



HAL
open science

A Polynomial-Division-Based Algorithm for Computing Linear Recurrence Relations

Jérémy Berthomieu, Jean-Charles Faugère

► **To cite this version:**

Jérémy Berthomieu, Jean-Charles Faugère. A Polynomial-Division-Based Algorithm for Computing Linear Recurrence Relations. 2018. <hal-01935229v1>

HAL Id: hal-01935229

<https://inria.hal.science/hal-01935229v1>

Preprint submitted on 26 Nov 2018 (v1), last revised 6 Jul 2021 (v3)

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

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

A Polynomial-Division-Based Algorithm for Computing Linear Recurrence Relations

Jérémy Berthomieu*, Jean-Charles Faugère

Sorbonne Université, CNRS, INRIA, Laboratoire d'Informatique de Paris 6, LIP6, Équipe PoLSys, 4 place Jussieu, F-75005, Paris, France

Abstract

Sparse polynomial interpolation, sparse linear system solving or modular rational reconstruction are fundamental problems in Computer Algebra. They come down to computing linear recurrence relations of a sequence with the Berlekamp–Massey algorithm. Likewise, sparse multivariate polynomial interpolation and multidimensional cyclic code decoding require guessing linear recurrence relations of a multivariate sequence.

Several algorithms solve this problem. The so-called Berlekamp–Massey–Sakata algorithm (1988) uses polynomial additions and shifts by a monomial. The SCALAR-FGLM algorithm (2015) relies on linear algebra operations on a multi-Hankel matrix, a multivariate generalization of a Hankel matrix. The Artinian Gorenstein border basis algorithm (2017) uses a Gram-Schmidt process.

We propose a new algorithm for computing the Gröbner basis of the ideal of relations of a sequence based solely on multivariate polynomial arithmetic. This algorithm allows us to both revisit the Berlekamp–Massey–Sakata algorithm through the use of polynomial divisions and to completely revise the SCALAR-FGLM algorithm without linear algebra operations.

A key observation in the design of this algorithm is to work on the mirror of the truncated generating series allowing us to use polynomial arithmetic modulo a monomial ideal. It appears to have some similarities with Padé approximants of this mirror polynomial.

As an addition from the paper published at the ISSAC conference, we give an adaptive variant of this algorithm taking into account the shape of the final Gröbner basis gradually as it is discovered. The main advantage of this algorithm is that its complexity in terms of operations and sequence queries only depends on the output Gröbner basis.

All these algorithms have been implemented in MAPLE and we report on our comparisons.

Keywords: Gröbner bases; linear recursive sequences; BERLEKAMP–MASSEY–SAKATA; extended Euclidean algorithm; Padé approximants

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

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

1. Introduction

The Berlekamp–Massey algorithm (BM), introduced by Berlekamp in 1968 Berlekamp (1968) and Massey in 1969 Massey (1969) is a fundamental algorithm in Coding Theory, Bose and Ray-Chaudhuri (1960); Hocquenghem (1959), and Computer Algebra. It allows one to perform efficiently sparse polynomial interpolation, sparse linear system solving or modular rational reconstruction.

In 1988, Sakata extended the BM algorithm to dimension n . This algorithm, known as the Berlekamp–Massey–Sakata algorithm (BMS), can be used to compute a Gröbner basis of the zero-dimensional ideal of the relations satisfied by a sequence, Sakata (1988, 1990, 2009). Analogously to dimension 1, the BMS algorithm allows one to decode cyclic code in dimension $n > 1$, an extension of Reed–Solomon’s codes. Furthermore, the latest versions of the SPARSE-FGLM algorithm rely heavily on the efficiency of the BMS algorithm to compute the change of ordering of a Gröbner basis, Faugère and Mou (2011); Faugère and Mou (2017).

1.1. Related Work

In dimension 1, it is well known that the BM algorithm can be seen in a matrix form requiring to solve a linear Hankel system of size D , the order of the recurrence, see Jonckheere and Ma (1989), or the Levinson–Durbin method, Levinson (1947); Wiener (1964). If we let $M(D)$ be a cost function for multiplying two polynomials of degree D , for instance $M(D) \in O(D \log D \log \log D)$, Cantor and Kaltofen (1991); Cooley and Tukey (1965), then solving a linear Hankel system of size D comes down to performing a truncated extended Euclidean algorithm called on two polynomials of degree D , Blackburn (1997); Brent et al. (1980); Dornstetter (1987). More precisely, it can be done in $O(M(D) \log D)$ operations.

In Berthomieu et al. (2015, 2017), the authors present the SCALAR-FGLM algorithm, extending the matrix version of the BM algorithm for multidimensional sequences. It consists in computing the relations of the sequence through the computation of a maximal full-rank matrix of a *multi-Hankel* matrix, a multivariate generalization of a Hankel matrix. Then, it returns the minimal Gröbner basis \mathcal{G} of the ideal of relations satisfied by the sequence. These notions are recalled in Section 2. If we denote by S the staircase defined by \mathcal{G} and T the input set of monomials containing $S \cup \mathcal{G}$, then the complexity of the SCALAR-FGLM algorithm is $O((\#T)^\omega)$, where $2 \leq \omega \leq 3$ is the linear algebra exponent. However, we do not know how to exploit the multi-Hankel structure to improve this complexity.

The ARTINIAN GORENSTEIN BORDER BASES algorithm (AGBB) was presented in Mourrain (2017) for computing a border basis \mathcal{B} of the ideal of relations. It extends the algorithm of Berthomieu et al. (2015) using polynomial arithmetic allowing it to reach the better complexity $O((\#S + \#\mathcal{B}) \cdot \#S \cdot \#T)$ with the above notation.

Another viewpoint is that computing linear recurrence relations can be seen as computing Padé approximants of a truncation of the generating series $\sum_{i_1, \dots, i_n \geq 0} w_{i_1, \dots, i_n} x_1^{i_1} \cdots x_n^{i_n}$. In Fitzpatrick and Flynn (1992), the authors extend the extended Euclidean algorithm for computing multivariate Padé approximants. Given a polynomial P and an ideal B , find polynomials F and C such that $P = \frac{F}{C} \bmod B$, where the leading monomials of F and C satisfy some constraints.

It is also worth noticing that we now know that both the BMS and the SCALAR-FGLM algorithms are not equivalent, see Berthomieu and Faugère (2017), i.e. it is not possible to tweak one algorithm to mimic the behavior of the other. However, if the input sequence is linear recurrent and sufficiently many sequence terms are visited, then both algorithms compute a Gröbner basis of the zero-dimensional ideal of relations.

1.2. Contributions

In all the paper, we assume that the input sets of the SCALAR-FGLM algorithm are the sets of all the monomials less than a given monomial. In order to improve the complexity of the algorithm, we will use polynomial arithmetic in all the operations. Even though they are not equivalent, this reduces the gap between the BMS and the SCALAR-FGLM algorithms and provides a unified presentation.

In Section 3, we present the BM, the BMS and the SCALAR-FGLM algorithms in a unified polynomial viewpoint. Using the mirror of the truncated generating series is a key ingredient letting us perform the computations modulo a specific monomial ideal B : a vector in the kernel of a multi-Hankel matrix is a polynomial C such that

$$\text{LM}(F) = \text{LM}(PC \bmod B) < t_C, \quad (1)$$

where P is the mirror of the truncated generating series, LM denotes the leading monomial and t_C is a monomial associated to C .

One interpretation of this is the computation of multivariate Padé approximants $\frac{E}{C}$ of P modulo B with different constraints than in Fitzpatrick and Flynn (1992) since we require that $\text{LM}(C)$ is in a given set of terms and $\text{LM}(F)$ satisfies equation (1).

This polynomial point of view allows us to design the POLYNOMIAL SCALAR-FGLM algorithm (Algorithm 4.18) in Section 4 based on multivariate polynomial divisions. It computes polynomials whose product with P modulo B must satisfy equation (1). If they do not, by polynomial divisions, we make new ones until finding minimal polynomials satisfying this constraint. It is worth noticing that in dimension 1, we recover the truncated extended Euclidean algorithm applied to the mirror polynomial of the generated series of the input sequence, truncated in degree D , and x^{D+1} . All the examples are available on Berthomieu and Faugère (2018).

Our main result is Theorem 21, a simplified version of which is

Theorem 1. *Let \mathbf{w} be a sequence, $<$ be a total degree monomial ordering and a be a monomial. Let us assume that the Gröbner basis \mathcal{G} of the ideal of relations of \mathbf{w} for $<$ and its staircase S satisfy $a \geq \max(S \cup \text{LM}(\mathcal{G}))$ and for all $g \leq a$, $s = \max_{\sigma \leq a} \{\sigma, \sigma g \leq a\}$, we have $\max(S) \leq s$. Then, the POLYNOMIAL SCALAR-FGLM algorithm terminates and computes a Gröbner basis of the ideal of relations of \mathbf{w} for $<$ in $O(\#S (\#S + \#\mathcal{G}) \#\{\sigma, \sigma \leq a\})$ operations in the base field.*

In applications such as the SPARSE-FGLM one, sequence queries are costly, Faugère and Mou (2017). In Berthomieu et al. (2015), an adaptive variant of the SCALAR-FGLM algorithm was designed aiming to minimize the number of sequence queries to recover the relations.

In Section 5, we show how we can transform the ADAPTIVE SCALAR-FGLM algorithm of Berthomieu et al. (2015) into an algorithm using polynomial arithmetic. This algorithm is output sensitive and probabilistic, like the ADAPTIVE SCALAR-FGLM algorithm is. However, its main advantage is that its complexity only depends on the sizes of the computed staircase and Gröbner basis.

Theorem 2 (see Theorem 24). *Let \mathbf{w} be a sequence, $<$ be a total degree monomial ordering.*

Let us assume that calling the ADAPTIVE POLYNOMIAL SCALAR-FGLM algorithm on \mathbf{w} and $<$ yields the Gröbner basis \mathcal{G} and its staircase S .

Then, the ADAPTIVE POLYNOMIAL SCALAR-FGLM algorithm performs at most $O((\#S + \#\mathcal{G})^2 \#2S)$ operations in the base field and $\#2(S \cup \text{LM}(\mathcal{G}))$ table queries to recover \mathcal{G} , where for a set T , $2T = \{t t', t, t' \in T\}$.

Finally, in Section 6, we compare the POLYNOMIAL SCALAR-FGLM algorithm with our implementations of the BMS, the SCALAR-FGLM and the AGBB algorithms. Our algorithm performs always fewer arithmetic operations than the others starting from a certain size. Even for an example family favorable towards the BMS algorithm, our algorithm performs better.

Although we have compared the numbers of arithmetic operations, it would be beneficial to have an efficient implementation. This would be the first step into designing a similar efficient algorithm for computing linear recurrence relations with polynomial coefficients, extending the Beckermann–Labahn algorithm Beckermann and Labahn (1994) for computing multivariate Hermite–Padé approximants.

Amongst the changes from the ISSAC version of the paper, Berthomieu and Faugère (2018), the main addition is a full description of the ADAPTIVE POLYNOMIAL SCALAR-FGLM algorithm, an adaptive variant of the POLYNOMIAL SCALAR-FGLM algorithm using polynomial divisions as well, in Section 5.2. Generically, one could expect to make one relation with leading monomial $m x_i^2$ through a division of polynomials related to relations with leading monomials m and $m x_i$. Yet, the naive approach given in (Berthomieu and Faugère, 2018, Section 5) could not do so as it does not perform any division. The ADAPTIVE POLYNOMIAL SCALAR-FGLM algorithm visits the monomials in the same order as the ADAPTIVE SCALAR-FGLM algorithm to recover the relations and replaces any linear algebra computations by polynomial ones, see Berthomieu et al. (2015). We also give the complexity of this algorithm in terms both of the number of operations and the number of sequence queries.

Furthermore, one of the main obstructions to the design of this adaptive variant is that at each step, some polynomials are updated. This update process adds terms supposed to be small with respect to the ordering. Yet, their leading terms were not stable during this update process.

Lastly, in Section 3, we now more clearly define the condition for a relation to be valid (see Proposition 13) in the execution of the POLYNOMIAL SCALAR-FGLM algorithm.

2. Notation

We give a brief description of classical notation used in the paper.

2.1. Sequences and relations

For $n \geq 1$, we let $\mathbf{i} = (i_1, \dots, i_n) \in \mathbb{N}^n$ and for $\mathbf{x} = (x_1, \dots, x_n)$, we write $\mathbf{x}^{\mathbf{i}} = x_1^{i_1} \cdots x_n^{i_n}$.

Definition 3. Let \mathbb{K} be a field, $\mathcal{K} \subseteq \mathbb{N}^n$ be finite and $f = \sum_{\mathbf{k} \in \mathcal{K}} \gamma_{\mathbf{k}} \mathbf{x}^{\mathbf{k}} \in \mathbb{K}[\mathbf{x}]$. We let $[f]_{\mathbf{w}}$, or $[f]$, be the linear combination $\sum_{\mathbf{k} \in \mathcal{K}} \gamma_{\mathbf{k}} w_{\mathbf{k}}$. If for all $\mathbf{i} \in \mathbb{N}^n$, $[\mathbf{x}^{\mathbf{i}} f] = 0$, then we say that f is the polynomial of the relation induced by $\gamma = (\gamma_{\mathbf{k}})_{\mathbf{k} \in \mathcal{K}} \in \mathbb{K}^{\#\mathcal{K}}$.

The main benefit of the $[\]$ notation resides in the immediate fact that for all index \mathbf{i} , its shift by $\mathbf{x}^{\mathbf{i}}$ is $[\mathbf{x}^{\mathbf{i}} f] = \sum_{\mathbf{k} \in \mathcal{K}} \gamma_{\mathbf{k}} w_{\mathbf{k}+\mathbf{i}}$.

Example 4. Let $\mathbf{b} = \left(\binom{i}{j} \right)_{(i,j) \in \mathbb{N}^2}$ be the sequence of the binomial coefficients. Then, $xy - y - 1$ is the polynomial of Pascal’s rule:

$$\forall (i, j) \in \mathbb{N}^2, [x^i y^j (xy - y - 1)] = \mathbf{b}_{i+1, j+1} - \mathbf{b}_{i, j+1} - \mathbf{b}_{i, j} = 0.$$

Definition 5 (Fitzpatrick and Norton (1990); Sakata (1988)). Let $\mathbf{w} = (w_{\mathbf{i}})_{\mathbf{i} \in \mathbb{N}^n}$ be an n -dimensional sequence with coefficients in \mathbb{K} . The sequence \mathbf{w} is linear recurrent if from a nonzero finite number of initial terms $\{w_{\mathbf{i}}, \mathbf{i} \in S\}$, and a finite number of relations, without any contradiction, one can compute any term of the sequence.

Equivalently, \mathbf{w} is linear recurrent if $\{f, \forall m \in \mathbb{K}[\mathbf{x}], [m f] = 0\}$, its ideal of relations, is zero-dimensional.

As the input parameters of the algorithms are the first terms of a sequence, a *table* shall denote a finite subset of terms of a sequence.

