HAL
open science

Computing syzygies in finite dimension using fast linear algebra

Vincent Neiger, Éric Schost

To cite this version:

Vincent Neiger, Éric Schost. Computing syzygies in finite dimension using fast linear algebra. Journal of Complexity, In press, 10.1016/j.jco.2020.101502 . hal-02392488v2

HAL Id: hal-02392488

https://unilim.hal.science/hal-02392488v2

Submitted on 19 Jun 2020

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.

Computing syzygies in finite dimension using fast linear algebra

Vincent Neiger

Univ. Limoges, CNRS, XLIM, UMR 7252, F-87000 Limoges, France

Éric Schost

University of Waterloo, Waterloo ON, Canada

Abstract

We consider the computation of syzygies of multivariate polynomials in a finite-dimensional setting: for a $\mathbb{K}[X_1, \ldots, X_r]$-module $\mathcal{M}$ of finite dimension D as a $\mathbb{K}$-vector space, and given elements $f_1, \ldots, f_m$ in $\mathcal{M}$, the problem is to compute syzygies between the f_i's, that is, polynomials $(p_1, \ldots, p_m)$ in $\mathbb{K}[X_1, \ldots, X_r]^m$ such that $p_1 f_1 + \cdots + p_m f_m = 0$ in $\mathcal{M}$. Assuming that the multiplication matrices of the r variables with respect to some basis of $\mathcal{M}$ are known, we give an algorithm which computes the reduced Gröbner basis of the module of these syzygies, for any monomial order, using $O(mD^{\omega-1} + rD^{\omega} \log(D))$ operations in the base field $\mathbb{K}$, where ω is the exponent of matrix multiplication. Furthermore, assuming that $\mathcal{M}$ is itself given as $\mathcal{M} = \mathbb{K}[X_1, \ldots, X_r]^n / \mathcal{N}$, under some assumptions on $\mathcal{N}$ we show that these multiplication matrices can be computed from a Gröbner basis of $\mathcal{N}$ within the same complexity bound. In particular, taking $n = 1$, $m = 1$ and $f_1 = 1$ in $\mathcal{M}$, this yields a change of monomial order algorithm along the lines of the FGLM algorithm with a complexity bound which is sub-cubic in D.

Keywords: Gröbner basis, syzygies, complexity, fast linear algebra.

1. Introduction

In what follows, $\mathbb{K}$ is a field and we consider the polynomial ring $\mathbb{K}[\boldsymbol{X}] = \mathbb{K}[X_1, \ldots, X_r]$. The set of $m \times n$ matrices over a ring $\mathcal{R}$ is denoted by $\mathcal{R}^{m \times n}$; when orientation matters, a vector in $\mathcal{R}^n$ is considered as being in $\mathcal{R}^{1 \times n}$ (row vector) or in $\mathcal{R}^{n \times 1}$ (column vector). We are interested in the efficient computation of relations, known as syzygies, between elements of a $\mathbb{K}[\boldsymbol{X}]$-module $\mathcal{M}$.

Let us write the $\mathbb{K}[\boldsymbol{X}]$-action on $\mathcal{M}$ as $(p, f) \in \mathbb{K}[\boldsymbol{X}] \times \mathcal{M} \mapsto p \cdot f$, and let $f_1, \ldots, f_m$ be in $\mathcal{M}$. Then, for a given monomial order $\prec$ on $\mathbb{K}[\boldsymbol{X}]^m$, we want to compute the Gröbner basis of the kernel of the homomorphism

$$\begin{array}{rcl} \mathbb{K}[\boldsymbol{X}]^m & \to & \mathcal{M} \\ (p_1, \ldots, p_m) & \mapsto & p_1 \cdot f_1 + \cdots + p_m \cdot f_m. \end{array}$$

This kernel is called the *module of syzygies* of $(f_1, \ldots, f_m)$ and written $\mathrm{Syz}_{\mathcal{M}}(f_1, \ldots, f_m)$.

In this paper, we focus on the case where $\mathcal{M}$ has finite dimension D as a $\mathbb{K}$-vector space; as a result, the quotient $\mathbb{K}[\boldsymbol{X}]^m / \mathrm{Syz}_{\mathcal{M}}(f_1, \ldots, f_m)$ has dimension at most D as a $\mathbb{K}$-vector space. Then one may adopt a linear algebra viewpoint detailed in the next paragraph, where the elements of $\mathcal{M}$ are seen as row vectors of length D and the multiplication by the variables is represented

by so-called multiplication matrices. This representation was used and studied in [2, 37, 1, 27], mainly in the context where $\mathcal{M}$ is a quotient $\mathbb{K}[\boldsymbol{X}]/\mathcal{I}$ for some ideal $\mathcal{I}$ (thus zero-dimensional of degree D) and more generally a quotient $\mathbb{K}[\boldsymbol{X}]^n/\mathcal{N}$ for some submodule $\mathcal{N} \subseteq \mathbb{K}[\boldsymbol{X}]^n$ with $n \in \mathbb{N}_{>0}$ (see [1, Sec. 4.4 and 6]). This representation with multiplication matrices allows one to perform computations in such a quotient via linear algebra operations.

