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

Jérémy Berthomieu, Jean-Charles Faugère

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

Jérémy Berthomieu
Sorbonne Université, CNRS, INRIA,
Laboratoire d'Informatique de Paris 6, LIP6, Équipe PolSys
F-75005, Paris, France
jeremy.berthomieu@lip6.fr

Jean-Charles Faugère
Sorbonne Université, CNRS, INRIA,
Laboratoire d'Informatique de Paris 6, LIP6, Équipe PolSys
F-75005, Paris, France
jean-charles.faugere@inria.fr

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.

Finally, we give a partial solution to the transformation of this algorithm into an adaptive one.

CCS CONCEPTS

• **Comput. method.** → **Symbolic calculus algorithms**;

KEYWORDS

Gröbner bases; linear recursive sequences; Berlekamp–Massey–Sakata; extended Euclidean algorithm; Padé approximants

ACM Reference Format:
Jérémy Berthomieu and Jean-Charles Faugère. 2018. A Polynomial-Division-Based Algorithm for Computing Linear Recurrence Relations. In *ISSAC '18: 2018 ACM International Symposium on Symbolic and Algebraic Computation, July 16–19, 2018, New York, NY, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3208976.3209017

ISSAC '18, July 16–19, 2018, New York, NY, USA

ACM ISBN 978-1-4503-5550-6/18/07…$15.00
https://doi.org/10.1145/3208976.3209017

1 INTRODUCTION

The Berlekamp–Massey algorithm (BM), introduced by Berlekamp in 1968 [2] and Massey in 1969 [22] is a fundamental algorithm in Coding Theory [8, 19] 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) [24–26], can be used to compute a Gröbner basis of the zero-dimensional ideal of the relations satisfied by a sequence. 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 [15, 16] algorithm rely heavily on the efficiency of the BMS algorithm to compute the change of ordering of a Gröbner basis.

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 [20] or the Levinson–Durbin method [21, 27]. If we let $\mathsf{M}(D)$ be a cost function for multiplying two polynomials of degree D, for instance $\mathsf{M}(D) \in O(D \log D \log\log D)$ [11, 12], 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 [7, 10, 14]. More precisely, it can be done in $O(\mathsf{M}(D) \log D)$ operations.

In [3, 4], 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 $2 \le \omega \le 3$ 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 [23] for computing a border basis $\mathcal{B}$ of the ideal of relations. It extends the algorithm of [3] 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 $\sum_{i_1,\ldots,i_n\geq 0} w_{i_1,\ldots,i_n} x_1^{i_1}\cdots x_n^{i_n}$. In [17], 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 [5], 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

$$\text{LM}(F) = \text{LM}(P\,C \bmod B) \prec t_C, \tag{1}$$

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 $\frac{F}{C}$ of P modulo B with different constraints than in [17] 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 4.4) 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 [6].

Our main result is Theorem 4.7, a simplified version of which is

Theorem 1.1. *Let $\mathbf{w}$ be a sequence, $\prec$ 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 $\prec$ and its staircase S satisfy $a \geq \max(S \cup \text{LM}(\mathcal{G}))$ and for all $g \preceq a$, $s = \max_{\sigma\preceq a}\{\sigma,\ \sigma\, g \preceq a\}$, we have $\max(S) \preceq s$. Then, the Polynomial Scalar-FGLM algorithm terminates and computes a Gröbner basis of the ideal of relations of $\mathbf{w}$ for $\prec$ in $O\Big(\#S\,(\#S + \#\mathcal{G})\,\#\{\sigma, \sigma \preceq a\}\Big)$ operations in the base field.*

In applications such as the Sparse-FGLM one [16], sequence queries are costly. In [3], 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 partially transform the Adaptive Scalar-FGLM algorithm of [3] into an algorithm using polynomial arithmetic. One of the main issues to do so is that now, the monomial ideal B is not fixed: it will grow as the algorithm progresses.

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 [1] for computing multivariate Hermite–Padé approximants.

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

Definition 2.1. 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 $\mathbf{i} \in \mathbb{N}^n$, $\left[\mathbf{x}^{\mathbf{i}}\, f\right] = 0$, then we say that f is the *polynomial of the relation induced by* $\boldsymbol{\gamma} = (\gamma_{\mathbf{k}})_{\mathbf{k}\in\mathcal{K}} \in \mathbb{K}^{\#\mathcal{K}}$.

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 $\left[\mathbf{x}^{\mathbf{i}}\, f\right] = \sum_{\mathbf{k}\in\mathcal{K}} \gamma_{\mathbf{k}}\, w_{\mathbf{k}+\mathbf{i}}$.

Example 2.2. Let $\mathbf{b} = \left(\binom{i}{j}\right)_{(i,j)\in\mathbb{N}^2}$ be the sequence of the binomial coefficients. Then, the Pascal's rule is associated to $x\,y - y - 1$:

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

Definition 2.3 ([18, 24]). 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 $\prec$ on $\mathbb{K}[\mathbf{x}]$ is an order relation satisfying the following three classical properties:

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

For a monomial ordering $\prec$ on $\mathbb{K}[\mathbf{x}]$, the *leading monomial* of f, denoted $\text{LM}(f)$, is the greatest monomial in the support of f for $\prec$. 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 2.4. Let I be a nonzero ideal of $\mathbb{K}[\mathbf{x}]$ and let $\prec$ 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 $\text{LM}(g) | \text{LM}(f)$.

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 $m \in \operatorname{supp} g'$, $\text{LM}(g) \nmid m$.