2.2. Gröbner bases

Let $\mathcal{T} = \{\mathbf{x}^{\mathbf{i}}, \mathbf{i} \in \mathbb{N}^n\}$ be the set of all monomials in $\mathbb{K}[\mathbf{x}]$. A monomial ordering $<$ on $\mathbb{K}[\mathbf{x}]$ is an order relation satisfying the following three classical properties:

1. for all $m \in \mathcal{T}$, $1 \leq m$;
2. for all $m, m', s \in \mathcal{T}$, $m < m' \Rightarrow m s < m' s$;
3. every subset of \mathcal{T} has a least element for $<$.

For a monomial ordering $<$ on $\mathbb{K}[\mathbf{x}]$, the *leading monomial* of f , denoted $\text{LM}(f)$, is the greatest monomial in the support of f for $<$. The *leading coefficient* of f , denoted $\text{LC}(f)$, is the nonzero coefficient of $\text{LM}(f)$. The *leading term* of f , $\text{LT}(f)$, is defined as $\text{LT}(f) = \text{LC}(f) \text{LM}(f)$. For an ideal I , we denote $\text{LM}(I) = \{\text{LM}(f), f \in I\}$.

We recall briefly the definition of a Gröbner basis and a staircase.

Definition 6. Let I be a nonzero ideal of $\mathbb{K}[\mathbf{x}]$ and let $<$ be a monomial ordering. A set $\mathcal{G} \subseteq I$ is a Gröbner basis of I if for all $f \in I$, there exists $g \in \mathcal{G}$ such that $\text{LM}(g) \mid \text{LM}(f)$.

A Gröbner basis \mathcal{G} of I is minimal if for any $g \in \mathcal{G}$, $\langle \mathcal{G} \setminus \{g\} \rangle \neq I$.

Furthermore, \mathcal{G} is reduced if for any $g, g' \in \mathcal{G}$, $g \neq g'$ and any monomial $m \in \text{supp } g'$, $\text{LM}(g) \nmid m$.

The staircase of \mathcal{G} is defined as $S = \text{Staircase}(\mathcal{G}) = \{s \in \mathcal{T}, \forall g \in \mathcal{G}, \text{LM}(g) \nmid s\}$. It is also the canonical basis of $\mathbb{K}[\mathbf{x}]/I$.

Gröbner basis theory allows us to choose any monomial ordering, among which, we mainly use the

LEX($x_n < \dots < x_1$) **ordering** which satisfies $\mathbf{x}^{\mathbf{i}} < \mathbf{x}^{\mathbf{i}'}$ if and only if there exists k , $1 \leq k \leq n$ such that for all $\ell < k$, $i_{\ell} = i'_{\ell}$ and $i_k < i'_k$, see (Cox et al., 2015, Chapter 2, Definition 3);

DRL($x_n < \dots < x_1$) **ordering** which satisfies $\mathbf{x}^{\mathbf{i}} < \mathbf{x}^{\mathbf{i}'}$ if and only if $i_1 + \dots + i_n < i'_1 + \dots + i'_n$ or $i_1 + \dots + i_n = i'_1 + \dots + i'_n$ and there exists k , $2 \leq k \leq n$ such that for all $\ell > k$, $i_{\ell} = i'_{\ell}$ and $i_k > i'_k$, see (Cox et al., 2015, Chapter 2, Definition 6).

However, in the BMS algorithm, we need to be able to enumerate all the monomials up to a bound monomial. This forces the user to take an ordering $<$ such that for all $M \in \mathcal{T}$, the set $\mathcal{T}_{\leq a} = \{m \leq a, m \in \mathcal{T}\}$ is finite. Such an ordering $<$ makes $(\mathbb{N}^n, <)$ isomorphic to $(\mathbb{N}, <)$. Hence, for a monomial m , it makes sense to speak about the previous (resp. next) monomial m^- (resp. m^+) for $<$. The DRL ordering is an example for an ordering on which every term other than 1 has an immediate predecessor.

This request excludes for instance the LEX ordering, and more generally any elimination ordering. In other words, only weighted degree ordering, or *weight ordering*, should be used.

Now that a monomial ordering is defined, we can say that a relation given by a polynomial $f \in \mathbb{K}[\mathbf{x}]$ fails when shifted by s if for all monomials $\sigma < s$, $[\sigma f] = 0$ but $[s f] \neq 0$, see also Sakata (1990, 2009).

2.3. Multi-Hankel matrices

A matrix $H \in \mathbb{K}^{m \times n}$ is *Hankel*, if there exists a sequence $\mathbf{w} = (w_i)_{i \in \mathbb{N}}$ such that for all $(i, i') \in \{1, \dots, m\} \times \{1, \dots, n\}$, the coefficient $h_{i, i'}$ lying on the i th row and i' th column of H satisfies $h_{i, i'} = w_{i+i'}$.

In a multivariate setting, we can extend this notion to *multi-Hankel* matrices. For two sets of monomials U and T , we let $H_{U,T}$ be the multi-Hankel matrix with rows (resp. columns) indexed with U (resp. T) so that the coefficient of $H_{U,T}$ lying on the row labeled with $\mathbf{x}^i \in U$ and column labeled with $\mathbf{x}^{i'} \in T$ is $w_{i+i'}$.

Example 7. Let $\mathbf{w} = (w_{i,j,k})_{(i,j,k) \in \mathbb{N}^3}$ be a sequence.

- For $U = \{1, z, z^2, y, yz, yz^2\} = \{1, z, z^2\} \cup y\{1, z, z^2\}$ and $T = \{1, z, y, yz, y^2, y^2z\} = \{1, z\} \cup y\{1, z\} \cup y^2\{1, z\}$, ordered for $\text{LEX}(z < y < x)$,

$$H_{U,T} = \begin{array}{c} \begin{array}{c} 1 \\ z \\ z^2 \\ y \\ yz \\ yz^2 \end{array} \end{array} \left(\begin{array}{cc|cc|cc} 1 & z & y & yz & y^2 & y^2z \\ \hline w_{0,0,0} & w_{0,0,1} & w_{0,1,0} & w_{0,1,1} & w_{0,2,0} & w_{0,2,1} \\ w_{0,0,1} & w_{0,0,2} & w_{0,1,1} & w_{0,1,2} & w_{0,2,1} & w_{0,2,2} \\ w_{0,0,2} & w_{0,0,3} & w_{0,1,2} & w_{0,1,3} & w_{0,2,2} & w_{0,2,3} \\ \hline w_{0,1,0} & w_{0,1,1} & w_{0,2,0} & w_{0,2,1} & w_{0,3,0} & w_{0,3,1} \\ w_{0,1,1} & w_{0,1,2} & w_{0,2,1} & w_{0,2,2} & w_{0,3,1} & w_{0,3,2} \\ w_{0,1,2} & w_{0,1,3} & w_{0,2,2} & w_{0,2,3} & w_{0,3,2} & w_{0,3,3} \end{array} \right)$$

is a 2×3 -block-Hankel matrix with 3×2 -Hankel blocks.

- For $\tilde{U} = U \cup xU \cup x^2U \cup x^3U$ and $\tilde{T} = T \cup xT \cup x^2T \cup x^3T \cup x^4T$, also ordered for $\text{LEX}(z < y < x)$,

$$H_{\tilde{U}, \tilde{T}} = \begin{array}{c} \begin{array}{c} U \\ xU \\ x^2U \\ x^3U \end{array} \end{array} \left(\begin{array}{c|c|c|c|c} T & xT & x^2T & x^3T & x^4T \\ \hline H_{U,T} & H_{U,xT} & H_{U,x^2T} & H_{U,x^3T} & H_{U,x^4T} \\ \hline H_{xU,T} & H_{xU,xT} & H_{xU,x^2T} & H_{xU,x^3T} & H_{xU,x^4T} \\ \hline H_{x^2U,T} & H_{x^2U,xT} & H_{x^2U,x^2T} & H_{x^2U,x^3T} & H_{x^2U,x^4T} \\ \hline H_{x^3U,T} & H_{x^3U,xT} & H_{x^3U,x^2T} & H_{x^3U,x^3T} & H_{x^3U,x^4T} \end{array} \right),$$

where $H_{x^i U, x^{i'} T} = H_{x^{i+i'} U, T}$ for any i, i' since $H_{x^i U, x^{i'} T}$ is the same matrix as $H_{U,T}$ where each coefficient $w_{0,j,k}$ has been replaced by $w_{i+i', j, k}$. Therefore, $H_{\tilde{U}, \tilde{T}}$ is a 4×5 -block-Hankel matrix with 6×6 -multi-Hankel blocks like $H_{U,T}$.

- For $T = \{1, y, x, y^2, xy, x^2\}$, ordered for $\text{DRL}(z < y < x)$, $H_{T,T}$ is a multi-Hankel matrix whose structure is less clear. It can be considered as a block-Hankel matrix with blocks of different sizes, noticing that $T = \{1\} \cup y\{1, \frac{x}{y}\} \cup y^2\{1, \frac{x}{y}, \frac{x^2}{y^2}\}$,

$$H_{T,T} = \begin{array}{c} \begin{array}{c} 1 \\ y \\ x \\ y^2 \\ xy \\ x^2 \end{array} \end{array} \left(\begin{array}{c|cc|cc|c} 1 & y & x & y^2 & xy & x^2 \\ \hline w_{0,0,0} & w_{0,1,0} & w_{1,0,0} & w_{0,2,0} & w_{1,1,0} & w_{2,0,0} \\ w_{0,1,0} & w_{0,2,0} & w_{1,1,0} & w_{0,3,0} & w_{1,2,0} & w_{2,1,0} \\ w_{1,0,0} & w_{1,1,0} & w_{2,0,0} & w_{1,2,0} & w_{2,1,0} & w_{3,0,0} \\ \hline w_{0,2,0} & w_{0,3,0} & w_{1,2,0} & w_{0,4,0} & w_{1,3,0} & w_{2,2,0} \\ w_{1,1,0} & w_{1,2,0} & w_{2,1,0} & w_{1,3,0} & w_{2,2,0} & w_{3,1,0} \\ w_{2,0,0} & w_{2,1,0} & w_{3,0,0} & w_{2,2,0} & w_{3,1,0} & w_{4,0,0} \end{array} \right).$$

4. For $U = \{1, z, y, x, z^2, yz, xz, y^2, xy, x^2\} = \{1\} \cup z \left\{1, \frac{y}{z}, \frac{x}{z}\right\} \cup z^2 \left\{1, \frac{y}{z}, \frac{x}{z}, \frac{y^2}{z^2}, \frac{xy}{z^2}, \frac{x^2}{z^2}\right\}$, also ordered for $\text{DRL}(z < y < x)$, the matrix $H_{U,U}$ can be seen as a block-matrix like $H_{T,T}$ except each block is a multi-Hankel matrix in two variables. In fact, the bottom-right block would be the same as $H_{T,T}$ where each coefficient $w_{i,j,0}$ is replaced by $w_{i,j,4-i-j}$.

2.4. Polynomials associated to multi-Hankel matrices

For two sets of terms T and U , we let $T + U$ denote their Minkowsky sum, i.e. $T + U = \{t + u, t \in T, u \in U\}$, and $2T = T + T$.

For a set of terms T , we let $M = \text{LCM}(T)$. We let P_T be the mirror polynomial of the truncated generating series of a sequence \mathbf{w} , i.e.

$$P_T = \sum_{t \in T} [t] \frac{M}{t}.$$

Example 8. Let $\mathbf{w} = (w_{i,j,k})_{(i,j,k) \in \mathbb{N}^3}$ be a sequence and $T = \{1, z, y, x, z^2, yz\}$, then $M = xy z^2$ and

$$\begin{aligned} P_T &= [1] xy z^2 + [z] xyz + [y] xz^2 + [x] yz^2 + [z^2] xy + [yz] xz \\ &= w_{0,0,0} xy z^2 + w_{0,0,1} xyz + w_{0,1,0} xz^2 + w_{1,0,0} yz^2 + w_{0,0,2} xy + w_{0,1,1} xz. \end{aligned}$$

In this paper, we will mostly deal with polynomials P_{T+U} as there is a strong connection between $H_{U,T}$ and P_{T+U} .

Finally, letting $M = \text{LCM}(T+U) = x_1^{D_1} \cdots x_n^{D_n}$ and B be the monomial ideal $(x_1^{D_1+1}, \dots, x_n^{D_n+1})$, we will use pairs of multivariate polynomials $R_m = [F_m, C_m]$ where $\text{LM}(C_m) = m$ and $F_m = P_{T+U} C_m \bmod B$.

3. From matrices to polynomials

Before detailing the unified polynomial viewpoint, we recall the linear algebra viewpoint of the BM, the BMS and the SCALAR-FGLM algorithms.

3.1. The BM algorithm

Let $\mathbf{w} = (w_i)_{i \in \mathbb{N}}$ be a one-dimensional table. Classically, when calling the BM algorithm, one does not know in advance the order of the output relation. Therefore, from a matrix viewpoint, one wants to compute the greatest collection of vectors

$$\begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_{x^{d-1}} \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \gamma_1 \\ \vdots \\ \gamma_{x^{d-1}} \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \dots, \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \gamma_1 \\ \vdots \\ \gamma_{x^{d-1}} \\ 1 \end{pmatrix}$$

in the kernel of $H_{\{1, \dots, x^D\}} = \begin{pmatrix} 1 & \dots & x^D \\ w_0 & \dots & w_D \end{pmatrix}$, that is $\gamma_1, \dots, \gamma_{x^{d-1}}$ such that the relation $[C_{x^d}] = w_d + \sum_{k=0}^{d-1} \gamma_{x^k} w_k$ and its shifts, $[x C_{x^d}], \dots, [x^{D-d} C_{x^d}]$, are all 0. Equivalently, we look

for the least d such that $H_{\mathcal{T}_{\leq x^{D-d}}, \mathcal{T}_{\leq x^d}} \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_{x^{d-1}} \\ 1 \end{pmatrix} = 0$.

This Hankel matrix-vector product can be extended into

$$\begin{pmatrix} w_0 & \cdots & w_{d-1} & w_d \\ w_1 & \cdots & w_d & w_{d+1} \\ \vdots & & \vdots & \vdots \\ w_{D-d} & \cdots & w_{D-1} & w_D \\ w_{D-d+1} & \cdots & w_D & 0 \\ \vdots & \ddots & \ddots & \vdots \\ w_D & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_{x^{d-1}} \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ f_{x^{d-1}} \\ \vdots \\ f_1 \end{pmatrix}, \quad (2)$$

representing the product of polynomials $P_{\mathcal{T}_{\leq x^D}} = \sum_{i=0}^D w_i x^{D-i}$ and $C_{x^d} = x^d + \sum_{k=0}^{d-1} \gamma_{x^k} x^k$ modulo $B = x^{D+1}$. The requirement for C_{x^d} to encode a valid relation is now that $\text{LM}(F_{x^d}) < x^d$ with $F_{x^d} = P_{\mathcal{T}_{\leq x^D}} C_{x^d} \bmod B$.

