithmetic Gorenstein property, presented at MACIS 2015.

Elkadi, M., Mourrain, B., 2007. Introduction à la résolution des systèmes polynomiaux. Vol. 59 of Mathématiques et Applications. Springer.
URL `https://hal.inria.fr/inria-00170536`

Fasino, D., Tilli, P., 2000. Spectral clustering properties of block multilevel hankel matrices. Linear Algebra Appl. 306 (1-3), 155 – 163.
URL `http://www.sciencedirect.com/science/article/pii/S0024379599002517`

Faugère, J.-Ch., Gianni, P., Lazard, D., Mora, T., 1993. Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering. J. Symbolic Comput. 16 (4), 329–344.
URL http://www-salsa.lip6.fr/~jcf/Papers/FGLM.pdf

Faugère, J.-Ch., Mou, C., 2011. Fast Algorithm for Change of Ordering of Zero-dimensional Gröbner Bases with Sparse Multiplication Matrices. In: Proc. of the 36th ISSAC. ACM, pp. 115–122.
URL http://www-polsys.lip6.fr/~jcf/Papers/FM11.pdf

Fitzpatrick, P., Norton, G., 1990. Finding a basis for the characteristic ideal of an n-dimensional linear recurring sequence. IEEE Trans. Inform. Theory 36 (6), 1480–1487.

Gianni, P., Mora, T., 1989. Algebraic solution of systems of polynomial equations using Gröbner bases. In: Proc. of AAECC-5, vol. 356 of LNCS. Springer, pp. 247–257.

Gorenstein, D., 1952. An arithmetic theory of adjoint plane curves. Trans. Amer. Math. Soc. 72, 414–436.

Jonckheere, E., Ma, C., 1989. A simple Hankel interpretation of the Berlekamp-Massey algorithm. Linear Algebra Appl. 125 (0), 65 – 76.
URL http://www.sciencedirect.com/science/article/pii/0024379589900323

Kaltofen, E., Pan, V., 1991. Processor efficient parallel solution of linear systems over an abstract field. In: SPAA '91. ACM Press, New York, N.Y., pp. 180–191.

Kaltofen, E., Yuhasz, G., 2013a. A fraction free Matrix Berlekamp/Massey algorithm. Linear Algebra Appl. 439 (9), 2515–2526.

Kaltofen, E., Yuhasz, G., 2013b. On the Matrix Berlekamp-Massey Algorithm. ACM Trans. Algorithms 9 (4), 33:1–33:24.
URL http://doi.acm.org/10.1145/2500122

Koutschan, C., 2013. Creative Telescoping for Holonomic Functions. In: Schneider, C., Blümlein, J. (Eds.), Computer Algebra in Quantum Field Theory. Springer Vienna, pp. 171–194.
URL http://dx.doi.org/10.1007/978-3-7091-1616-6_7

Lakshman, Y. N., 1990. On the Complexity of Computing a Gröbner Basis for the Radical of a Zero Dimensional Ideal. In: Proc. of the 22nd Annual ACM STOC.

ACM, pp. 555–563.
URL http://doi.acm.org/10.1145/100216.100294

Levinson, N., 1947. The Wiener RMS (Root-Mean-Square) error criterion in the filter design and prediction. J. Math. Phys. 25, 261–278.

Macaulay, F. S., 1934. Modern algebra and polynomial ideals. Mathematical Proceedings of the Cambridge Philosophical Society 30, 27–46.
URL http://journals.cambridge.org/article_S0305004100012354

Massey, J. L., 1969. Shift-register synthesis and BCH decoding. IEEE Trans. Inform. Theory IT-15, 122–127.

Poteaux, A., Schost, É., 2013. On the complexity of computing with 0-dimensional triangular sets. J. Symbolic Comput. 50 (0), 110–138.
URL http://www.sciencedirect.com/science/article/pii/S0747717112001083

Ruzsa, I. Z., 1994. Generalized arithmetical progressions and sumsets. Acta Math. Hungar. 65 (4), 379–388.
URL http://dx.doi.org/10.1007/BF01876039

Saints, K., Heegard, C., 1995. Algebraic-geometric codes and multi- dimensional cyclic codes: Theory and algorithms for decoding using Gröbner bases. IEEE Trans. Inform. Theory 41 (6), 1733–1751.

Sakata, S., 1988. Finding a minimal set of linear recurring relations capable of generating a given finite two-dimensional array. J. Symbolic Comput. 5 (3), 321–337.
URL http://www.sciencedirect.com/science/article/pii/S0747717188800336

Sakata, S., 1990. Extension of the Berlekamp-Massey algorithm to *N* Dimensions. Inform. and Comput. 84 (2), 207–239.
URL http://dx.doi.org/10.1016/0890-5401(90)90039-K

Sakata, S., 1991. Decoding binary 2-D cyclic codes by the 2-D Berlekamp-Massey algorithm. IEEE Trans. Inform. Theory 37 (4), 1200–1203.
URL http://dx.doi.org/10.1109/18.86974

Sakata, S., 2009. The BMS algorithm. In: Sala, M., Mora, T., Perret, L., Sakata, S., Traverso, C. (Eds.), Gröbner Bases, Coding, and Cryptography. Springer Berlin Heidelberg, pp. 143–163.
URL http://dx.doi.org/10.1007/978-3-540-93806-4_9

Serra-Capizzano, S., 2002. More inequalities and asymptotics for matrix valued linear positive operators: the noncommutative case. In: Böttcher, A., Gohberg, I., Junghanns, P. (Eds.), Toeplitz Matrices and Singular Integral Equations. Vol. 135 of Operator Theory: Advances and Applications. Birkhuser Basel, pp. 293–315.
URL http://dx.doi.org/10.1007/978-3-0348-8199-9_18

Wiener, N., 1964. Extrapolation, Interpolation, and Smoothing of Stationary Time Series. The MIT Press.