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

- $\textsc{lex}(x_n \prec \cdots \prec x_1)$ **ordering** which satisfies $\mathbf{x}^{\mathbf{i}} \prec \mathbf{x}^{\mathbf{i}'}$ 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 [13, Chapter 2, Definition 3];
- $\textsc{drl}(x_n \prec \cdots \prec x_1)$ **ordering** which satisfies $\mathbf{x}^{\mathbf{i}} \prec \mathbf{x}^{\mathbf{i}'}$ if and only if $i_1 + \cdots + i_n < i'_1 + \cdots + i'_n$ or $i_1 + \cdots + i_n = i'_1 + \cdots + i'_n$ 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 [13, 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 $\prec$ such that for all $M \in \mathcal{T}$, the set $\mathcal{T}_{\preceq a} = \{m \preceq a,\ m \in \mathcal{T}\}$ is finite. Such an ordering $\prec$ makes $(\mathbb{N}^n, \prec)$ isomorphic to $(\mathbb{N}, <)$. Hence, for a monomial m, it makes sense to speak about the previous (resp. next) monomial m^- (resp. m^+) for $\prec$. 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 $\sigma \prec s$, $[\sigma\, f] = 0$ but $[s\, f] \neq 0$, see also [25, 26].

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 $(i, i') \in \{1, \ldots, m\} \times \{1, \ldots, n\}$, the coefficient $h_{i,i'}$ lying on the ith 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 $\mathbf{x}^{\mathbf{i}} \in U$ and column labeled with $\mathbf{x}^{\mathbf{i}'} \in T$ is $w_{\mathbf{i}+\mathbf{i}'}$.

Example 2.5. Let $\mathbf{w} = (w_{i,j})_{(i,j) \in \mathbb{N}^2}$ be a sequence.

(1) For $U = \{1, y, y^2, x, x\,y, x\,y^2\}$ and $T = \{1, y, x, x\,y, x^2, x^2\,y\}$,

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

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

(2) For $T = \{1, y, x, y^2, x\,y, x^2\}$,

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

is a multi-Hankel, yet not block-Hankel, matrix.

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 $2\,T = T + T$.

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 2.6. Let $\mathbf{w} = (w_{i,j})_{(i,j) \in \mathbb{N}^2}$ be a sequence and $T = \{1, y, x, y^2\}$, then $M = x\,y^2$ and $P_T = [1]\,x\,y^2 + [y]\,x\,y + [x]\,y^2 + [y^2]\,x = w_{0,0}\,x\,y^2 + w_{0,1}\,x\,y + w_{1,0}\,y^2 + w_{0,2}\,x$.

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 $\left(x_1^{D_1+1}, \ldots, x_n^{D_n+1}\right)$, 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

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

in the kernel of $H_{\{1\},\{1,\ldots,x^D\}} = \begin{array}{c} \\ 1 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & \cdots & x^D \end{array} \\ \begin{pmatrix} w_0 & \cdots & w_D \end{pmatrix} \end{array}$, that is $\gamma_1, \ldots, \gamma_{x^{d-1}}$ such that the relation $[C_{x^d}] = w_d + \sum_{k=0}^{d-1} \gamma_{x^k}\, w_k$ and its shifts, $[x\, C_{x^d}], \ldots, [x^{D-d}\, C_{x^d}]$, are all 0. Equivalently, we look for the least d such that $H_{\mathcal{T}_{\preceq x^{D-d}}, \mathcal{T}_{\preceq x^d}} \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_{x^{d-1}} \\ 1 \end{pmatrix} = 0$.

This Hankel matrix-vector product can be extended into

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

representing the product of polynomials $P_{\mathcal{T}_{\preceq x^D}} = \sum_{i=0}^{D} w_i\, x^{D-i}$ 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 $\text{LM}(F_{x^d}) \prec x^d$ with $F_{x^d} = P_{\mathcal{T}_{\preceq x^D}}\, C_{x^d} \bmod B$.

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 $R_1 = [F_1, C_1] = [P_{\mathcal{T}_{\preceq x^D}}, 1]$. Compute Q the quotient of the Euclidean division of $F_B = B$ by F_1 and then compute $R_{\text{LT}(Q)} = R_B - Q\, R_1 = [F_B - Q\, F_1, C_B - Q\, C_1] = [F_{\text{LT}(Q)}, C_{\text{LT}(Q)}]$. Repeat with R_1 and $R_{\text{LT}(Q)}$ until reaching a pair $R_{x^d} = [P_{\mathcal{T}_{\preceq x^D}}\, C_{x^d} \bmod B, C_{x^d}] = [F_{x^d}, C_{x^d}]$ with $\text{LM}(C_{x^d}) = x^d$ and $\text{LM}(F_{x^d}) \prec x^d$. 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 3.1. Let us consider the Fibonacci table $\mathbf{F} = (F_i)_{i\in\mathbb{N}}$ with $F_0 = F_1 = 1$ and assume $D = 5$ so that we have $B = x^6$, $R_B = [B, 0]$ and $R_1 = [x^5 + x^4 + 2\,x^3 + 3\,x^2 + 5\,x + 8, 1]$. As we can see $R_1 = [F_1, C_1]$ with $\text{LM}(C_1) = 1$ and $\text{LM}(F_1) = x^5 \succeq 1$.

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

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

Remark 3.2. The BM algorithm always returns a relation. If no pair $R_{x^\delta} = [F_{x^\delta}, C_{x^\delta}]$ satisfies the requirements, then it will return a pair R_{x^d} with $\text{LM}(C_{x^d}) \succ x^D$. From a matrix viewpoint, it returns an element of the kernel of the empty matrix $H_{\emptyset, \mathcal{T}_{\preceq x^d}}$.

3.2 Multidimensional extension

For a multidimensional table $\mathbf{w} = (w_\mathbf{i})_{\mathbf{i}\in\mathbb{N}^n}$, the BMS algorithm extends the BM algorithm by computing vectors in the kernel of a multi-Hankel matrix $H_{\{1\},\mathcal{T}_{\preceq a}} = \begin{array}{c} \\ 1 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & \cdots & a \end{array} \\ \begin{pmatrix} [1] & \cdots & [a] \end{pmatrix} \end{array}$ corresponding to having relations $[C_g] = 0$, with $\text{LM}(C_g) = g$ minimal for $|$ and for all t such that $t\,g \preceq a$, $[t\,C_g] = 0$ as well. This also comes down to finding the least (for the partial order $|$) monomials $g_1, \ldots, g_r \preceq a$ such that $\dim\ker H_{\mathcal{T}_{\preceq s_k}, \mathcal{T}_{\preceq g_k}} > 0$ with s_k the greatest monomial such that $s_k\, g_k \preceq a$ for all k, $1 \le k \le r$.