This viewpoint gave rise to the following version of the BM algorithm: Start with $R_B = [F_B, C_B] = [B, 0]$ and $B = x^{D+1}$, and $R_1 = [F_1, C_1] = [P_{\mathcal{T}_{\leq x^D}}, 1]$. Compute Q the quotient of the Euclidean division of $F_B = B$ by F_1 and then compute $R_{\text{LM}(Q)} = R_B - Q R_1 = [F_B - Q F_1, C_B - Q C_1] = [F_{\text{LM}(Q)}, C_{\text{LM}(Q)}]$. Repeat with R_1 and $R_{\text{LM}(Q)}$ until reaching a pair $R_{x^d} = [P_{\mathcal{T}_{\leq x^D}} C_{x^d} \bmod B, C_{x^d}] = [F_{x^d}, C_{x^d}]$ with $\text{LM}(C_{x^d}) = x^d$ and $\text{LM}(F_{x^d}) < x^d$. This is in fact the extended Euclidean algorithm called on $B = x^{D+1}$ and F_1 without any computation of the Bézout's cofactors of x^{D+1} .

Example 9. Let us consider the Fibonacci table $\mathbf{F} = (F_i)_{i \in \mathbb{N}}$ with $F_0 = F_1 = 1$ and assume $D = 5$. On the one hand, although the kernel of

$$H_{\{1, \{1, x, x^2, x^3, x^4, x^5\}\}} =_1 \begin{pmatrix} 1 & x & x^2 & x^3 & x^4 & x^5 \\ 1 & 1 & 2 & 3 & 5 & 8 \end{pmatrix}$$

has dimension 5 and $\begin{pmatrix} -1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ is in this kernel, it corresponds to $[x - 1] = 0$, its shifted vectors

$\begin{pmatrix} 0 \\ -1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 1 \end{pmatrix}$ are not, as they correspond to $[x^i(x-1)] \neq 0$, for $1 \leq i \leq 4$. However, these vectors

$$\begin{pmatrix} -1 \\ -1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \\ -1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ -1 \\ -1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ 1 \end{pmatrix}$$

are in the kernel and form the greatest collection of shifted vectors as such. They correspond to $[x^i(x^2 - x - 1)] = 0$, for $0 \leq i \leq 3$. Finally, we have

$$\begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 5 \\ 3 & 5 & 8 \\ 5 & 8 & 0 \\ 8 & 0 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -13 \\ -8 \end{pmatrix},$$

where the gray zeroes (0) are due to the matrix extension and not the sequence itself.

On the other hand, $B = x^6$, $R_B = [B, 0]$ and $R_1 = [x^5 + x^4 + 2x^3 + 3x^2 + 5x + 8, 1]$. As we can see $R_1 = [F_1, C_1]$ with $\text{LM}(C_1) = 1$ and $\text{LM}(F_1) = x^5 \geq 1$.

The first step of the extended Euclidean algorithm yields $R_x = [x^4 + x^3 + 2x^2 + 3x - 8, x - 1] = [F_x, C_x]$ with $\text{LM}(C_x) = x$ and $\text{LM}(F_x) = x^4 \geq x$.

Then, the second step yields $R_{x^2} = [-13x - 8, x^2 - x - 1] = [F_{x^2}, C_{x^2}]$ with $\text{LM}(C_{x^2}) = x^2$ and $\text{LM}(F_{x^2}) = x < x^2$ so C_{x^2} is a valid relation. We return C_{x^2} .

Remark 10. The BM algorithm always returns a relation. If no pair $R_{x^d} = [F_{x^d}, C_{x^d}]$ satisfies the requirements, then it will return a pair R_{x^d} with $\text{LM}(C_{x^d}) > x^d$. From a matrix viewpoint, it returns an element of the kernel of the empty matrix $H_{0, \mathcal{T}_{\leq x^d}}$.

3.2. Multidimensional extension

For a multidimensional table $\mathbf{w} = (w_i)_{i \in \mathbb{N}^n}$, the BMS algorithm extends the BM algorithm by computing vectors in the kernel of a multi-Hankel matrix

$$H_{(1), \mathcal{T}_{\leq a}} = 1 \begin{pmatrix} 1 & \dots & a^- & a \\ [1] & \dots & [a^-] & [a] \end{pmatrix}$$

corresponding to having relations $[C_g] = 0$, with $\text{LM}(C_g) = g$ minimal for $|$ and for all t such that $t g \leq a$, $[t C_g] = 0$ as well. This also comes down to finding the least (for the partial order $|$) monomials $g_1, \dots, g_r \leq a$ such that $\dim \ker H_{\mathcal{T}_{\leq s_k}, \mathcal{T}_{\leq g_k}} > 0$ with s_k the greatest monomial such that $s_k g_k \leq a$ for all k , $1 \leq k \leq r$. Then, each multi-Hankel matrix-vector product can be extended further as in equation 2, taking the multi-Hankel matrix $H_{\mathcal{T}_{\leq a}, \mathcal{T}_{\leq g_k}}$ and setting to zero any sequence term $[t u]$ with $t u \notin \mathcal{T}_{\leq a}$.

$$\begin{pmatrix} [1] & \dots & [g_k^-] & g_k \\ [1^+] & \dots & [1^+ g_k^-] & [1^+ g_k] \\ \vdots & & \vdots & \vdots \\ [s_k] & \dots & [s_k g_k^-] & [s_k g_k] \\ [s_k^+] & \dots & [s_k^+ g_k^-] & 0 \\ \vdots & & \ddots & \vdots \\ [a] & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_{g_k^-} \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ f_{M/s_k^+} \\ \vdots \\ f_{M/a} \end{pmatrix}, \quad (3)$$

where $M = \text{LCM}(\mathcal{T}_{\leq a}) = x_1^{D_1} \dots x_n^{D_n}$. It then represents the product of polynomials $P_{\mathcal{T}_{\leq a}} = \sum_{t \leq a} [t] \frac{M}{t}$ and $C_{g_k} = g_k + \sum_{t < g_k} \gamma_t t$ modulo $B = (x_1^{D_1+1}, \dots, x_n^{D_n+1})$. The requirement for C_{g_k} to encode a valid relation is now that $\text{LM}(F_{g_k}) < \frac{M}{s_k}$ with $F_{g_k} = P_{\mathcal{T}_{\leq a}} C_{g_k} \bmod B$.

Let us notice that $[s_k^+ g_k^-]$ can also be a 0 if $s_k^+ g_k^- > a$ and that, more generally, the gray zeroes need not be diagonally aligned like they are in the univariate case. This is illustrated by the following example.

Example 11. Let us consider the binomial table $\mathbf{b} = \left(\binom{i}{j} \right)_{(i,j) \in \mathbb{N}^2}$ with $\text{DRL}(y < x)$ and assume $a = x^2 y$. The kernel of

$$H_{(1), \mathcal{T}_{\leq x^2 y}} = 1 \begin{pmatrix} 1 & y & x & y^2 & xy & x^2 & y^3 & xy^2 & x^2 y \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 2 \end{pmatrix}$$

has clearly dimension 8. Two vectors in the kernel, $\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} -1 \\ \alpha \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, with α any, corresponds to the independant relation $[y] = [x + \alpha y - 1] = 0$. However, not all their shifts belong to the kernel. For the former, $\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, corresponding to $[y^2] = 0$, belongs to the kernel but $\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ does not, as $[xy] \neq 0$. For the latter, $\begin{pmatrix} 0 \\ -1 \\ \alpha \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ does not belong to the kernel, as $[xy + \alpha y^2 - y] \neq 0$, whatever α is.

Therefore, the vectors in the kernel that we seek must correspond to relations $[C_g] = 0$ with $g \in \{y^2, xy, x^2, \dots, x^2 y\}$.

In fact, as $[m y^2] = 0$ for $1 \leq m \leq x$, the vectors $\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, $\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, $\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$, are in the kernel and fulfill the requirements. We can indeed notice that

$$H_{\mathcal{T}_{\leq x}, \mathcal{T}_{\leq y^2}} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{matrix} 1 \\ y \\ x \end{matrix} \begin{pmatrix} 1 & y & x & y^2 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad \begin{matrix} 1 \\ y \\ x \\ y^2 \\ xy \\ x^2 \\ y^3 \\ xy^2 \\ x^2 y \end{matrix} \begin{pmatrix} 1 & y & x & y^2 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

where the gray zeroes (0) are due to the matrix extension and not the binomial sequence itself. From the polynomial point of view,

$$\begin{aligned} F_{y^2} &= P_{\mathcal{T}_{\leq x^2 y}} C_{y^2} \bmod B \\ &= (x^2 y^3 + x y^3 + x y^2 + y^3 + 2 y^2) y^2 \bmod (x^3, y^4) \\ &= 0. \end{aligned}$$

Likewise, the vectors $\begin{pmatrix} -(1+\alpha) \\ \beta \\ \alpha \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, $\begin{pmatrix} 0 \\ -(1+\alpha) \\ 0 \\ \beta \\ \alpha \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$, $\begin{pmatrix} 0 \\ 0 \\ -(1+\alpha) \\ 0 \\ \beta \\ \alpha \\ 0 \\ 0 \\ 1 \end{pmatrix}$, are in the kernel for $\alpha = 0$ and $\beta = -1$, as they correspond to $[m(xy + \alpha x + \beta y - (1 + \alpha))] = [m(xy - y - 1)] = 0$, for $1 \leq m \leq x$.

Furthermore,

$$\begin{array}{l}
 1 \\
 y \\
 x \\
 y^2 \\
 xy \\
 x^2 \\
 y^3 \\
 xy^2 \\
 x^2y
 \end{array}
 \begin{array}{c}
 1 \quad y \quad x \quad y^2 \quad xy \\
 \begin{pmatrix}
 1 & 0 & 1 & 0 & 1 \\
 0 & 0 & 1 & 0 & 0 \\
 1 & 1 & 1 & 0 & 2 \\
 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & 2 & 0 & 0 \\
 1 & 2 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 \\
 2 & 0 & 0 & 0 & 0
 \end{pmatrix}
 \begin{pmatrix}
 -1 \\
 -1 \\
 0 \\
 0 \\
 1
 \end{pmatrix}
 =
 \begin{pmatrix}
 0 \\
 0 \\
 0 \\
 -1 \\
 -3 \\
 0 \\
 0 \\
 0 \\
 -2
 \end{pmatrix},
 \end{array}$$

From the polynomial point of view, $F_{xy} = P_{\mathcal{T}_{\leq xy}}(xy - y - 1) \bmod (x^3, y^4) = -xy^2 - 3y^3 - 2y^2$.

Finally, the vectors $\begin{pmatrix} -(1+\alpha) \\ \beta \\ \alpha \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, $\begin{pmatrix} 0 \\ -(1+\alpha) \\ 0 \\ \beta \\ \alpha \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$, are in the kernel for $\alpha = -2$ and $\beta = 0$, as they correspond to $[m(x^2 + \alpha x + \beta y - (1 + \alpha))] = [m(x^2 - 2x + 1)] = 0$, for $1 \leq m \leq y$. Furthermore,

$$\begin{array}{l}
 1 \\
 y \\
 x \\
 y^2 \\
 xy \\
 x^2 \\
 y^3 \\
 xy^2 \\
 x^2y
 \end{array}
 \begin{array}{c}
 1 \quad y \quad x \quad y^2 \quad xy \quad x^2 \\
 \begin{pmatrix}
 1 & 0 & 1 & 0 & 1 & 1 \\
 0 & 0 & 1 & 0 & 0 & 2 \\
 1 & 1 & 1 & 0 & 2 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & 2 & 0 & 0 & 0 \\
 1 & 2 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 \\
 2 & 0 & 0 & 0 & 0 & 0
 \end{pmatrix}
 \begin{pmatrix}
 1 \\
 0 \\
 -2 \\
 0 \\
 0 \\
 1
 \end{pmatrix}
 =
 \begin{pmatrix}
 0 \\
 0 \\
 -1 \\
 0 \\
 -3 \\
 1 \\
 0 \\
 0 \\
 2
 \end{pmatrix}.
 \end{array}$$

From the polynomial point of view, $F_{x^2} = P_{\mathcal{T}_{\leq x^2}}(x^2 - 2x + 1) \bmod (x^3, y^4) = -xy^3 - 3xy^2 + y^3 + 2y^2$.

The BMS algorithm will return the three relations $C_{y^2} = y^2$, $C_{xy} = xy - y - 1$ and $C_{x^2} = x^2 - 2x + 1$.

Remark 12. As for the BM algorithm, the BMS algorithm will always return a relation C_g with $\text{LM}(C_g) = g$ a pure power in each variable. Therefore, it can return C_g with $g > a$, corresponding to a vector in the kernel of the empty matrix $H_{0, \mathcal{T}_{\leq g}}$.

The SCALAR-FGLM algorithm corresponds to computing vectors in the kernel of a more general multi-Hankel matrix $H_{U, T}$, with T and U two ordered sets of monomials, corresponding also to relations $[C_g] = 0$ with $\text{LM}(C_g) = g$, such that for all $t \in \mathcal{T}$, if $t \in T$, then $[tC_g] = 0$, i.e. the vectors corresponding to tC_g are also in the kernel of $H_{U, T}$. We now consider that both sets of terms T and U satisfy $T = \mathcal{T}_{\leq a}$ and $U = \mathcal{T}_{\leq b}$. This allows us to encompass both the BMS algorithm and the SCALAR-FGLM algorithm.

The multi-Hankel matrix-vector product

$$\begin{array}{c} 1 \\ \vdots \\ b^- \\ b \end{array} \begin{pmatrix} 1 & \cdots & a^- & a \\ [1] & \cdots & [a^-] & [a] \\ \vdots & & \vdots & \vdots \\ [b^-] & \cdots & [a^- b^-] & [a b^-] \\ [b] & \cdots & [a^- b] & [a b] \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_{g^-} \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \quad (4)$$

can be extended in the same way as in equations (2) and (3) with rows up to monomial ab and any table term $[tu]$ set to zero whenever $tu > ab$. This extension corresponds to the product of $P_{T+U} = \sum_{\tau \in (T+U)} [\tau] \frac{M}{\tau}$, where $M = x_1^{D_1} \cdots x_n^{D_n} = \text{LCM}(T+U)$, and $C_g = g + \sum_{t < g} \gamma_t t$, with $g \leq a$, modulo $B = (x_1^{D_1+1}, \dots, x_n^{D_n+1})$. In that case, equation (4) is equivalent to asking that the coefficients of monomials $\frac{M}{u}$ of $F_g = P_{T+U} C_g \bmod B$ are all zero for $u \leq b$, i.e. $\text{LM}(F_g) < \frac{M}{b}$. Thus, we are aiming for a Padé approximant of P_{T+U} of the form $\frac{F_g}{C_g}$ with $\text{LM}(F_g) < \frac{M}{b}$ and $\text{LM}(C_g) = g$, see also Fitzpatrick and Flynn (1992).

The SCALAR-FGLM algorithm requires to compute the kernel of $H_{\mathcal{T}_{\leq b}, \mathcal{T}_{\leq a}}$ as relations $[C_g] = 0$ such that $[t C_g] = 0$, for any $t \in \mathcal{T}$ such that $t g \in T$. Denoting s the greatest monomial such that $s g \leq a$, this comes down to requiring that C_g represents a kernel vector of the matrix $H_{\mathcal{T}_{\leq b} + \mathcal{T}_{\leq s}, \mathcal{T}_{\leq g}}$. As the set $\mathcal{T}_{\leq s} + \mathcal{T}_{\leq b}$ is stable by division, some monomials are missing compared to the set $\mathcal{T}_{\leq b s}$. Then, we can extend the matrix $H_{\mathcal{T}_{\leq b} + \mathcal{T}_{\leq s}, \mathcal{T}_{\leq g}}$ by adding all the rows with label in $\mathcal{T}_{\leq a} + \mathcal{T}_{\leq b}$ except those with monomials divisible by a monomial in $\mathcal{T}_{\leq b s} \setminus (\mathcal{T}_{\leq b} + \mathcal{T}_{\leq s})$. Doing so, the extended matrix-vector product does not encode the polynomial product $F_g = P_{T+U} C_g \bmod B$ any more, as some monomials are missing. Specifically, monomials $\frac{M}{m}$ where m is a multiple of a monomial in $\mathcal{T}_{\leq b s} \setminus (\mathcal{T}_{\leq b} + \mathcal{T}_{\leq s})$.