Remark 3.3. 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 $g \succ a$, corresponding to a vector in the kernel of the empty matrix $H_{\emptyset, \mathcal{T}_{\preceq g}}$.

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 $t\,g \in T$, then $[t\,C_g] = 0$, i.e. the vectors corresponding to $t\,C_g$ are also in the kernel of $H_{U,T}$. We now consider that both sets of terms T and U satisfy $T = \mathcal{T}_{\preceq a}$ and $U = \mathcal{T}_{\preceq b}$. This allows us to encompass both the BMS algorithm and the Scalar-FGLM algorithm.

The multi-Hankel matrix-vector product

$$\begin{array}{c} \\ 1 \\ \vdots \\ b^- \\ b \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & \cdots & a^- & a \end{array} \\ \begin{pmatrix} [1] & \cdots & [a^-] & [a] \\ \vdots & & \vdots & \vdots \\ [b^-] & \cdots & [a^-\,b^-] & [a\,b^-] \\ [b] & \cdots & [a^-\,b] & [a\,b] \end{pmatrix} \end{array} \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_{g^-} \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \quad (3)$$

can be extended in the same way as in equation (2) with rows up to monomial $a\,b$ and any table term $[t\,u]$ set to zero whenever $t\,u \succ a\,b$. 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 $C_g = g + \sum_{t\prec g} \gamma_t\, t$, with $g \preceq a$, modulo $B = (x_1^{D_1+1}, \ldots, x_n^{D_n+1})$. In that case, equation (3) 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 $u \preceq b$, i.e. $\text{LM}(F_g) \prec \frac{M}{b}$. Thus, we are aiming for a Padé approximant of P_{T+U} of the form $\frac{F_g}{C_g}$ with $\text{LM}(F_g) \prec \frac{M}{b}$ and $\text{LM}(C_g) = g$, see also [17].

The Scalar-FGLM algorithm requires to compute the kernel of $H_{\mathcal{T}_{\preceq b}, \mathcal{T}_{\preceq a}}$ as relations $[C_g] = 0$ such that $[t\,C_g] = 0$, for any $t \in \mathcal{T}$ such that $t\,g \in T$. Therefore, it comes down to finding $|$-minimal pairs $R_g = [F_g, C_g] = [P_{\mathcal{T}_{\preceq b}, \mathcal{T}_{\preceq a}}\, C_g \bmod B, C_g]$ with $g = \text{LM}(C_g) \preceq a$ and a condition on $\text{LM}(F_g)$ so that C_g represents a vector in the kernel of $H_{\mathcal{T}_{\preceq s}, \mathcal{T}_{\preceq g}}$ with s as large as possible. It is clear that this matrix should only have table terms $[\tau]$ that also appear in $H_{\mathcal{T}_{\preceq b}, \mathcal{T}_{\preceq a}}$, i.e. $\tau \in (T+U)$ and that s should be the greatest monomial as such. Hence,

$$s = \max_{\sigma\in\mathcal{T}} \Big\{\sigma,\ \{\sigma\} + \mathcal{T}_{\preceq g} \subseteq (T+U)\Big\}. \quad (4)$$

We can now deduce that

Proposition 3.4. *A pair $R_g = [F_g, C_g] = [P_{\mathcal{T}_{\preceq a}+\mathcal{T}_{\preceq b}}\, C_g \bmod B, C_g]$ corresponds to a kernel vector of the multi-Hankel matrix $H_{\mathcal{T}_{\preceq b}, \mathcal{T}_{\preceq a}}$ if and only if $\text{LM}(F_g) \prec \frac{M}{s}$, where $M = \text{LCM}(\mathcal{T}_{\preceq a} + \mathcal{T}_{\preceq b})$ and s is defined as in equation (4): $s = \max_{\sigma\in\mathcal{T}} \Big\{\sigma,\ \{\sigma\} + \mathcal{T}_{\preceq g} \subseteq (\mathcal{T}_{\preceq a} + \mathcal{T}_{\preceq b})\Big\}$.*

Remark 3.5. Taking $s = \max_{\sigma\in\mathcal{T}}\{\sigma,\ \sigma\, g \preceq a\, b\}$ is not sufficient as proved by the following example. Let $\prec$ be the $\text{DRL}(y \prec x)$ monomial ordering and $T = U = \mathcal{T}_{\preceq x\,y}$. If $g = y^2$, then $x^2 = \max_{\sigma\in\mathcal{T}}\{\sigma,\ \sigma\, g \preceq (x\,y)^2\}$. But $H_{\mathcal{T}_{\preceq x^2}, \mathcal{T}_{\preceq y^2}}$ has the table term $[x^3]$, which is not in $H_{\mathcal{T}_{\preceq x\,y}, \mathcal{T}_{\preceq x\,y}}$.

4 A DIVISION-BASED ALGORITHM

The goal is now to design an algorithm based on polynomial division to determine all the C_g for $|$ minimal g such that $\text{LM}(F_c) = \text{LM}(P_{T+U}\, C_g \bmod B)$ is small enough.