Therefore, it comes down to finding $|\cdot$ -minimal pairs $R_g = [F_g, C_g] = [P_{\mathcal{T}_{\leq b}, \mathcal{T}_{\leq a}} C_g \bmod B, C_g]$ with $g = \text{LM}(C_g) \leq a$ and a condition on F_g so that C_g represents a vector in the kernel of $H_{U_s, \mathcal{T}_{\leq g}}$ with U_s as large as possible. It is clear that this matrix should only have table terms $[\tau]$ that also appear in $H_{\mathcal{T}_{\leq b}, \mathcal{T}_{\leq a}}$, i.e. $\tau \in (T+U)$ and that s should be the greatest monomial as such. Hence, we have the following proposition.

Proposition 13. *Let $T = \mathcal{T}_{\leq a}$ and $U = \mathcal{T}_{\leq b}$ be finite sets of monomial terms in $\mathbb{K}[\mathbf{x}]$, let $M = \text{LCM}(T+U) = x_1^{D_1} \cdots x_n^{D_n}$ and $B = (x_1^{D_1+1}, \dots, x_n^{D_n+1})$.*

Let C_g be a polynomial with support in T and with leading monomial g and let s be the greatest monomial such that $s g \leq a$.

Let \mathcal{G}_s be a minimal set of monomials generating the sets of monomials less than $b s$ but not in $\mathcal{T}_{\leq s} + \mathcal{T}_{\leq b}$, i.e. \mathcal{G}_s is a Gröbner basis of $\mathcal{T}_{\leq b s} \setminus (\mathcal{T}_{\leq s} + \mathcal{T}_{\leq b})$.

From $F_g = P_{T+U} C_g \bmod B = \sum_{\tau \in (T+U)} f_{g,\tau} \frac{M}{\tau}$, we let \tilde{F}_g be the polynomial obtained from F_g by setting to zero all the coefficients of monomials $\frac{M}{mu}$ of F_g , with $mu|M$ and $u \in \mathcal{G}_s$. That is, $\tilde{F}_g = \sum_{\tau \in (T+U)} \tilde{f}_{g,\tau} \frac{M}{\tau}$, where $\text{NormalForm}(\sum_{\tau \in (T+U)} f_{g,\tau} \tau, \mathcal{G}_s) = \sum_{\tau \in (T+U)} \tilde{f}_{g,\tau} \tau$.

Then, $[u \sigma C_g] = 0$ for all $u \in U$ and all $\sigma \in \mathcal{T}_{\leq s}$ if, and only if, $\text{LM}(\tilde{F}_g) < \frac{M}{b s}$.

Example 14. *We still consider the binomial table \mathbf{b} with $\text{DRL}(y < x)$. We let $T = U = \mathcal{T}_{\leq x^2 y}$ so*

that

$$H_{U,T} = \begin{matrix} & \begin{matrix} 1 & y & x & y^2 & xy & x^2 & y^3 & xy^2 & x^2y \end{matrix} \\ \begin{matrix} 1 \\ y \\ x \\ y^2 \\ xy \\ x^2 \\ y^3 \\ xy^2 \\ x^2y \end{matrix} & \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 2 & 1 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 1 & 3 & 0 & 0 & 3 \\ 1 & 2 & 1 & 1 & 3 & 1 & 0 & 3 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 & 1 \\ 2 & 1 & 3 & 0 & 3 & 4 & 0 & 1 & 6 \end{pmatrix} \end{matrix}$$

The computation of the kernel of this matrix yields the vectors $\begin{pmatrix} -1 \\ -1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, $\begin{pmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, $\begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$ corresponding to $[u(xy - y - 1)] = [uy(xy - y - 1)] = [ux(xy - y - 1)] = 0$ for all $u \in U$ and the vector $\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$ corresponding to $[uy^3] = 0$ for all $u \in U$.

For $g = xy$, we thus have $s = x$ and the extended matrix-vector product

$$\begin{matrix} & \begin{matrix} 1 & y & x & y^2 & xy \end{matrix} \\ \begin{matrix} 1 \\ y \\ x \\ y^2 \\ xy \\ x^2 \\ y^3 \\ xy^2 \\ x^2y \\ x^3 \\ y^4 \\ xy^3 \\ x^2y^2 \\ x^3y \\ y^5 \\ xy^4 \\ x^2y^3 \\ x^3y^2 \\ y^6 \\ xy^5 \\ x^2y^4 \\ x^3y^3 \end{matrix} & \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 1 \\ 1 & 2 & 1 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 2 & 1 & 3 & 0 & 3 \\ 1 & 3 & 1 & 3 & 4 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 3 & 0 & 1 \\ 3 & 3 & 4 & 1 & 6 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 3 & 1 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -4 \\ 0 \\ 0 \\ 0 \\ -1 \end{pmatrix}.$$

We start with two sets of terms $T = \mathcal{T}_{\leq a}$ and $U = \mathcal{T}_{\leq b}$ so that $M = x_1^{D_1} \cdots x_n^{D_n} = \text{LCM}(T + U)$. We initialize $B = (B_1, \dots, B_n) = (x_1^{D_1+1}, \dots, x_n^{D_n+1})$, $R_{B_1} = [B_1, 0], \dots, R_{B_n} = [B_n, 0]$ and $R_1 = [P_{T+U}, 1]$.

For any $R_g = [F_g, C_g] = [P_{T+U} C_g \bmod B, C_g]$, by Proposition 13, C_g is a valid relation if $g \in T$ and $\text{LM}(\tilde{F}_g) < \frac{M}{b \cdot s}$ with $s = \max\{\sigma, \sigma g \leq a\}$. To go along with the fact that the BMS algorithm always returns a relation C_g with $g = \text{LM}(C_g)$ a pure power of a variable, if $g \notin T$, then C_g will automatically be considered a valid relation as well.

From a failing relation C_m , we get that m is in the staircase of the Gröbner basis of relations. Thus, each time a built relation is not valid, we update the staircase of the ideal of relations. At each step, we know the staircase S and equivalently the set $\mathcal{H} = \min\{h \in \mathcal{T} \setminus S\}$ which are the leading terms of the candidate relations.

For $h \in \mathcal{H}$, we now want to build R_h with the least $\text{LM}(F_h)$.

Instruction 15. Pick a failing pair $R_m = [F_m, C_m]$ with $h = qm$.

1. if there exists another failing pair $R_{m'} = [F_{m'}, C_{m'}]$ such that $\text{LM}(F_{m'}) = q \text{LM}(F_m)$, then compute R_h as the normal form of $R_{m'}$ wrt. the list $[R_m, R_{B_1}, \dots, R_{B_n}, R_{t_1}, \dots, R_{t_n}]$ where C_{t_1}, \dots, C_{t_r} are failing relations and $\text{LM}(F_{t_1}) > \dots > \text{LM}(F_{t_r})$. To do so, compute first $Q_m, Q_{B_1}, \dots, Q_{B_n}, Q_{t_1}, \dots, Q_{t_r}$ the quotients of the division of $F_{m'}$ by the list of polynomials $[F_m, B_1, \dots, B_n, F_{t_1}, \dots, F_{t_r}]$ and then return $R_h = R_{m'} - Q_m R_m - Q_{B_1} R_{B_1} - \dots - Q_{B_n} R_{B_n} - Q_{t_1} R_{t_1} - \dots - Q_{t_r} R_{t_r}$.
2. otherwise, compute R_h as the normal form of $q R_m$ wrt. to the list $[R_{B_1}, \dots, R_{B_n}, R_{t_1}, \dots, R_{t_n}]$.

Remark 16. If $q \text{LM}(F_m)$ is in the ideal spanned by B , then case 2 of Instruction 15 is equivalent to computing the normal form of $[q \text{LM}(F_m), 0]$ wrt. $[R_m, R_{B_1}, \dots, R_{B_n}, R_{t_1}, \dots, R_{t_r}]$. In fact, at the start, $R_1 = [P_{T+U}, 1]$ fails when shifted by a monomial $s = x_1^{i_1} \cdots x_n^{i_n}$ and we have to make new pairs $R_{x_1^{i_1+1}}, \dots, R_{x_n^{i_n+1}}$. Since $\text{LM}(P_{T+U}) = \frac{M}{s}$, then these pairs can be computed as the normal forms of $[x_k^{i_k+1} \frac{M}{s}, 0] = [B_k, 0] M'$, with $M' \in \mathcal{T}$, wrt. the ordered list $[R_1, R_{B_1}, \dots, R_{B_n}]$. In dimension 1, this comes down to reducing $[x_1^{D_1+1}, 0] = [B_1, 0] = R_{B_1}$ wrt. $[R_1, R_{B_1}]$, and thus R_1 only. This is indeed the first step of the extended Euclidean algorithm called on B_1 and F_1 as described in Section 3.1.

Example 17 (See Berthomieu and Faugère (2018)). Let $\mathbf{b} = \left(\binom{i}{j} \right)_{(i,j) \in \mathbb{N}^2}$ be the binomial table, $<$ be the $\text{DRL}(y < x)$ monomial ordering and $T = \mathcal{T}_{\leq x^3}$ and $U = \{1\}$ be sets of terms. We have $a = x^3$, $b = 1$, $T = T + U$ and $M = \text{LCM}(T) = x^3 y^3$ so that $P_T = x^3 y^3 + x^2 y^3 + x^2 y^2 + x y^3 + 2 x y^2 + y^3$, $R_1 = [P_T, 1] = [F_1, C_1]$ and $R_{B_1} = [x^4, 0]$, $R_{B_2} = [y^4, 0]$.

- $m = 1$, thus $s = x^3$ and as $\text{LM}(F_1) = \text{LM}(P_T) = x^3 y^3 = \frac{M}{1}$, then the relation C_1 fails when shifting by 1 so that 1 is in the staircase. Thus $\mathcal{H} = \{y, x\}$. We create R_y by computing the normal form of $[y \text{LM}(F_1), 0] = [x^3 y^4, 0]$ wrt. $[R_1, R_{B_1}, R_{B_2}]$ and get $R_y = [F_y, C_y] = [x^2 y^3 + 2 x y^3, y]$. Likewise $R_x = [F_x, C_x] = [x^3 y^2 + x^2 y^2 - 2 x y^2 - y^3, x - 1]$.
- $m = y$, thus $s = x^2$ and as $\text{LM}(F_y) = x^2 y^3 = \frac{M}{x}$, then the relation C_y fails when shifting by x so that y is in the staircase. Thus $\mathcal{H} = \{x, y^2\}$. We create
 - $R_{y^2} = [0, y^2]$ by computing the normal form of $[y \text{LM}(F_y), 0] = [x^2 y^4, 0]$ wrt. $[R_y, R_{B_1}, R_{B_2}, R_1, R_x]$.
- $m = x$, thus $s = x^2$ and as $\text{LM}(F_x) = x^3 y^2 = \frac{M}{y}$, then the relation C_x fails when shifting by y so that x is in the staircase. Thus $\mathcal{H} = \{y^2, x y, x^2\}$. We create

- $R_{xy} = [-x^2 y^2 - 3xy^3 - 2xy^2 - y^3, xy - y - 1]$ by computing the normal form of R_1 wrt. $[R_y, R_{B_1}, R_{B_2}, R_x]$;
- $R_{x^2} = [-3x^2 y^2 - xy^3 + 2xy^2 + y^3, x^2 - 2x + 1]$ by computing the normal form of $[x \text{ LM}(F_x), 0] = [x^4 y^2, 0]$ wrt. $[R_x, R_{B_1}, R_{B_2}, R_1, R_y]$.
- $m = y^2$, thus $s = x$ and as $F_{y^2} = 0$, then the relation is necessarily valid.
- $m = xy$, thus $s = x$ and as $\text{LM}(F_{xy}) = x^2 y^2 = \frac{M}{xy}$, then the relation C_{xy} is valid.
- $m = x^2$, thus $s = x$ and, likewise, as $\text{LM}(F_{x^2}) = x^2 y^2 = \frac{M}{xy}$, then the relation C_{x^2} is valid.

We return $C_{y^2} = y^2$, $C_{xy} = xy - y - 1$ and $C_{x^2} = x^2 - 2x + 1$.

The algorithm is the following Algorithm 4.18. It uses functions

NormalForm($R_m, [R_{B_1}, \dots, R_{B_n}, R_{t_1}, \dots, R_{t_r}]$), for computing the normal form of $[F_m, C_m]$ wrt. to the list $R_{B_1}, \dots, R_{B_n}, R_{t_1}, \dots, R_{t_r}$ with $\text{LM}(F_{t_1}) > \dots > \text{LM}(F_{t_r})$;

Stabilize(S), for computing the true staircase containing S , i.e. all the divisors of terms in S ;

Border(S), for computing the least terms for $|$ outside of S .

Algorithm 4.18: POLYNOMIAL SCALAR-FGLM

Input: A table $w = (w_i)_{i \in \mathbb{N}^n}$ with coefficients in \mathbb{K} , a monomial ordering $<$ and two monomials a and b .

Output: A set G of relations generating the ideal of relations.

$T := \{t \in \mathcal{T}, t \leq a\}$, $U := \{u \in \mathcal{T}, u \leq b\}$.

$M := \text{LCM}(T) \text{LCM}(U)$.

For i **from** 1 **to** n **do** $R_{B_i} := [x_i^{1+\deg_{x_i} M}, 0]$. // pairs on the edge

$P := \sum_{\tau \in (T+U)} [\tau] \frac{M}{\tau}$. // the mirror of the truncated generating series

$R := \{[P, 1]\}$. // set of pairs $[F_m, C_m] = [P \cdot C_m \bmod B, C_m]$ to be tested

$R' := \emptyset$. // set of failing pairs

$G := \emptyset, S := \emptyset$. // the future Gröbner basis and staircase

While $R \neq \emptyset$ **do**

$R_m = [F_m, C_m] :=$ first element of R and remove it from R .

If $m \notin T$ or $\text{LM}(\tilde{F}_m)$ is as in Proposition 13 **then** // good relation

$G := G \cup \{C_m\}$.

Else // bad relation

$R' := R' \cup \{R_m\}$.

For all $r \in R$ **do** // reduce next pairs with it

$r := \text{NormalForm}(r, [R_{B_1}, \dots, R_{B_r}, R_m])$.

$S := \text{Stabilize}(S \cup \{m\})$.

$H := \text{Border}(S)$.

For all $h \in H$ **do** // compute new pairs

If there is no relation $C_h \in G$ or no pair $R_h \in R$ **then**

Make a new pair $R_h = [F_h, C_h]$ following Instruction 15 and add it to R .

Return G .