We start with two sets of terms $T = \mathcal{T}_{\preceq a}$ and $U = \mathcal{T}_{\preceq b}$ so that $M = x_1^{D_1} \cdots x_n^{D_n} = \text{LCM}(T+U)$. We initialize $B = (B_1, \ldots, B_n) =$

$\left(x_1^{D_1+1}, \ldots, x_n^{D_n+1}\right)$, $R_{B_1} = [B_1, 0], \ldots, R_{B_n} = [B_n, 0]$ 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 3.4, C_g is a valid relation if $g \in T$ and $\text{LM}(F_g) \prec \frac{M}{s}$ with $s = \max_{\sigma \in \mathcal{T}}\{\sigma,\ \{\sigma\} + \mathcal{T}_{\preceq g} \subseteq (T+U)\}$. 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, two pieces of information can be retrieved: m and $\frac{M}{\text{LM}(F_m)}$ (Sakata's fail [26]) are 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 4.1. *Pick a failing pair $R_m = [F_m, C_m]$ with $h = q\, m$.*

(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 $[R_m, R_{B_1}, \ldots, R_{B_n}, R_{t_1}, \ldots, R_{t_n}]$ where $C_{t_1}, \ldots, C_{t_r}$ are failing relations and $\text{LM}(F_{t_1}) \succ \cdots \succ \text{LM}(F_{t_r})$. To do so, compute first $Q_m, Q_{B_1}, \ldots, Q_{B_n}, Q_{t_1}, \ldots, Q_{t_r}$ the quotients of the division of $F_{m'}$ by the list of polynomials $[F_m, B_1, \ldots, B_n, F_{t_1}, \ldots, F_{t_r}]$ and then return $R_h = R_{m'} - Q_m\, R_m - Q_{B_1}\, R_{B_1} - \cdots - Q_{B_n}\, R_{B_n} - Q_{t_1}\, R_{t_1} - \cdots - Q_{t_r}\, R_{t_r}$.*

(2) *otherwise, compute R_h as the normal form of $q\, R_m$ wrt. to the list $[R_{B_1}, \ldots, R_{B_n}, R_{t_1}, \ldots, R_{t_n}]$.*

Remark 4.2. If $q\ \text{LM}(F_m)$ is in the ideal spanned by B, then case 2 of Instruction 4.1 is equivalent to computing the normal form of $[q\ \text{LM}(F_m), 0]$ wrt. $[R_m, R_{B_1}, \ldots, R_{B_n}, R_{t_1}, \ldots, R_{t_r}]$. 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 $R_{x_1^{i_1+1}}, \ldots, R_{x_n^{i_n+1}}$. 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 $[R_1, R_{B_1}, \ldots, R_{B_n}]$. 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 4.3 (See [6]). Let $\mathbf{b} = \left(\binom{i}{j}\right)_{(i,j)\in\mathbb{N}^2}$ be the binomial table, $\prec$ be the $\text{DRL}(y \prec x)$ monomial ordering and $T = \mathcal{T}_{\preceq x^3}$ and $U = \{1\}$ be sets of terms. We have $T = T+U$ 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 $R_{B_1} = [x^4, 0]$, $R_{B_2} = [y^4, 0]$.

- 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 $R_x = [F_x, C_x] = [x^3\, y^2 + x^2\, y^2 - 2\, x\, y^2 - y^3, x - 1]$.
- 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 and x are both in the staircase. Thus $\mathcal{H} = \{y^2, x\, y, x^2\}$. 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. $[R_y, R_{B_1}, R_{B_2}, R_1, R_x]$;
  - $R_{x\,y} = [-x^2\, y^2 - 3\, x\, y^3 - 2\, x\, y^2 - y^3, x\, y - y - 1]$ by computing the normal form of R_1 wrt. $[R_y, R_{B_1}, R_{B_2}, R_x]$;
  - $R_{x^2} = [-3\, x^2\, y^2 - x\, y^3 + 2\, x\, y^2 + y^3, x^2 - 2\, x + 1]$ by computing the normal form of $[x\ \text{LM}(F_x), 0] = [x^4\, y^2, 0]$ wrt. $[R_x, R_{B_1}, R_{B_2}, R_1, R_y]$.
- There is no need to test R_x since we know x is in the staircase.
- As $F_{y^2} = 0$, then the relation is necessarily valid.
- As $\text{LM}(F_{x\,y}) = x^2\, y^2 = \frac{M}{x\,y}$, then the relation $C_{x\,y}$ might fail when shifting by $x\, y$ or greater. As it can only be tested up to a shift $s = \max_{\sigma\in\mathcal{T}} = \{\sigma,\ \sigma\, x\, y \preceq x^3\} = x$, then it actually succeeds.
- Likewise, as $\text{LM}(F_{x^2}) = x^2\, y^2 = \frac{M}{x\,y}$, then the relation C_{x^2} can only be tested up to a shift $s = \max_{\sigma\in\mathcal{T}} = \{\sigma,\ \sigma\, x^2 \preceq x^3\} = x$, then it succeeds.

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

The algorithm is the following Algorithm 4.4. It uses functions

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

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 4.4: Polynomial Scalar-FGLM

Input: A table $\mathbf{w} = (w_i)_{i\in\mathbb{N}^n}$ with coefficients in $\mathbb{K}$, a monomial ordering $\prec$ and two monomials a and b.
Output: A set G of relations generating the ideal of relations.

$T \coloneqq \{t \in \mathcal{T},\ t \preceq a\}$, $U \coloneqq \{u \in \mathcal{T},\ u \preceq b\}$.
$M \coloneqq \text{LCM}(T)\ \text{LCM}(U)$.
For i **from** 1 **to** n **do** $R_{B_i} \coloneqq \left[x_i^{1+\deg_{x_i} M}, 0\right]$. // pairs on the edge
$P \coloneqq \sum_{\tau\in(T+U)} [\tau]\, \frac{M}{\tau}$. // the mirror of the truncated generating series
$R \coloneqq \{[P, 1]\}$. // set of pairs $[F_m, C_m] = [P \cdot C_m \bmod B, C_m]$ to be tested
$R' \coloneqq \emptyset$. // set of failing pairs
$G \coloneqq \emptyset$, $S \coloneqq \emptyset$. // the future Gröbner basis and staircase
While $R \neq \emptyset$ **do**
  $R_m = [F_m, C_m] \coloneqq$ first element of R and remove it from R.
  If *$m \notin T$ or $\text{LM}(F_m)$ is as in Proposition 3.4* **then** // good relation
    $G \coloneqq G \cup \{C_m\}$.
  Else // bad relation
    $R' \coloneqq R' \cup \{R_m\}$.
    For all $r \in R$ **do** // reduce next pairs with it
      $r \coloneqq$ NormalForm$(r, [R_{B_1}, \ldots, R_{B_r}, R_m])$.
    $S \coloneqq$ Stabilize $\left(S \cup \left\{m, \frac{M}{\text{LM}(F_m)}\right\}\right)$.
    $H \coloneqq$ 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 4.1 and add it to R.
Return G.

Remark 4.5. 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 4.6 (See [6]). We give the trace of the Polynomial Scalar-FGLM algorithm with the slight modification above called on the table $\mathbf{w} = ((2\,i+1) + (2\,j-1)\,(-1)^{i+j})_{(i,j)\in\mathbb{N}^2}$, the stopping monomials 1 and y^5 and the monomial ordering $\text{DRL}(y \prec x)$.

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

Pair $R_1 = [F_1, C_1] = [P, 1]$, $R \coloneqq \emptyset$ and since $1 \in T$ but $\text{LM}(F_1) = x^3\,y^5 \succeq \frac{M}{s} = x^4$, then
- $R' \coloneqq \{R_1\}, S \coloneqq \{1, x\}$ and $H \coloneqq \{y, x^2\}$.
- We make new pairs added to R:
  - $R_y = [F_y, C_y] \coloneqq \text{NormalForm}([y\ \text{LM}(F_1), 0], [R_1, R_{B_1}, R_{B_2}])$ which can be normalized into $R_y, = [4\,x^4\,y^4 - \cdots, y - 1]$;
  - $R_{x^2} = [F_{x^2}, C_{x^2}] \coloneqq \text{NormalForm}([x^2\ \text{LM}(F_1), 0], [R_1, R_{B_1}, R_{B_2}])$ which can be normalized into $R_{x^2} = [4\,x^4\,y^3 - \cdots, x^2 - x - 1]$.

Pair $R_y = [F_y, C_y], R \coloneqq \{R_{x^2}\}$ and since $y \in T$ but $\text{LM}(F_y) = x^4\,y^4 \succeq \frac{M}{s} = x^4\,y$, then
- $R' \coloneqq \{R_1, R_y\}, S \coloneqq \{1, y, x\}$ and $H \coloneqq \{y^2, x\,y, x^2\}$.
- 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 $R_{y^2} = [F_{y^2}, C_{y^2}] \coloneqq \text{NormalForm}(y\,R_y, [R_{B_1}, R_{B_2}, R_1, R_y])$ which can be normalized into $R_{y^2} = [-4\,x^4\,y^3 - \cdots, y^2 - x + 2\,y - 1]$;
  - $R_{x\,y} = [F_{x\,y}, C_{x\,y}] \coloneqq \text{NormalForm}([x\ \text{LM}(F_y), 0], [R_y, R_{B_1}, R_{B_2}, R_1])$ which can be normalized into $R_{x\,y} = [4\,x^4\,y^2 - \cdots, x\,y - x + y - 1]$.
  - Nothing is done for x^2 since R_{x^2} already exists.

Pair $R_{y^2} = [F_{y^2}, C_{y^2}], R \coloneqq \{R_{x\,y}, R_{x^2}\}$ and since $y^2 \in T$ but $\text{LM}(F_{y^2}) = x^4\,y^3 \succeq \frac{M}{s} = x^4\,y^2$, then
- As $\text{LM}(F_{x^2}) \succeq \text{LM}(F_{y^2})$, we reduce it and obtain $R_{x^2} \coloneqq [-8\,x^2\,y^4 - \cdots, x^2 + y^2 - 2\,x + 2\,y - 2]$.
- $R' \coloneqq \{R_1, R_y, R_{y^2}\}, S \coloneqq \{1, y, x, y^2\}$ and $H \coloneqq \{x\,y, x^2, y^3\}$.
- We make new pairs added to R:
  - $R_{x\,y}$ 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 $R_{y^3} = [F_{y^3}, C_{y^3}] \coloneqq \text{NormalForm}(R_y, [R_{y^2}, R_{B_1}, R_{B_2}, R_y, R_1])$ which can be normalized into $R_{y^3} = [4\,x^3\,y^4 - \cdots, y^3 - x\,y + y^2 + x - 2\,y]$.

Pair $R_{x\,y} = [F_{x\,y}, C_{x\,y}], R \coloneqq \{R_{x^2}, R_{y^3}\}$ and since $x\,y \in T$ and $\text{LM}(F_{x\,y}) = x^4\,y^2 \prec \frac{M}{s} = x^2\,y^5$, then
- $G \coloneqq \{x\,y - x + y - 1\}$.
- As $C_{y^3} = y^3 - x\,y + y^2 + x - 2\,y$ has a term in $x\,y$, we update $R_{y^3} \coloneqq R_{y^3} + R_{x\,y} = [4\,x^3\,y^4 - \cdots, y^3 + y^2 - y - 1]$.

Pair $R_{x^2} = [F_{x^2}, C_{x^2}], R \coloneqq \{R_{y^3}\}$ and since $x^2 \in T$ and $\text{LM}(F_{x^2}) = x^2\,y^4 \prec \frac{M}{s} = x^2\,y^5$, then
- $G \coloneqq \{x\,y - x + y - 1, x^2 + y^2 - 2\,x + 2\,y - 2\}$.

Pair $R_{y^3} = [F_{y^3}, C_{y^3}], R \coloneqq \emptyset$ and since $y^3 \in T$ and $\text{LM}(F_{y^3}) = x^3\,y^4 \prec \frac{M}{s} = x^4\,y^3$, then
- $G \coloneqq \{x\,y - x + y - 1, x^2 + y^2 - 2\,x + 2\,y - 2, y^3 + y^2 - y - 1\}$.

We return G.