Remark 19. Like the BMS algorithm, this algorithm creates new potential relations by making polynomial combinations of failing relations. As a consequence, at each step of the main loop, the potential relations, i.e. elements of R , are not necessarily interreduced. Either we can interreduce the final Gröbner basis before returning it at the last line of the algorithm, or when C_g is added to the set G we can update all the current relations by removing multiples of $[F_g, C_g]$ and likewise, reduce by $[F_g, C_g]$, any subsequent pair $[F_m, C_m]$.

Example 20 (See Berthomieu and Faugère (2018)). We give the trace of the POLYNOMIAL SCALAR-FGLM algorithm with the slight modification above called on the table $\mathbf{w} = ((2i + 1) + (2j - 1)(-1)^{i+j})_{(i,j) \in \mathbb{N}^2}$, the stopping monomials 1 and y^5 and the monomial ordering $\text{DRL}(y < x)$.

We set $T := \mathcal{T}_{\leq y^5}$, $U := \{1\}$, $M := x^4 y^5$ and $P = 4x^3 y^5 + 4x^4 y^3 + 4x^3 y^4 + 4x^2 y^5 - 4x^4 y^2 + 4x^2 y^4 + 8xy^5 + 8x^4 y + 8x^3 y^2 + 8x^2 y^3 + 8xy^4 + 8y^5 - 8x^4$, $R_{B_1} := [x^5, 0]$, $R_{B_2} := [y^6, 0]$, $R = [[P, 1]]$.

Pair $R_1 = [F_1, C_1] = [P, 1]$, $R := \emptyset$ and since $1 \in T$ but $\text{LM}(F_1) = x^3 y^5 \geq \frac{M}{s} = x^4$, then

- $R' := \{R_1\}$, $S := \{1, x\}$ and $H := \{y, x^2\}$.
- We make new pairs added to R :
 - $R_y = [F_y, C_y] := \text{NormalForm}(y \text{LM}(F_1), 0), [R_1, R_{B_1}, R_{B_2}]$ which can be normalized into $R_y = [4x^4 y^4 - \dots, y - 1]$;
 - $R_{x^2} = [F_{x^2}, C_{x^2}] := \text{NormalForm}(x^2 \text{LM}(F_1), 0), [R_1, R_{B_1}, R_{B_2}]$ which can be normalized into $R_{x^2} = [4x^4 y^3 - \dots, x^2 - x - 1]$.

Pair $R_y = [F_y, C_y]$, $R := \{R_{x^2}\}$ and since $y \in T$ but $\text{LM}(F_y) = x^4 y^4 \geq \frac{M}{s} = x^4 y$, then

- $R' := \{R_1, R_y\}$, $S := \{1, y, x\}$ and $H := \{y^2, xy, x^2\}$.
- We make new pairs added to R :
 - As $y \text{LM}(F_y) = x^4 y^5 \notin \langle x^5, y^6 \rangle$ and $\text{LM}(F_1) \neq y \text{LM}(F_y)$, we can only set $R_{y^2} = [F_{y^2}, C_{y^2}] := \text{NormalForm}(y R_y, [R_{B_1}, R_{B_2}, R_1, R_y])$ which can be normalized into $R_{y^2} = [-4x^4 y^3 - \dots, y^2 - x + 2y - 1]$;
 - $R_{xy} = [F_{xy}, C_{xy}] := \text{NormalForm}(x \text{LM}(F_y), 0), [R_y, R_{B_1}, R_{B_2}, R_1]$ which can be normalized into $R_{xy} = [4x^4 y^2 - \dots, xy - x + y - 1]$.
 - Nothing is done for x^2 since R_{x^2} already exists.

Pair $R_{y^2} = [F_{y^2}, C_{y^2}]$, $R := \{R_{xy}, R_{x^2}\}$ and since $y^2 \in T$ but $\text{LM}(F_{y^2}) = x^4 y^3 \geq \frac{M}{s} = x^4 y^2$, then

- As $\text{LM}(F_{x^2}) \geq \text{LM}(F_{y^2})$, we reduce it and obtain $R_{x^2} := [-8x^2 y^4 - \dots, x^2 + y^2 - 2x + 2y - 2]$.
- $R' := \{R_1, R_y, R_{y^2}\}$, $S := \{1, y, x, y^2\}$ and $H := \{xy, x^2, y^3\}$.
- We make new pairs added to R :
 - R_{xy} and R_{x^2} already exist so we do nothing for them.
 - Since $\text{LM}(F_y) = y \text{LM}(F_{y^2})$, we can set $R_{y^3} = [F_{y^3}, C_{y^3}] := \text{NormalForm}(R_y, [R_{y^2}, R_{B_1}, R_{B_2}, R_y, R_1])$ which can be normalized into $R_{y^3} = [4x^3 y^4 - \dots, y^3 - xy + y^2 + x - 2y]$.

Pair $R_{xy} = [F_{xy}, C_{xy}]$, $R := \{R_{x^2}, R_{y^3}\}$ and since $xy \in T$ and $\text{LM}(F_{xy}) = x^4 y^2 < \frac{M}{s} = x^2 y^5$, then

- $G := \{xy - x + y - 1\}$.

- As $C_{y^3} = y^3 - xy + y^2 + x - 2y$ has a term in xy , we update $R_{y^3} := R_{y^3} + R_{xy} = [4x^3y^4 - \dots, y^3 + y^2 - y - 1]$.

Pair $R_{x^2} = [F_{x^2}, C_{x^2}]$, $R := \{R_{y^3}\}$ and since $x^2 \in T$ and $\text{LM}(F_{x^2}) = x^2y^4 < \frac{M}{s} = x^2y^5$, then

- $G := \{xy - x + y - 1, x^2 + y^2 - 2x + 2y - 2\}$.

Pair $R_{y^3} = [F_{y^3}, C_{y^3}]$, $R := \emptyset$ and since $y^3 \in T$ and $\text{LM}(F_{y^3}) = x^3y^4 < \frac{M}{s} = x^4y^3$, then

- $G := \{xy - x + y - 1, x^2 + y^2 - 2x + 2y - 2, y^3 + y^2 - y - 1\}$.

We return G .

Theorem 21. Let a table \mathbf{w} , a monomial ordering $<$ and two monomials a and b be the input of the POLYNOMIAL SCALAR-FGLM algorithm. Let us assume that the Gröbner basis \mathcal{G} of the ideal of relations of \mathbf{w} for $<$ and its staircase S satisfy $a \geq \max(S \cup \text{LM}(\mathcal{G}))$ and for all $g \leq a$, $s = \max\{\sigma \in \mathcal{T}, \sigma g \leq a\}$, we have $\max(S) \leq s$.

Then, the POLYNOMIAL SCALAR-FGLM algorithm terminates and computes a Gröbner basis of the ideal of relations of \mathbf{w} for $<$ in $O(\#S(\#S + \#\mathcal{G})\#(\mathcal{T}_{\leq a} + \mathcal{T}_{\leq b}))$ operations in the base field.

Proof. The proof is mainly based on the termination and validity of the BMS algorithm. For any monomial $m \in \mathcal{T}_{\leq a}$, we denote by C_m^* the last (and therefore one with the largest fail) relation made by the BMS algorithm starting with m , if there is any.

Starting with $R_1 = [F_1, C_1] = [P_{\mathcal{T}_{\leq a} + \mathcal{T}_{\leq b}}, 1]$, $\text{LM}(F_1)$ yields exactly the fail of relation $C_1 = C_1^*$ so that, as in the BMS algorithm, we know the leading monomials of the potential next relations.

Let us assume now that for any monomial $\mu < h$, the pair $R_\mu = [F_\mu, C_\mu]$ made by the POLYNOMIAL SCALAR-FGLM algorithm is equivalent to C_μ^* , that is either both C_μ and C_μ^* fail when shifted by exactly the same monomial or they both succeed on $\mathcal{T}_{\leq a} + \mathcal{T}_{\leq b}$.

Since C_μ and C_μ^* are equivalent, the current discovered staircase by the BMS and the POLYNOMIAL SCALAR-FGLM algorithms are the same. Thus either h is a leading monomial of a relation to be built by both algorithms or it is not. Without loss of generality, we can assume it is. There exists a monomial m such that $m|h$ and $R_m = [F_m, C_m]$ and C_m^* have been made. In the BMS algorithm, the relation C_h^* is obtained as $\frac{h}{m}C_m^* - \sum_{\mu < h} q_\mu^* C_\mu^*$ while in the POLYNOMIAL SCALAR-FGLM algorithm, C_h is made as $\frac{h}{m}C_m - \sum_{\mu < h} q_\mu C_\mu$. In each computation, q_μ^* and q_μ are chosen so that C_m^* and C_m have the largest fail (or equivalently \tilde{F}_m has the least leading monomial), hence C_m^* and C_m are equivalent. For $h \in S$, the potential relation C_h made by the algorithm must fail when shifted by a monomial in S . Thus, there exist σ_1, σ_2 such that $\sigma_1 \sigma_2 \in S$, $\sigma_1 h \leq a$, $\sigma_2 \leq b$ and the column labeled with $\sigma_1 h$ of the matrix $H_{\mathcal{T}_{\leq b}, \mathcal{T}_{\leq a}}$ is independent from the previous ones. For $g \in \text{LM}(\mathcal{G})$, by Section 3.2, the relation C_g has been tested shifted by all the monomials in $\mathcal{T}_{\leq s} + \mathcal{T}_{\leq b}$, with $s = \max\{\sigma \in \mathcal{T}, \sigma g \leq a\}$. The theorem hypothesis is exactly that the full staircase is included in the set of tested shifts, hence we can ensure that C_g corresponds to a kernel vector of $H_{S, S \cup \{g\}}$ with the last coordinate equal to 1.

Concerning the complexity of the algorithm. Since $\mathcal{T}_{\leq a}$ and $\mathcal{T}_{\leq b}$ are stable by division, so is $\mathcal{T}_{\leq a} + \mathcal{T}_{\leq b}$. Let us recall that the support of $P_{\mathcal{T}_{\leq a} + \mathcal{T}_{\leq b}}$ is $\{\frac{M}{\tau}, \tau \in (\mathcal{T}_{\leq a} + \mathcal{T}_{\leq b})\}$, $M = \text{lcm}(\mathcal{T}_{\leq a} + \mathcal{T}_{\leq b})$. Since each F_m satisfies $F_m = P_{\mathcal{T}_{\leq a} + \mathcal{T}_{\leq b}} C_m \text{ mod } B$, then the monomials in the support of F_m are multiples of the monomials in the support of $P_{\mathcal{T}_{\leq a} + \mathcal{T}_{\leq b}}$ and thus are included in the support of $P_{\mathcal{T}_{\leq a} + \mathcal{T}_{\leq b}}$. Each pair $R_m = [F_m, C_m]$ for $m \in S \cup \text{LM}(\mathcal{G})$ must be reduced by all the previous ones lying in the staircase in at most $\#S \#(\mathcal{T}_{\leq a} + \mathcal{T}_{\leq b})$ operations. Reducing the relations to obtain a minimal Gröbner basis can be done in $O(\#S \#\mathcal{G}\#(\mathcal{T}_{\leq a} + \mathcal{T}_{\leq b}))$ operations, hence this part is not the bottleneck of the algorithm. \square

5. An adaptive variant

In general, the BMS, the SCALAR-FGLM and thus the POLYNOMIAL SCALAR-FGLM algorithms are efficient when the staircase of the Gröbner basis of the ideal of relations is closer to a simplex than to a line. Indeed, for the DRL($x_n < \dots < x_1$) ordering, whether the Gröbner basis is $\{x_n, \dots, x_2, x_1^d\}$ or all the monomials of degree d : $\{x_n^d, x_{n-1} x_n^{d-1}, \dots, x_1 x_n^{d-1}, \dots, x_1^d\}$, the BMS algorithm requires to visit all the monomials up to x_1^{2d-1} making the algorithm costly in the former case compared to the size of the staircase.

The ADAPTIVE SCALAR-FGLM algorithm Berthomieu et al. (2015) was designed to take into account the shape of the Gröbner basis gradually as it is discovered. The idea is to start from S , the empty set, and a list of monomial to visit in increasing order. At step m , the first monomial of that list, if $H_{S \cup \{m\}, S \cup \{m\}}$ has a greater rank than $H_{S, S}$, then m is added to S and removed from the list. Otherwise, a relation C_m is found and any multiple of m is removed from the list. For instance, in the line staircase with Gröbner basis $\{x_n, \dots, x_2, x_1^d\}$, it computes the rank of matrices $H_{\{1\}, \{1\}}, H_{\{1, x_n\}, \{1, x_n\}}, \dots, H_{\{1, x_2\}, \{1, x_2\}}, H_{\{1, x_1\}, \{1, x_1\}}, H_{\{1, x_1, x_1^2\}, \{1, x_1, x_1^2\}}, \dots, H_{\{1, x_1, \dots, x_1^d\}, \{1, x_1, \dots, x_1^d\}}$. It thus requires merely $2(n + d) - 1$ table terms instead of $\binom{n+2d-1}{n}$.

The main issue with the POLYNOMIAL SCALAR-FGLM algorithm in this context is that it is based on polynomials from matrices with columns set $\mathcal{T}_{\leq a}$ and rows set $\mathcal{T}_{\leq b}$.

5.1. A naive approach

The idea of the adaptive variant of the POLYNOMIAL SCALAR-FGLM algorithm is to replace the linear algebra arithmetic with matrices $H_{S \cup \{m\}, S \cup \{m\}}$ by polynomial arithmetic like in Sections 2.4 and 3.2. At each step, we have the polynomial P_{2S} , a monomial ideal B_{2S} , determined as in Section 3.2 and pairs $R_{2S, t} = [F_{2S, t}, C_t] = [P_{2S} C_t \bmod B_{2S}, C_t]$. At the next step, we compute $P_{2(S \cup \{m\})}$ from P_{2S} by shifting it by $\frac{\text{LCM}(S \cup \{m\})^2}{\text{LCM}(S)^2}$ and adding the missing terms, update B_{2S} into $B_{2(S \cup \{m\})}$ and likewise update each $R_{2(S \cup \{m\}), t}$ by shifting $F_{2S, t}$ and adding the missing terms to make $F_{2(S \cup \{m\}), t}$. Then, $R_{2(S \cup \{m\}), m}$ is initialized as $m R_{2(S \cup \{m\}), 1}$ and then reduced, as in Section 4, by $R_{2(S \cup \{m\}), B_1}, \dots, R_{2(S \cup \{m\}), B_n}$, where $B_1, \dots, B_n \in B_{2S}$.

When a relation C_g is found, any multiple of g is removed from the set of potential monomials to add to S . Moreover, we can further reduce a future relation $R_{2(S \cup \{m\}), m} = [F_{2(S \cup \{m\}), m}, C_m]$ with any pair $[\frac{M}{\mu g}, 0]$, $m \geq \mu g$, like we reduce it with $R_{2(S \cup \{m\}), B_1}, \dots, R_{2(S \cup \{m\}), B_n}$.

Although this naive approach is based on polynomial arithmetic, it does not use polynomial division like the POLYNOMIAL SCALAR-FGLM does.