Theorem 4.7. *Let a table $\mathbf{w}$, a monomial ordering $\prec$ 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 $\prec$ and its staircase S satisfy $a \succeq \max(S \cup \text{LM}(\mathcal{G}))$ and for all $g \preceq a$, $s = \max_{\sigma\in\mathcal{T}}\{\sigma,\ \{\sigma\} + \mathcal{T}_{\preceq g} \subseteq (\mathcal{T}_{\preceq a} + \mathcal{T}_{\preceq b})\}$, we have $\max(S) \preceq s$.*

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

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

Starting with $R_1 = [F_1, C_1] = [P_{\mathcal{T}_{\preceq a}+\mathcal{T}_{\preceq b}}, 1]$, $\text{LM}(F_1)$ yields exactly the fail of relation $C_1 = C^\star_1$ 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 $\mu \prec h$, the pair $R_\mu = [F_\mu, C_\mu]$ made by the Polynomial Scalar-FGLM algorithm is equivalent to $C^\star_\mu$, that is either both C_μ and $C^\star_\mu$ fail when shifting by exactly the same monomial or they both succeed on $\mathcal{T}_{\preceq a} + \mathcal{T}_{\preceq b}$.

Since C_μ and $C^\star_\mu$ 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 $C^\star_m$ have been made. In the BMS algorithm, the relation $C^\star_h$ is obtained as $\frac{h}{m}\,C^\star_m - \sum_{\mu\prec h} q^\star_\mu\,C^\star_\mu$ while in the Polynomial Scalar-FGLM algorithm, C_h is made as $\frac{h}{m}\,C_m - \sum_{\mu\prec h} q_\mu\,C_\mu$. In each computation, $q^\star_\mu$ and q_μ are chosen so that $C^\star_m$ and C_m have the largest fail (or equivalently F_m has the least leading monomial), hence $C^\star_m$ 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 $\sigma_1\,\sigma_2 \in S$, $\sigma_1\,h \preceq a$, $\sigma_2 \preceq b$ and the column labeled with $\sigma_1\,h$ of the matrix $H_{\mathcal{T}_{\preceq b}, \mathcal{T}_{\preceq a}}$ is independent from the previous ones. For $g \in \text{LM}(\mathcal{G})$, by Section 3.2, the relation C_g has been tested with a shift $s = \max_{\sigma\in\mathcal{T}}\{\sigma,\ \{\sigma\} + \mathcal{T}_{\preceq g} \subseteq (\mathcal{T}_{\preceq a} + \mathcal{T}_{\preceq b}) \subseteq \mathcal{T}_{\preceq a\,b}\}$. 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 $\mathcal{T}_{\preceq a}$ and $\mathcal{T}_{\preceq b}$ are stable by division, so is $\mathcal{T}_{\preceq a} + \mathcal{T}_{\preceq b}$. Let us recall that the support of $P_{\mathcal{T}_{\preceq a}+\mathcal{T}_{\preceq b}}$ is $\left\{\frac{M}{\tau},\ \tau \in (\mathcal{T}_{\preceq a} + \mathcal{T}_{\preceq b})\right\}$, $M = \text{LCM}(\mathcal{T}_{\preceq a} + \mathcal{T}_{\preceq b})$. Since each F_m satisfies $F_m = P_{\mathcal{T}_{\preceq a}+\mathcal{T}_{\preceq b}}\,C_m \bmod B$, then the monomials in the support of F_m are multiples of the monomials in the support of $P_{\mathcal{T}_{\preceq a}+\mathcal{T}_{\preceq b}}$ and thus are included in the support of $P_{\mathcal{T}_{\preceq a}+\mathcal{T}_{\preceq b}}$. 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. □

5 AN ADAPTIVE VARIANT

In this section, due to space limitation, we only give some ideas to design an adaptive version of the Polynomial Scalar-FGLM.

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 $\text{DRL}(x_n \prec \cdots \prec x_1)$ ordering, whether the Gröbner basis is $\{x_n, \ldots, x_2, x_1^d\}$ or all the monomials of degree d: $\{x_n^d, x_{n-1}\,x_n^{d-1}, \ldots, x_1\,x_n^{d-1}, \ldots, x_1^d\}$, the BMS algorithm requires to visit all the monomials up to $x_1^{2\,d-1}$ making the algorithm costly in the former case compared to the size of the staircase.

The Adaptive Scalar-FGLM algorithm [3] 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 $\{x_n, \ldots, x_2, x_1^d\}$, it computes the rank of matrices $H_{\{1\},\{1\}}, H_{\{1,x_n\},\{1,x_n\}}, \ldots, H_{\{1,x_2\},\{1,x_2\}}, H_{\{1,x_1\},\{1,x_1\}}$, $H_{\{1,x_1,x_1^2\},\{1,x_1,x_1^2\}}, \ldots, H_{\{1,x_1,\ldots,x_1^d\},\{1,x_1,\ldots,x_1^d\}}$. It thus requires merely $2\,(n+d)-1$ table terms instead of $\binom{n+2\,d-1}{n}$.

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 $\mathcal{T}_{\leq b}$.

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 $P_{2\,S}$, a monomial ideal $B_{2\,S}$, determined as in Section 3.2 and pairs $R_{2\,S,t} = [F_{2\,S,t}, C_t] = [P_{2\,S}\,C_t \bmod B_{2\,S}, C_t]$. At the next step, we compute $P_{2\,(S\cup\{m\})}$ from $P_{2\,S}$ by shifting it by $\frac{\text{LCM}(S\cup\{m\})}{\text{LCM}(S)}$ and adding the missing terms, update $B_{2\,S}$ into $B_{2\,(S\cup\{m\})}$ and likewise update each $R_{2\,(S\cup\{m\}),t}$ by shifting $F_{2\,S,t}$ 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 $R_{2\,(S\cup\{m\}),B_1}, \ldots, R_{2\,(S\cup\{m\}),B_n}$, where $B_1, \ldots, B_n \in B_{2\,S}$.

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 $R_{2\,(S\cup\{m\}),B_1}, \ldots, R_{2\,(S\cup\{m\}),B_n}$.

This transformation of the Polynomial Scalar-FGLM is only partial as we do not know yet how to initialize $R_{2\,(S\cup\{m\}),m}$ as the quotient of two pairs of polynomials.

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 $\text{DRL}(z \prec y \prec x)$ ordering and denote by S the staircase and $\text{LM}(\mathcal{G})$ the set of the leading terms of the Gröbner basis of relations.

- **Simplex tables:** $\text{LM}(\mathcal{G}) = \left\{y^d, x\,y^{d-1}, \ldots, x^d\right\}$ in dimension 2 and $\text{LM}(\mathcal{G}) = \left\{z^d, y\,z^{d-1}, x\,z^{d-1}, \ldots, y^d, x\,y^{d-1}, \ldots, x^d\right\}$ in dimension 3, i.e. all the monomials of degree d.
- **L-shape tables:** $\text{LM}(\mathcal{G}) = \left\{x\,y, y^d, x^d\right\}$ in dimension 2 and $\text{LM}(\mathcal{G}) = \left\{y\,z, x\,z, x\,y, z^d, y^d, x^d\right\}$ in dimension 3.
- **Rectangle tables:** $\text{LM}(\mathcal{G}) = \left\{y^{\lfloor d/2\rfloor}, x^d\right\}$ in dimension 2 and $\text{LM}(\mathcal{G}) = \left\{z^{\lceil d/3\rceil}, y^{\lfloor d/2\rfloor}, x^d\right\}$ dimension 3.

Let $a = \max(S \cup \text{LM}(\mathcal{G}))$. Generically, a relation C_m fails when shifted by m. From [9, 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 higher number of operations. The needed input set of monomials for the Artinian Gorenstein border bases algorithm (AGbb) of [23] to recover the ideal of relations was higher than expected in some situations. It explains the big overhead in Figure 1 in dimension 2:

- **Simplex tables:** The correct border basis is not recovered. In dimension 2, we have to visit all the monomials of degree around $5\,d$ or $6\,d$ to close the staircase. This also yields relations of degree higher than expected. For $d = 8$, we obtained $\text{LM}(\mathcal{B}) = \left\{x^i\,y^{18-i},\ 0 \leq i \leq 18\right\}$.
- **L-shape tables:** The correct border basis is not recovered. In dimension 2, we have to visit all the monomials of degree $5\,d - 10$ to close the staircase. This yields relations of degree higher than expected, though. For $d = 8$, we obtained $\text{LM}(\mathcal{B}) = \left\{x\,y^i, x^i\,y, y^{13}, x^{13},\ 1 \leq i \leq 12\right\}$.
- **Rectangle tables:** The border basis is recovered whenever the algorithm visits all the monomials of degree at most $2\ (d + \lfloor d/2\rfloor + \lceil d/3\rceil - 1)$.

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 $(\#S)^2 = O(\#S\,(\#S + \#\mathcal{G}))$ 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.
- **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.

ACKNOWLEDGMENTS

We thank the anonymous referees for their careful reading and their helpful comments. The authors are partially supported by the PGMO grant GAMMA.

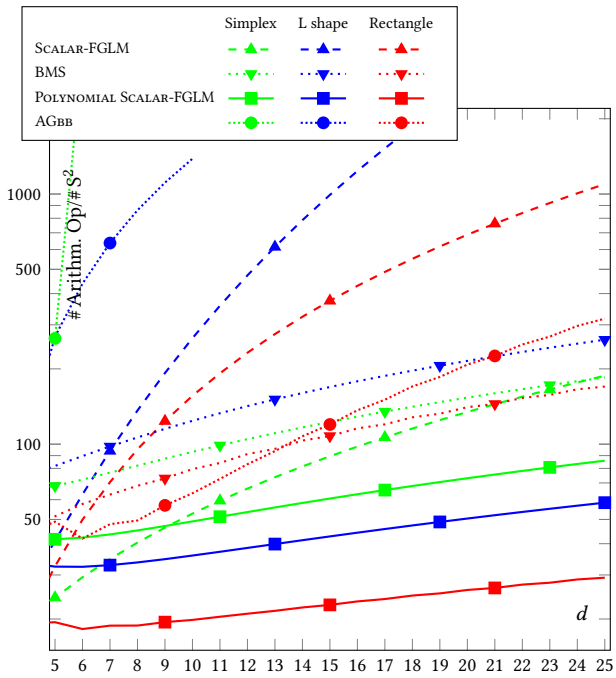

Figure 1: Number of arithmetic operations (2D)

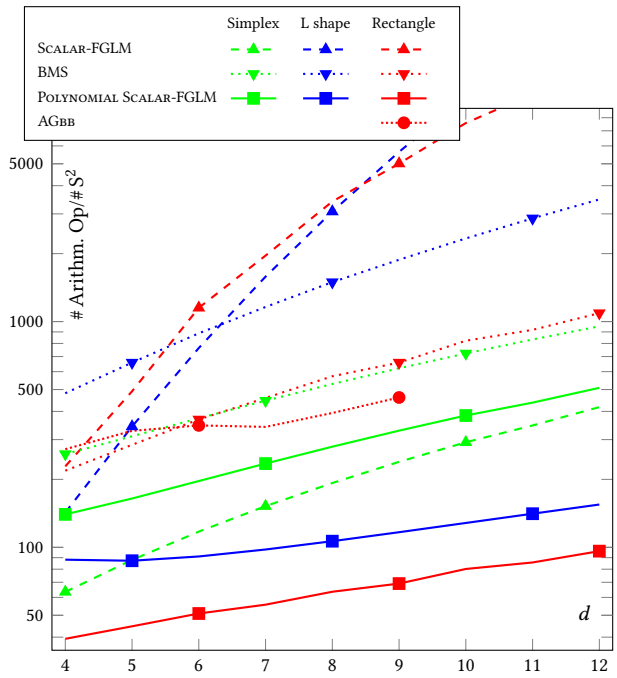

Figure 2: Number of arithmetic operations (3D)