5.2. A division-based adaptive variant

In Berthomieu and Faugère (2018), we did not go further in the design of an adaptive variant of the POLYNOMIAL SCALAR-FGLM. In particular, that version could not initialize a pair $R_{2(S \cup \{m\}), m}$ as the quotient of two pairs of polynomials. In the following part of this section, we show how to initialize a new pair as the quotient of two previously computed pairs of polynomials.

The algorithm uses some new procedures that are not needed in POLYNOMIAL SCALAR-FGLM. First NormalFormRightSide behaves like NormalForm except the normal form algorithm is applied between the right sides of all the pairs, and then passed on the left sides. By reducing pairs with pairs $R_{2S', t}$, $t \in G$, this allows us to ensure that the support of C_m is in $S' = S \cup \{m\}$.

Procedure NormalFormHigherPart behaves like NormalForm, except only the higher part $\tilde{F}_{2S', t}$ of a polynomial $F_{2S', t}$ is used. For $t = B_i$, $\tilde{F}_{2S', t} = F_{2S', t} = B_i$, otherwise $\tilde{F}_{2S', t}$ is obtained from $F_{2S', t}$ by only removing any monomial dividing $\frac{\text{LCM}(S')^2}{g}$ with $g \in \text{LM}(G)$. Then

Algorithm 5.22: ADAPTIVE POLYNOMIAL SCALAR-FGLM**Input:** A table $\mathbf{w} = (w_i)_{i \in \mathbb{N}^n}$ with coefficients in \mathbb{K} , a monomial ordering $<$.**Output:** A set G of relations generating the ideal of relations. $L := \{1\}$. $G := \emptyset, S := \emptyset$.

// the future Gröbner basis and staircase

 $B := \{B_1, \dots, B_n\} = \{x_1^0, \dots, x_n^0\}$. $R_{B_1} := [B_1, 0], \dots, R_{B_n} := [B_n, 0]$.**While** $L \neq \emptyset$ **do** $m :=$ first element of L and remove it from L . $S' := S \cup \{m\}$. Update B and all pairs $R_{2S,t}$ into $R_{2S',t}$ for $t \in S \cup G \cup B$. **If** $m = 1$ **then** $R_1 := [w_{0,\dots,0}, 1]$. **Else if** $m = \mu x_i^2$ with $\mu, \mu x_i \in S$ **then** $R_{2S',m} := \text{NormalFormHigherPart}(R_{2S',\mu}, [R_{2S',\mu x_i}, R_{2S',B_1}, \dots, R_{2S',B_n}])$. **Else** $R_{2S',m} := \text{NormalForm}(x_i R_{2S',\mu}, [R_{2S',B_1}, \dots, R_{2S',B_n}])$.// $m = \mu x_i$ with $\mu \in S$ $R_{2S',m} := \text{NormalFormRightSide}(R_{2S',m}, [R_{2S',g_1}, \dots, R_{2S',g_r}])$.// $g_1, \dots, g_r \in G$ $R_{2S',m} := \text{NormalForm}(R_{2S',m}, [R_{2S',B_1}, \dots, R_{2S',B_n}])$. $R_{2S',m} := \text{NormalFormHigherPart}(R_{2S',m}, [R_{2S',s_1}, \dots, R_{2S',s_q}])$.// $s_1, \dots, s_q \in S$ $R_{2S',m} = [F_{2S',m}, C_m]$. **If** $\text{LM}(\tilde{F}_{2S',m}) < \frac{\text{lcm}(S')^2}{m}$ **then**

// Relation succeeds

 Update $R_{2S',m}$ into $R_{2S,m}$. $G := G \cup \{R_{2S,m}\}$. Remove multiples of m in L . **Else**

// Relation fails

 Delete every pair $R_{2S,t}$ for $t \in S \cup G \cup B$. $S := S'$. $L := L \cup \{x_1 m, \dots, x_n m\}$, remove any multiples of $\text{LM}(G)$ and sort it by increasing order.**return** G .

NormalFormHigherPart($R_{2S',m}, [R_{2S',s_1}, \dots, R_{2S',s_q}]$) computes the normal form of $\tilde{F}_{2S',m}$ with respect to $[\tilde{F}_{2S',s_1}, \dots, \tilde{F}_{2S',s_q}]$ and the corresponding quotients Q_1, \dots, Q_q . It then returns $R_{2S',m} - Q_1 R_{2S',s_1} - \dots - Q_q R_{2S',s_q}$.

For the correctness of the algorithm, we need to prove the following lemma.

Lemma 23. *Let S be the current computed staircase and let \mathcal{G} be the current computed Gröbner basis. Let m be the least monomial not in S and not divisible by $\text{LM}(\mathcal{G})$ and let $S' = S \cup \{m\}$. Then, for all t , $\text{LM}(\tilde{F}_{2S',t}) = \frac{\text{LCM}(S')^2}{\text{LCM}(S)^2} \text{LM}(\tilde{F}_{2S,t})$ and $\text{LM}(\tilde{F}_{2S',m}) \leq \frac{\text{LCM}(S')^2}{m}$. Furthermore, this weak inequality is an equality if, and only if, C_m is not a valid relation.*

Proof. If $\mathcal{G} = \emptyset$, then the result is clear as $\tilde{F}_{2S',m} = F_{2S',m}$. Otherwise, the leading monomial of $\tilde{F}_{2S',m}$ is less than $\frac{\text{LCM}(S')}{t}$ with $t \in S$ and cannot be a divisor of $\frac{\text{LCM}(S')}{g}$ with $g \in \text{LM}(\mathcal{G})$. Hence it must be at most $\frac{\text{LCM}(S')}{m}$.

Furthermore, for $S = \{1, \dots, s_q\}$, the coefficient of $\frac{\text{LCM}(S')}{m}$ can be read as the last coefficient of this product

$$\begin{matrix} & 1 & \cdots & s_q & m \\ 1 & \left(\begin{array}{cccc} [1] & \cdots & [s_q] & [m] \\ \vdots & & \vdots & \vdots \\ [s_q] & \cdots & [s_q^2] & [m s_q] \\ [m] & \cdots & [m s_q] & [m^2] \end{array} \right) & \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_{s_q} \\ 1 \end{pmatrix} & = & \begin{pmatrix} 0 \\ \vdots \\ 0 \\ f_{2S',m} \end{pmatrix}. \end{matrix}$$

Hence, the relation is valid if, and only if, this coefficient is zero. \square

The leading monomial of the higher part of $F_{2S',m}$ tells us whether C_m is a valid relation or not.

Theorem 24. *Assuming the ADAPTIVE POLYNOMIAL SCALAR-FGLM called on table \mathbf{w} returns a Gröbner basis \mathcal{G} with staircase S , then the algorithm does not need more than $\#2(S \cup \text{LM}(\mathcal{G}))$ table queries to recover \mathcal{G} .*

Furthermore, it performs at most $O((\#S + \#\mathcal{G})^2 \#2S)$ operations to recover \mathcal{G} .

Proof. Since $\tilde{F}_{2S,t}$ is obtained from $F_{2S,t}$ by removing all the monomials dividing $\frac{\text{LCM}(S)^2}{g}$, $g \in \text{LM}(\mathcal{G})$, then $\tilde{F}_{2S,t}$ is actually the polynomial obtained from the product of the extended matrix $H_{T,S}$ and the vector representing C_t , where $T = 2S \setminus \{g\tau, g \in \text{LM}(\mathcal{G}), \tau \in \mathcal{T}\}$. Hence, updating from $\tilde{F}_{2S,t}$ to $\tilde{F}_{2S',t}$ is equivalent to updating $H_{T,S}$ to $H_{T',S'}$, with $T' = 2S' \setminus \{g\tau, g \in \text{LM}(\mathcal{G}), \tau \in \mathcal{T}\}$ in this matrix-vector product. Yet, the first nonzero coefficient of this product remains the same through this updating process.

At step m , $F_{2S',t} = P_{2S'} C_t \bmod B_{2S'}$ for any $t \in S \cup \text{LM}(\mathcal{G})$. As in the end of the step, m is either found to be a member of the final S or the final $\text{LM}(\mathcal{G})$, then $S' \subseteq (S \cup \text{LM}(\mathcal{G}))$ and thus $2S' \subseteq 2(S \cup \text{LM}(\mathcal{G}))$.

At step m , a pair $R_{2S,t} = [F_{2S,t}, C_t]$ is updated into $R_{2(S \cup \{m\},t)}$ by adding terms with support in $2(S \cup \{m\})$, so in the last step, all the polynomials have support in a set of size $\#2(S \cup \{g\})$ where $g \in \text{LM}(\mathcal{G})$.

Furthermore, for each monomial $t \in S \cup \text{LM}(\mathcal{G})$, the pair $R_{2S',t}$ is reduced by all the previous ones lying in the staircase or the Gröbner basis in at most $(\#S + \#\mathcal{G}) \#2S$ operations. \square

Example 25. We consider the following sequence $\mathbf{w} = (w_{i,j})_{(i,j) \in \mathbb{N}^2}$

$$\mathbf{w} = \begin{pmatrix} 6 & 9 & 5 & 1 & 10 & -6 & -9 & \cdots \\ 3 & 12 & 2 & 4 & 7 & -3 & -12 & \cdots \\ 6 & 9 & 5 & 1 & 10 & -6 & -9 & \cdots \\ 3 & 12 & 2 & 4 & 7 & -3 & -12 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

For a pair $R_{2S,t} = [F_{2S,t}, C_t]$, we let $\tilde{R}_{2S,t} = [\tilde{F}_{2S,t}, C_t]$.

We start with $L = \{1\}$, $G = S = \emptyset$, $B = (1, 1)$ and $R_{2S,B_1} = [1, 0]$, $R_{2S,B_2} = [1, 0]$.

1. We set $m = 1$, $S' = \{1\}$, $R_{2S',B_1} = [1, 0]$, $R_{2S',B_2} = [y, 0]$ and initialize $R_{2S',1} = [6, 1]$.
As $\tilde{F}_{2S',1} = F_{2S',1} = 1$ and $\text{LM}(\tilde{F}_{2S',1}) = 1 = \frac{\text{LCM}(S')^2}{1}$ the relation fails.
 L is updated to $\{y, x\}$.
2. We set $m = y$, $S' = \{1, y\}$, $R_{2S',B_1} = [x, 0]$, $R_{2S',B_2} = [y^3, 0]$ and $R_{2S',1} = [6y^2 + 9y + 5, 1] = \tilde{R}_{2S',1}$.
We initialize $R_{2S',y} = [9y^2 + 5y, y]$ using NormalForm on $yR_{2S',1}$ and then reduce it to $[-\frac{17}{2}y - \frac{15}{2}, y - \frac{3}{2}]$.
As $\tilde{F}_{2S',y} = F_{2S',y}$ and $\text{LM}(\tilde{F}_{2S',y}) = \frac{\text{LCM}(S')^2}{y}$ the relation fails.
 L is updated to $\{x, y^2\}$.
3. We set $m = x$, $S' = \{1, y, x\}$, $R_{2S',B_1} = [x^3, 0]$, $R_{2S',B_2} = [y^3, 0]$,
 - $R_{2S',1} = [6x^2y^2 + 9x^2y + 3xy^2 + 5x^2 + 12xy + 6y^2, 1] = \tilde{R}_{2S',1}$,
 - $R_{2S',y} = [-\frac{17}{2}x^2y + \frac{15}{2}xy^2 - \frac{15}{2}x^2 - 18xy - 9y^2, y - \frac{3}{2}] = \tilde{R}_{2S',y}$.
We initialize $R_{2S',x} = [3x^2y^2 + 12x^2y + 6xy^2, x]$ using NormalForm on $xR_{2S',1}$ and then reduce it to $[\frac{189}{17}xy^2 - \frac{155}{17}x^2 - \frac{372}{17}xy - \frac{186}{17}y^2, x + \frac{15}{17}y - \frac{31}{17}]$.
As $\tilde{F}_{2S',x} = F_{2S',x}$ and $\text{LM}(\tilde{F}_{2S',x}) = \frac{\text{LCM}(S')^2}{x}$ the relation fails.
 L is updated to $\{y^2, xy, x^2\}$.
4. We set $m = y^2$, $S' = \{1, y, x, y^2\}$, $R_{2S',B_1} = [x^3, 0]$, $R_{2S',B_2} = [y^5, 0]$,
 - $R_{2S',1} = [6x^2y^4 + \cdots, 1] = \tilde{R}_{2S',1}$,
 - $R_{2S',y} = [-\frac{17}{2}x^2y^3 + \cdots, y - \frac{3}{2}] = \tilde{R}_{2S',y}$,
 - $R_{2S',x} = [\frac{189}{17}xy^4 + \cdots, x + \frac{15}{17}y - \frac{31}{17}] = \tilde{R}_{2S',x}$.
We initialize $R_{2S',y^2} = [-\frac{106}{17}xy^4 + \cdots, y^2 - \frac{13}{17}y + \frac{16}{51}]$ using NormalFormHigherPart on $R_{2S',1}$ and $R_{2S',y}$ and then reduce it to $[\frac{1381}{189}x^2y^2 + \cdots, y^2 + \frac{106}{189}x - \frac{17}{63}y - \frac{134}{189}]$.
As $\tilde{F}_{2S',y^2} = F_{2S',y^2}$ and $\text{LM}(\tilde{F}_{2S',y^2}) = \frac{\text{LCM}(S')^2}{y^2}$ the relation fails.
 L is updated to $\{xy, x^2, y^3\}$.
5. We set $m = xy$, $S' = \{1, y, x, y^2, xy\}$, $R_{2S',B_1} = [x^3, 0]$, $R_{2S',B_2} = [y^5, 0]$,
 - $R_{2S',1} = [6x^2y^4 + \cdots, 1] = \tilde{R}_{2S',1}$,
 - $R_{2S',y} = [-\frac{17}{2}x^2y^3 + \cdots, y - \frac{3}{2}] = \tilde{R}_{2S',y}$,
 - $R_{2S',x} = [\frac{189}{17}xy^4 + \cdots, x + \frac{15}{17}y - \frac{31}{17}] = \tilde{R}_{2S',x}$,
 - $R_{2S',y^2} = [\frac{1381}{189}x^2y^2 + \cdots, y^2 + \frac{106}{189}x - \frac{17}{63}y - \frac{134}{189}] = \tilde{R}_{2S',y^2}$.

We initialize $R_{2S',xy} = [-\frac{15}{2}x^2y^4 + \dots, xy - \frac{3}{2}x]$ using NormalForm on $xR_{2S',y}$ and then reduce it to $[-15y^4 - 7x^2y - xy^2 - 14y^3 - 10x^2 - 4xy - 5y^2, xy + x - y - 1]$.

As $\tilde{F}_{2S',xy} = F_{2S',xy}$ and $\text{LM}(\tilde{F}_{2S',xy}) = y^4 < \frac{\text{LCM}(S')^2}{xy}$ the relation succeeds!

We update $R_{2S,xy} = [-9xy^4 + \dots, xy + x - y - 1]$ and put it in G .