REFERENCES

[1] B. Beckermann and G. Labahn. 1994. A Uniform Approach for the Fast Computation of Matrix-Type Pade Approximants. *SIAM J. Matrix Anal. Appl.* 15, 3 (1994), 804–823. https://doi.org/10.1137/S0895479892230031

[2] E. Berlekamp. 1968. Nonbinary BCH decoding. *IEEE Trans. Inform. Theory* 14, 2 (1968), 242–242. https://doi.org/10.1109/TIT.1968.1054109

[3] J. Berthomieu, B. Boyer, and J.-Ch. Faugère. 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, 61–68. https://doi.org/10.1145/2755996.2756673

[4] J. Berthomieu, B. Boyer, and J.-Ch. Faugère. 2017. Linear Algebra for Computing Gröbner Bases of Linear Recursive Multidimensional Sequences. *Journal of Symbolic Computation* 83, Supplement C (Nov. 2017), 36–67. https://doi.org/10.1016/j.jsc.2016.11.005 Special issue on the conference ISSAC 2015: Symbolic computation and computer algebra.

[5] J. Berthomieu and J.-Ch. Faugère. 2017. In-depth comparison of the Berlekamp – Massey – Sakata and the Scalar-FGLM algorithms: the non adaptive variants. (May 2017). https://hal.inria.fr/hal-01516708 preprint.

[6] J. Berthomieu and J.-Ch. Faugère. 2018. Experiments. (2018). http://www-polsys.lip6.fr/~berthomieu/ISSAC2018.html

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

[8] R.C. Bose and D.K. Ray-Chaudhuri. 1960. On a class of error correcting binary group codes. *Information and Control* 3, 1 (1960), 68 – 79. https://doi.org/10.1016/S0019-9958(60)90287-4

[9] M. Bras-Amorós and M. E. O'Sullivan. 2006. The Correction Capability of the Berlekamp–Massey–Sakata Algorithm with Majority Voting. *Applicable Algebra in Engineering, Communication and Computing* 17, 5 (2006), 315–335. https://doi.org/10.1007/s00200-006-0015-8

[10] R. P. Brent, F. G. Gustavson, and D. Y.Y. Yun. 1980. Fast solution of Toeplitz systems of equations and computation of Padé approximants. *Journal of Algorithms* 1, 3 (1980), 259 – 295. https://doi.org/10.1016/0196-6774(80)90013-9

[11] D. G. Cantor and E. Kaltofen. 1991. On Fast Multiplication of Polynomials Over Arbitrary Algebras. *Acta Informatica* 28 (1991), 693–701.

[12] J. W. Cooley and J. W. Tukey. 1965. An Algorithm for the Machine Calculation of Complex Fourier Series. *Math. Comp.* 19, 90 (1965), 297–301. http://www.jstor.org/stable/2003354

[13] D. Cox, J. Little, and D. O'Shea. 2015. *Ideals, Varieties, and Algorithms* (fourth ed.). Springer, New York. xvi+646 pages. An introduction to computational algebraic geometry and commutative algebra.

[14] J. Dornstetter. 1987. On the equivalence between Berlekamp's and Euclid's algorithms (Corresp.). *IEEE Transactions on Information Theory* 33, 3 (1987), 428–431. https://doi.org/10.1109/TIT.1987.1057299

[15] J.-Ch. Faugère and Ch. Mou. 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, 115–122. https://doi.org/10.1145/1993886.1993908

[16] J.-Ch. Faugère and Ch. Mou. 2017. Sparse FGLM algorithms. *Journal of Symbolic Computation* 80, 3 (2017), 538 – 569. https://doi.org/10.1016/j.jsc.2016.07.025

[17] P. Fitzpatrick and J. Flynn. 1992. A Gröbner basis technique for Padé approximation. *J. Symbolic Comput.* 13, 2 (1992), 133 – 138. https://doi.org/10.1016/S0747-7171(08)80087-9

[18] P. Fitzpatrick and G.H. Norton. 1990. Finding a basis for the characteristic ideal of an n-dimensional linear recurring sequence. *IEEE Trans. Inform. Theory* 36, 6 (1990), 1480–1487. https://doi.org/10.1109/18.59953

[19] A. Hocquenghem. 1959. Codes correcteurs d'erreurs. *Chiffres* 2 (1959), 147 – 156.

[20] E. Jonckheere and Ch. Ma. 1989. A simple Hankel interpretation of the Berlekamp-Massey algorithm. *Linear Algebra Appl.* 125, 0 (1989), 65 – 76. https://doi.org/10.1016/0024-3795(89)90032-3

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

[22] J. L. Massey. 1969. Shift-register synthesis and BCH decoding. *IEEE Trans. Inform. Theory* IT-15 (1969), 122–127.

[23] Bernard Mourrain. 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, 333–340. https://doi.org/10.1145/3087604.3087632

[24] Sh. Sakata. 1988. Finding a minimal set of linear recurring relations capable of generating a given finite two-dimensional array. *J. Symbolic Comput.* 5, 3 (1988), 321–337. https://doi.org/10.1016/S0747-7171(88)80033-6

[25] Sh. Sakata. 1990. Extension of the Berlekamp-Massey Algorithm to N Dimensions. *Inform. and Comput.* 84, 2 (1990), 207–239. https://doi.org/10.1016/0890-5401(90)90039-K

[26] Sh. Sakata. 2009. The BMS Algorithm. In *Gröbner Bases, Coding, and Cryptography*, Massimiliano Sala, Shojiro Sakata, Teo Mora, Carlo Traverso, and Ludovic Perret (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 143–163. https://doi.org/10.1007/978-3-540-93806-4_9

[27] N. Wiener. 1964. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series.* The MIT Press, Cambridge, MA.