6. We set $m = x^2$, $S' = \{1, y, x, y^2, x^2\}$, $R_{2S',B_1} = [x^5, 0]$, $R_{2S',B_2} = [y^5, 0]$

- $R_{2S',1} = [6x^4y^4 + \dots, 1]$, $\tilde{R}_{2S',1} = [6x^4y^4 + \dots, 1]$,
- $R_{2S',y} = [-\frac{17}{2}x^4y^3 - + \dots, y - \frac{3}{2}]$, $\tilde{R}_{2S',y} = [-\frac{17}{2}x^4y^3 - + \dots, y - \frac{3}{2}]$,
- $R_{2S',x} = [\frac{189}{17}x^3y^4 + \dots, x + \frac{15}{17}y - \frac{31}{17}]$, $\tilde{R}_{2S',x} = [\frac{189}{17}x^3y^4 + \dots, x + \frac{15}{17}y - \frac{31}{17}]$,
- $R_{2S',y^2} = [\frac{1381}{189}x^4y^2 + \dots, y^2 + \frac{106}{189}x - \frac{17}{63}y - \frac{134}{189}]$, $\tilde{R}_{2S',y^2} = [\frac{1381}{189}x^4y^2 + \dots, y^2 + \frac{106}{189}x - \frac{17}{63}y - \frac{134}{189}]$,
- $R_{2S',xy} = [-4x^4y^2 + \dots, xy + x - y - 1]$.

We initialize $R_{2S',x^2} = [-\frac{945}{34}x^4y^3 - x^4y^3 + \dots, x^2 + \frac{15}{17}xy - \frac{45}{34}x + \frac{15}{34}y - \frac{47}{17}]$ using NormalFormHigherPart on $\tilde{R}_{2S',1}$ and $\tilde{R}_{2S',x}$ and notice that C_{x^2} has not its support in S' .

We then reduce it to $[-12x^3y^3 + \dots, x^2 - 1]$ by computing first the reduction of the right part by $xy + x - y - 1$ and then by computing the reduction of the left parts without taking into account any monomials $\frac{\text{LCM}(S')^2}{xyt} = \frac{x^3y^3}{t}$.

As $\tilde{F}_{2S',x^2} = -x^4y^3 + \dots \neq F_{2S',x^2}$ and $\text{LM}(\tilde{F}_{2S',x^2}) = x^4y^3 < \frac{\text{LCM}(S')^2}{x^2}$ the relation succeeds!

We update $R_{2S,x^2} = [-9x^2y^3 + \dots, x^2 - 1]$ and put it in G .

7. We set $m = y^3$, $S' = \{1, y, x, y^2, y^3\}$, $R_{2S',B_1} = [x^3, 0]$, $R_{2S',B_2} = [y^7, 0]$,

- $R_{2S',1} = [6x^2y^6 + \dots, 1]$, $\tilde{R}_{2S',1} = [6x^2y^6 + \dots, 1]$,
- $R_{2S',y} = [-\frac{17}{2}x^2y^5 - + \dots, y - \frac{3}{2}]$, $\tilde{R}_{2S',y} = [-\frac{17}{2}x^2y^5 - + \dots, y - \frac{3}{2}]$,
- $R_{2S',x} = [\frac{189}{17}xy^6 + \dots, x + \frac{15}{17}y - \frac{31}{17}]$, $\tilde{R}_{2S',x} = [\frac{189}{17}xy^6 + \dots, x + \frac{15}{17}y - \frac{31}{17}]$,
- $R_{2S',y^2} = [\frac{1381}{189}x^2y^4 + \dots, y^2 + \frac{106}{189}x - \frac{17}{63}y - \frac{134}{189}]$, $\tilde{R}_{2S',y^2} = [\frac{1381}{189}x^2y^4 + \dots, y^2 + \frac{106}{189}x - \frac{17}{63}y - \frac{134}{189}]$,
- $R_{2S',xy} = [-9xy^6 + \dots, xy + x - y - 1]$,
- $R_{2S',x^2} = [-9x^2y^5 + \dots, x^2 - 1]$.

We initialize $R_{2S',y^3} = [-\frac{4204}{1381}xy^5 + \dots, y^3 + \frac{1354}{1381}y^2 - \frac{607}{1381}x - \frac{190}{1381}y - \frac{770}{1381}]$ using NormalFormHigherPart on $R_{2S',y}$ and R_{2S',y^2} .

We then reduce it to $[-\frac{4204}{1381}xy^5 - \frac{4620}{1381}y^6 - \frac{25651}{1381}x^2y^3 + \dots, y^3 + \frac{1354}{1381}y^2 - \frac{607}{1381}x - \frac{190}{1381}y - \frac{770}{1381}]$ by computing the reduction of the left parts without taking into account any monomials $\frac{\text{LCM}(S')^2}{xyt} = \frac{xy^5}{t}$, $\frac{\text{LCM}(S')^2}{x^2t} = \frac{y^6}{t}$.

As $\tilde{F}_{2S',y^3} = -\frac{25651}{1381}x^2y^3 + \dots \neq F_{2S',y^3}$ and $\text{LM}(\tilde{F}_{2S',y^3}) = x^2y^3 = \frac{\text{LCM}(S')^2}{y^3}$ the relation fails!

L is updated to $\{y^4\}$.

8. We set $m = y^4$, $S' = \{1, y, x, y^2, y^3, y^4\}$, $R_{2S',B_1} = [x^3, 0]$, $R_{2S',B_2} = [y^9, 0]$,

- $R_{2S',1} = [6x^2y^8 + \dots, 1]$, $\tilde{R}_{2S',1} = [6x^2y^8 + \dots, 1]$,
- $R_{2S',y} = [-\frac{17}{2}x^2y^7 - + \dots, y - \frac{3}{2}]$, $\tilde{R}_{2S',y} = [-\frac{17}{2}x^2y^7 - + \dots, y - \frac{3}{2}]$,
- $R_{2S',x} = [\frac{189}{17}xy^8 + \dots, x + \frac{15}{17}y - \frac{31}{17}]$, $\tilde{R}_{2S',x} = [\frac{189}{17}xy^8 + \dots, x + \frac{15}{17}y - \frac{31}{17}]$,

- $R_{2S',y^2} = [\frac{1381}{189}x^2y^6 + \dots, y^2 + \frac{106}{189}x - \frac{17}{63}y - \frac{134}{189}]$, $\tilde{R}_{2S',y^2} = [\frac{1381}{189}x^2y^6 + \dots, y^2 + \frac{106}{189}x - \frac{17}{63}y - \frac{134}{189}]$,
- $R_{2S',y^3} = [\frac{5463}{1381}xy^7 - \frac{4620}{1381}y^8 - \frac{25651}{1381}x^2y^5 + \dots, y^3 + \frac{1354}{1381}y^2 - \frac{607}{1381}x - \frac{190}{1381}y - \frac{770}{1381}]$,
 $\tilde{R}_{2S',y^3} = [-\frac{25651}{1381}x^2y^5 + \dots, y^3 + \frac{1354}{1381}y^2 - \frac{607}{1381}x - \frac{190}{1381}y - \frac{770}{1381}]$,
- $R_{2S',xy} = [-9xy^8 + \dots, xy + x - y - 1]$,
- $R_{2S',x^2} = [-9x^2y^7 + \dots, x^2 - 1]$.

We initialize R_{2S',y^4} using NormalFormHigherPart on R_{2S',y^2} and R_{2S',y^3} and then reduce it to $[-\frac{571416}{25651}xy^7 + \dots + \frac{504287}{25651}x^2y^4 + \dots, y^4 - \frac{29900}{25651}y^3 + \frac{7703}{25651}y^2 + \frac{72041}{25651}x - \frac{35598}{25651}y - \frac{26811}{25651}]$ by computing the reduction of the left parts without taking into account any monomials $\frac{\text{LCM}(S')^2}{xyt} = \frac{xy^7}{t}$, $\frac{\text{LCM}(S')^2}{x^2t} = \frac{y^8}{t}$.

As $\tilde{F}_{2S',y^4} = \frac{504287}{25651}x^2y^4 + \dots \neq F_{2S',y^3}$ and $\text{LM}(\tilde{F}_{2S',y^4}) = x^2y^4 = \frac{\text{LCM}(S')^2}{y^4}$ the relation fails! L is updated to $\{y^5\}$.

9. We set $m = y^5$, $S' = \{1, y, x, y^2, y^3, y^4, y^5\}$, $R_{2S',B_1} = [x^3, 0]$, $R_{2S',B_2} = [y^{11}, 0]$,

- $R_{2S',1} = [6x^2y^{10} + \dots, 1]$, $\tilde{R}_{2S',1} = [6x^2y^{10} + \dots, 1]$,
- $R_{2S',y} = [-\frac{17}{2}x^2y^9 - \dots, y - \frac{3}{2}]$, $\tilde{R}_{2S',y} = [-\frac{17}{2}x^2y^9 - \dots, y - \frac{3}{2}]$,
- $R_{2S',x} = [\frac{189}{17}xy^{10} + \dots, x + \frac{15}{17}y - \frac{31}{17}]$, $\tilde{R}_{2S',x} = [\frac{189}{17}xy^{10} + \dots, x + \frac{15}{17}y - \frac{31}{17}]$,
- $R_{2S',y^2} = [\frac{1381}{189}x^2y^8 + \dots, y^2 + \frac{106}{189}x - \frac{17}{63}y - \frac{134}{189}]$, $\tilde{R}_{2S',y^2} = [\frac{1381}{189}x^2y^8 + \dots, y^2 + \frac{106}{189}x - \frac{17}{63}y - \frac{134}{189}]$,
- $R_{2S',y^3} = [\frac{5463}{1381}xy^9 - \frac{4620}{1381}y^{10} - \frac{25651}{1381}x^2y^7 + \dots, y^3 + \frac{1354}{1381}y^2 - \frac{607}{1381}x - \frac{190}{1381}y - \frac{770}{1381}]$,
 $\tilde{R}_{2S',y^3} = [-\frac{25651}{1381}x^2y^7 + \dots, y^3 + \frac{1354}{1381}y^2 - \frac{607}{1381}x - \frac{190}{1381}y - \frac{770}{1381}]$,
- $R_{2S',y^4} = [-\frac{648369}{25651}xy^9 + \dots + \frac{504287}{25651}x^2y^6 + \dots, y^4 - \frac{29900}{25651}y^3 + \frac{7703}{25651}y^2 + \frac{72041}{25651}x - \frac{35598}{25651}y - \frac{26811}{25651}]$,
 $\tilde{R}_{2S',y^4} = [\frac{504287}{25651}x^2y^6 + \dots, y^4 - \frac{29900}{25651}y^3 + \frac{7703}{25651}y^2 + \frac{72041}{25651}x - \frac{35598}{25651}y - \frac{26811}{25651}]$,
- $R_{2S',xy} = [-9xy^{10} + \dots, xy + x - y - 1]$,
- $R_{2S',x^2} = [-9x^2y^9 + \dots, x^2 - 1]$.

We initialize R_{2S',y^5} using NormalFormHigherPart on R_{2S',y^3} and R_{2S',y^4} and then reduce it to $[12xy^9 + 6y^{10} + 2xy^8 + 4xy^7 + 7xy^6 - 9x^2y^4 - 3xy^5 - 5x^2y^3 - x^2y^2 - 10x^2y + 6x^2, y^5 + 1]$ by computing the reduction of the left parts without taking into account any monomials $\frac{\text{LCM}(S')^2}{xyt} = \frac{xy^9}{t}$, $\frac{\text{LCM}(S')^2}{x^2t} = \frac{y^{10}}{t}$.

As $\tilde{F}_{2S',y^5} = -9x^2y^4 + \dots \neq F_{2S',y^3}$ and $\text{LM}(\tilde{F}_{2S',y^5}) = x^2y^4 < \frac{\text{LCM}(S')^2}{y^5}$ the relation succeeds! We update $R_{2S',y^5} = [3xy^8 + \dots, y^5 + 1]$ and put it in G .

The algorithms returns G , in particular the second part of each pair: $[xy + x - y - 1, x^2 - 1, y^5 + 1]$.

Remark 26. At step y^4 , updating R_{2S,y^3} into R_{2S',y^3} makes the leading monomial of F_{2S',y^3} , $\frac{5463}{1381} \frac{\text{LCM}(S')^2}{xy}$, totally different from this of F_{2S,y^3} , $-\frac{4204}{1381} \frac{\text{LCM}(S)^2}{xy}$. However, this is not the case when considering the leading monomials of \tilde{F}_{2S',y^3} and \tilde{F}_{2S,y^3} , as $\text{LM}(\tilde{F}_{2S',y^3}) = -\frac{25651}{1381} \frac{\text{LCM}(S')^2}{y^3} = \text{LM}(\tilde{F}_{2S,y^3}) \frac{\text{LCM}(S')^2}{\text{LCM}(S)^2}$.

6. Experiments

In this section, we report on the number of arithmetic operations done by the different algorithms for computing the Gröbner basis of the ideal of relations of some table families. They are counted using naive multiplications. Three families in dimension 2 (Figure 1) and dimension 3 (Figure 2) are tested. For each of them we use the $\text{DRL}(z < y < x)$ ordering and denote by S the staircase and $\text{LM}(\mathcal{G})$ the set of the leading terms of the Gröbner basis of relations.

Rectangle tables: $\text{LM}(\mathcal{G}) = \{y^{\lfloor d/2 \rfloor}, x^d\}$ in dimension 2 and $\text{LM}(\mathcal{G}) = \{z^{\lfloor d/3 \rfloor}, y^{\lfloor d/2 \rfloor}, x^d\}$ in dimension 3.

This case is the best for the size of the Gröbner basis compared to the size of the staircase.

L-shape tables: $\text{LM}(\mathcal{G}) = \{xy, y^d, x^d\}$ in dimension 2 and $\text{LM}(\mathcal{G}) = \{yz, xz, xy, z^d, y^d, x^d\}$ in dimension 3. This case is the worst for the number of table queries compared to the sizes of the staircase and the Gröbner basis.

Simplex tables: $\text{LM}(\mathcal{G}) = \{y^d, xy^{d-1}, \dots, x^d\}$ in dimension 2 and $\text{LM}(\mathcal{G}) = \{z^d, yz^{d-1}, xz^{d-1}, \dots, y^d, xy^{d-1}, \dots, x^d\}$ in dimension 3, i.e. all the monomials of degree d . This case is the best for the number of table queries and the worst for the size of the Gröbner basis, both compared to the size of the staircase.

Let $a = \max(S \cup \text{LM}(\mathcal{G}))$. Generically, a relation C_m fails when shifted by m . From (Bras-Amorós and O’Sullivan, 2006, Prop. 10), we know that the BMS algorithm recover all the relations when called up to monomial $\max(S) \max(S \cup \text{LM}(\mathcal{G}))$. Yet, if $\max(\text{LM}(\mathcal{G})) > \max(S)$, then for $g \in \text{LM}(\mathcal{G})$, the relation C_g is not necessarily shifted by g , so we called it with a^2 . The SCALAR-FGLM algorithm was called on $U = T = \mathcal{T}_{\leq a}$. The POLYNOMIAL SCALAR-FGLM algorithm was called on $U = \{1\}, T = \mathcal{T}_{\leq a^2}$ and $U = T = \mathcal{T}_{\leq a}$ and we report the lower number of operations. The needed input set of monomials for the ARTINIAN GORENSTEIN BORDER BASES algorithm (AGBB) of Mourrain (2017) to recover the ideal of relations was higher than expected in some situations. It explains the big overhead in Figure 1 in dimension 2:

Rectangle tables: The border basis is recovered whenever the algorithm visits all the monomials of degree at most $2(d + \lfloor d/2 \rfloor + \lfloor d/3 \rfloor - 1)$.

L-shape tables: The correct border basis is not recovered. In dimension 2, we have to visit all the monomials of degree $5d - 10$ to close the staircase. This yields relations of degree higher than expected, though. For instance, with $d = 8$, we obtained $\text{LM}(\mathcal{B}) = \{xy^i, x^i y, y^{13}, x^{13}, 1 \leq i \leq 12\}$.

Simplex tables: The correct border basis is not recovered. In dimension 2, we have to visit all the monomials of degree around $5d$ or $6d$ to close the staircase. This also yields relations of degree higher than expected. For instance, with $d = 8$, we obtained $\text{LM}(\mathcal{B}) = \{x^i y^{18-i}, 0 \leq i \leq 18\}$.

The POLYNOMIAL SCALAR-FGLM algorithm performs fewer arithmetic operations than the others, for large d . More precisely, its number of operations appears to be linear in $(\#S)^2 = O(\#S(\#S + \#cG))$ in fixed dimension.

Simplex tables: While it seems the SCALAR-FGLM algorithm is the fastest in Figure 2, we can expect that it will not be the case in higher degrees, like in Figure 1. This would confirm the observed speedup in dimension 2 to dimension 3.

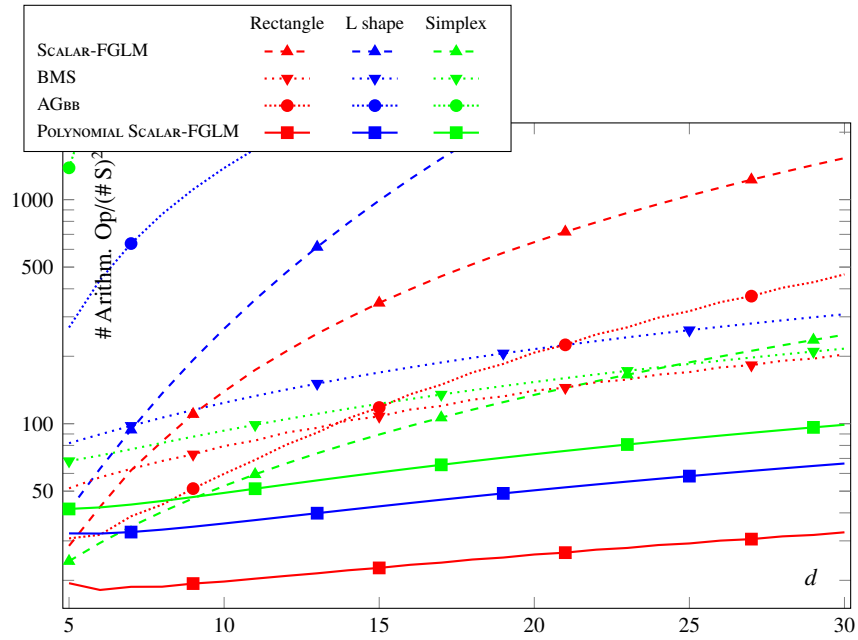


Figure 1: Number of arithmetic operations (2D)

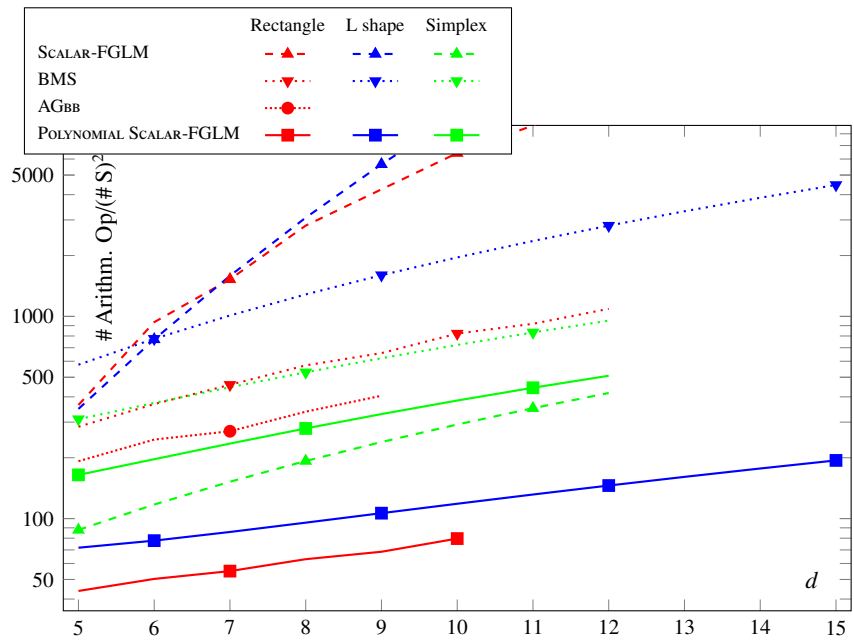


Figure 2: Number of arithmetic operations (3D)

L-shape tables: Although the obtained speedups are not negligible, the adaptive variant should allow us to perform even fewer operations. See Section 5.

Rectangle tables: While this family has the best behavior for the BMS algorithm, the POLYNOMIAL SCALAR-FGLM algorithm has an even greater speedup than in the Simplex case.

In the following Figures 3 and 4, we compare the number of operations performed by the POLYNOMIAL SCALAR-FGLM, the ADAPTIVE SCALAR-FGLM and the ADAPTIVE POLYNOMIAL SCALAR-FGLM algorithms on these three families. First, as we can expect, the ADAPTIVE POLYNOMIAL SCALAR-FGLM algorithms always performs fewer operations than the POLYNOMIAL SCALAR-FGLM algorithm. Even when giving the sharper sets of monomials $\mathcal{T}_{\leq a}$ and $\mathcal{T}_{\leq b}$ for the POLYNOMIAL SCALAR-FGLM algorithm. Furthermore, like in the non-adaptive case, the polynomial arithmetic seems to be faster than the linear algebra ones. Though, it would be interesting to compare with a less naive implementation of the ADAPTIVE SCALAR-FGLM algorithm.

That being said, in three variables, the ADAPTIVE SCALAR-FGLM algorithm is the better one for the L-shape family in small size, until the ADAPTIVE POLYNOMIAL SCALAR-FGLM becomes the better one for bigger sizes. As the crossing happens later in 3D than in 2D, it is possible that the more variables, the bigger the degrees need to be for the ADAPTIVE POLYNOMIAL SCALAR-FGLM algorithm to be better than the ADAPTIVE SCALAR-FGLM algorithm, for this family. This makes the ADAPTIVE SCALAR-FGLM algorithm still a suited algorithm in this context.

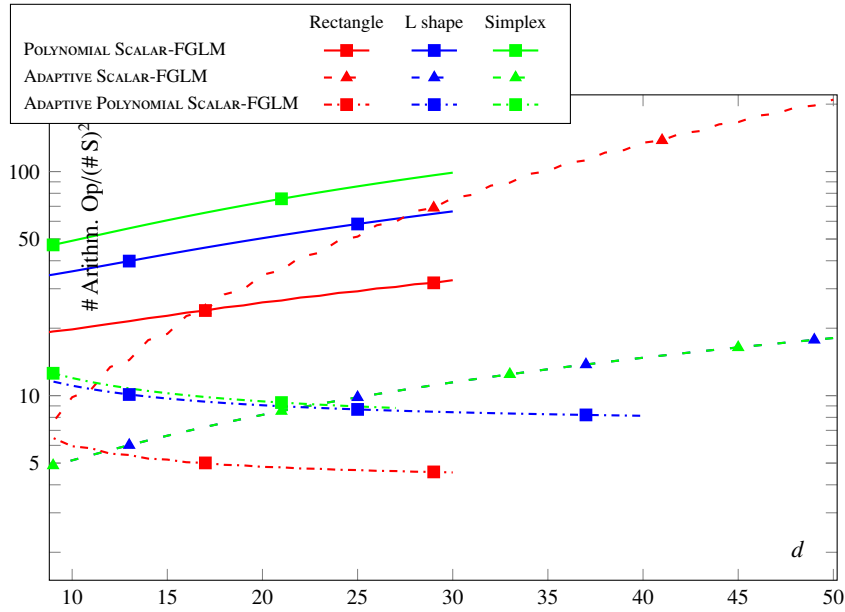


Figure 3: Number of arithmetic operations, for the adaptive variants, compared with the POLYNOMIAL SCALAR-FGLM algorithm (2D)

References

Beckermann, B., Labahn, G., 1994. A uniform approach for the fast computation of matrix-type pade approximants. SIAM J. Matrix Anal. Appl. 15 (3), 804–823.

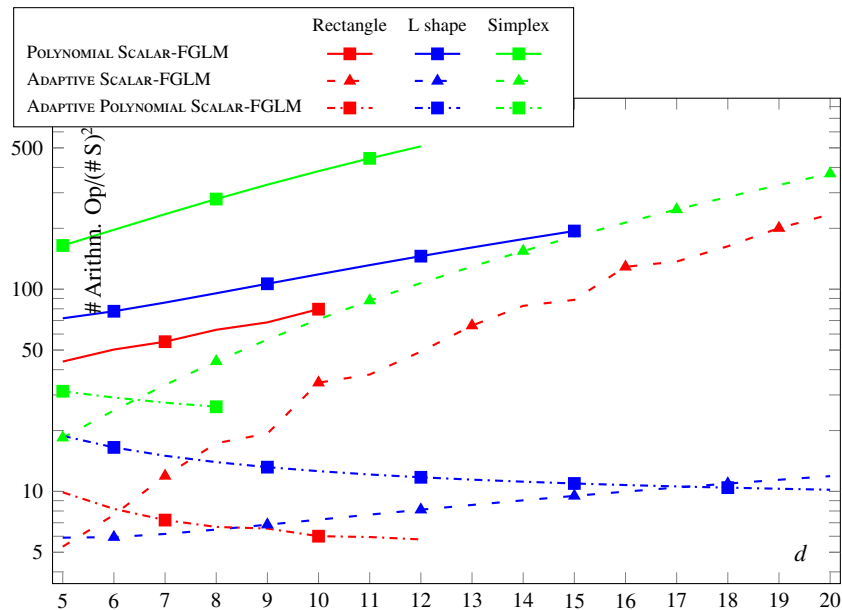


Figure 4: Number of arithmetic operations, for the adaptive variants, compared with the POLYNOMIAL SCALAR-FGLM algorithm (3D)

URL <http://dx.doi.org/10.1137/S0895479892230031>

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 gröbner bases of linear recursive multidimensional sequences. In: *Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation. ISSAC '15*. ACM, New York, NY, USA, pp. 61–68.
URL <http://doi.acm.org/10.1145/2755996.2756673>

Berthomieu, J., Boyer, B., Faugère, J.-Ch., Nov. 2017. Linear Algebra for Computing Gröbner Bases of Linear Recursive Multidimensional Sequences. *Journal of Symbolic Computation* 83 (Supplement C), 36–67, special issue on the conference ISSAC 2015: Symbolic computation and computer algebra.
URL <https://hal.inria.fr/hal-01253934>

Berthomieu, J., Faugère, J.-Ch., May 2017. In-depth comparison of the Berlekamp – Massey – Sakata and the Scalar-FGLM algorithms: the non adaptive variants, preprint.
URL <https://hal.inria.fr/hal-01516708>

Berthomieu, J., Faugère, J.-Ch., 2018. A Polynomial-Division-Based Algorithm for Computing Linear Recurrence Relations. In: *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation. ISSAC '18*. ACM, New York, NY, USA, pp. 79–86.
URL <http://doi.acm.org/10.1145/3208976.3209017>

Berthomieu, J., Faugère, J.-Ch., 2018. Experiments.
URL <http://www-polsys.lip6.fr/~berthomieu/ISSAC2018.html>

Blackburn, S. R., 1997. Fast rational interpolation, reed-solomon decoding, and the linear complexity profiles of sequences. *IEEE Transactions on Information Theory* 43 (2), 537–548.

Bose, R., Ray-Chaudhuri, D., 1960. On a class of error correcting binary group codes. *Information and Control* 3 (1), 68–79.
URL <http://www.sciencedirect.com/science/article/pii/S0019995860902874>

Bras-Amorós, M., O’Sullivan, M. E., 2006. The correction capability of the Berlekamp–Massey–Sakata algorithm with majority voting. *Applicable Algebra in Engineering, Communication and Computing* 17 (5), 315–335.
URL <http://dx.doi.org/10.1007/s00200-006-0015-8>

Brent, R. P., Gustavson, F. G., Yun, D. Y., 1980. Fast solution of Toeplitz systems of equations and computation of Padé approximants. *Journal of Algorithms* 1 (3), 259–295.

- URL <http://www.sciencedirect.com/science/article/pii/0196677480900139>
- Cantor, D. G., Kaltofen, E., 1991. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica* 28, 693–701.
- Coolley, J. W., Tukey, J. W., 1965. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation* 19 (90), 297–301.
URL <http://www.jstor.org/stable/2003354>
- Cox, D., Little, J., O’Shea, D., 2015. *Ideals, Varieties, and Algorithms*, 4th Edition. Undergraduate Texts in Mathematics. Springer, New York, an introduction to computational algebraic geometry and commutative algebra.
- Dornstetter, J., 1987. On the equivalence between Berlekamp’s and Euclid’s algorithms (corresp.). *IEEE Transactions on Information Theory* 33 (3), 428–431.
- Faugère, J.-C., Mou, C., 2011. Fast algorithm for change of ordering of zero-dimensional gröbner bases with sparse multiplication matrices. In: *Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation. ISSAC ’11*. ACM, New York, NY, USA, pp. 115–122.
URL <http://doi.acm.org/10.1145/1993886.1993908>
- Faugère, J.-Ch., Mou, C., 2017. Sparse FGLM algorithms. *Journal of Symbolic Computation* 80 (3), 538 – 569.
- Fitzpatrick, P., Flynn, J., 1992. A gröbner basis technique for padé approximation. *J. Symbolic Comput.* 13 (2), 133 – 138.
URL <http://www.sciencedirect.com/science/article/pii/S0747717108800879>
- 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.
- Hocquenghem, A., 1959. Codes correcteurs d’erreurs. *Chiffres* 2, 147 – 156.
- 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>
- Levinson, N., 1947. The Wiener RMS (Root-Mean-Square) error criterion in the filter design and prediction. *J. Math. Phys.* 25, 261–278.
- Massey, J. L., 1969. Shift-register synthesis and BCH decoding. *IEEE Trans. Inform. Theory* rr-15, 122–127.
- Mourrain, B., 2017. Fast algorithm for border bases of artinian gorenstein algebras. In: *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation. ISSAC ’17*. ACM, New York, NY, USA, pp. 333–340.
URL <http://doi.acm.org/10.1145/3087604.3087632>
- 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](http://dx.doi.org/10.1016/0890-5401(90)90039-K)
- Sakata, S., 2009. The bms algorithm. In: Sala, M., Sakata, S., Mora, T., Traverso, C., Perret, L. (Eds.), *Gröbner Bases, Coding, and Cryptography*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 143–163.
URL http://dx.doi.org/10.1007/978-3-540-93806-4_9
- Wiener, N., 1964. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. The MIT Press, Cambridge, MA.