Assume we are given a $\mathbb{K}$-vector space basis $\mathcal{F}$ of $\mathcal{M}$. For i in $\{1, \ldots, r\}$, the matrix of the structure morphism $f \mapsto X_i \cdot f$ with respect to this basis is denoted by $\boldsymbol{M}_i$; this means that for f in $\mathcal{M}$ represented by the vector $\boldsymbol{f} \in \mathbb{K}^{1\times D}$ of its coefficients on $\mathcal{F}$, the coefficients of $X_i \cdot f \in \mathcal{M}$ on $\mathcal{F}$ are $\boldsymbol{f}\boldsymbol{M}_i$. We call $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_r$ *multiplication matrices*; note that they are pairwise commuting. The data formed by these matrices defines the module $\mathcal{M}$ up to isomorphism; we use it as a representation of $\mathcal{M}$. For p in $\mathbb{K}[\boldsymbol{X}]$ and for f in $\mathcal{M}$ represented by the vector $\boldsymbol{f} \in \mathbb{K}^{1\times D}$ of its coefficients on $\mathcal{F}$, the coefficients of $p \cdot f \in \mathcal{M}$ on $\mathcal{F}$ are $\boldsymbol{f}\, p(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_r)$; hereafter this vector is written $p \cdot \boldsymbol{f}$. From this point of view, syzygy modules can be described as follows.

Definition 1.1. *For m and D in $\mathbb{N}_{>0}$, let $\boldsymbol{M} = (\boldsymbol{M}_1, \ldots, \boldsymbol{M}_r)$ be pairwise commuting matrices in $\mathbb{K}^{D\times D}$, and let $\boldsymbol{F} \in \mathbb{K}^{m\times D}$. Denoting by $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_m$ the rows of $\boldsymbol{F}$, for $\boldsymbol{p} = (p_1, \ldots, p_m) \in \mathbb{K}[\boldsymbol{X}]^m$ we write*

$$\boldsymbol{p} \cdot \boldsymbol{F} = p_1 \cdot \boldsymbol{f}_1 + \cdots + p_m \cdot \boldsymbol{f}_m = \boldsymbol{f}_1\, p_1(\boldsymbol{M}) + \cdots + \boldsymbol{f}_m\, p_m(\boldsymbol{M}) \in \mathbb{K}^{1\times D}.$$

The syzygy module $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$*, whose elements are called syzygies for $\boldsymbol{F}$, is defined as*

$$\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F}) = \{\boldsymbol{p} \in \mathbb{K}[\boldsymbol{X}]^m \mid \boldsymbol{p} \cdot \boldsymbol{F} = \boldsymbol{0}\};$$

as noted above, $\mathbb{K}[\boldsymbol{X}]^m / \mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$ has dimension at most D as a $\mathbb{K}$-vector space.

In particular, if in the above context $\boldsymbol{F}$ is the matrix of the coefficients of $f_1, \ldots, f_m \in \mathcal{M}$ on the basis $\mathcal{F}$, then $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F}) = \mathrm{Syz}_{\mathcal{M}}(f_1, \ldots, f_m)$. Our main goal in this paper is to give a fast algorithm to solve the following problem (for the notions of monomial orders and Gröbner basis for modules, we refer to [13] and the overview in Section 2).

Problem 1 – *Gröbner basis of syzygies*

Input:

- a monomial order $\prec$ on $\mathbb{K}[\boldsymbol{X}]^m$,
- pairwise commuting matrices $\boldsymbol{M} = (\boldsymbol{M}_1, \ldots, \boldsymbol{M}_r)$ in $\mathbb{K}^{D\times D}$,
- a matrix $\boldsymbol{F} \in \mathbb{K}^{m\times D}$.

Output: the reduced $\prec$-Gröbner basis of $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$.

Example 1.2. An important class of examples has $r = 1$; in this case, we are working with univariate polynomials. Restricting further, consider the case $\mathcal{M} = \mathbb{K}[X_1]/\langle X_1^D\rangle$ endowed with the canonical $\mathbb{K}[X_1]$-module structure; then $\boldsymbol{M}_1$ is the upper shift matrix, whose entries are all 0 except those on the superdiagonal which are 1. Given $f_1, \ldots, f_m$ in $\mathcal{M}$, $(p_1, \ldots, p_m) \in \mathbb{K}[X_1]^m$ is a syzygy for $f_1, \ldots, f_m$ if $p_1 f_1 + \cdots + p_m f_m = 0 \bmod X_1^D$. Such syzygies are known as *Hermite-Padé approximants* of $(f_1, \ldots, f_m)$ [22, 40]. Using moduli other than X_1^D leads one to generalizations such as *M-Padé approximants* or *rational interpolants* (corresponding to a modulus that splits into linear factors) [33, 4, 49].

For $r = 1$, $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$ is known to be free of rank m. Bases of such $\mathbb{K}[X_1]$-modules are often described by means of their so-called *Popov* form [43, 26]. In commutative algebra terms, this

is a *term over position* Gröbner basis. Another common choice is the *Hermite* form, which is a *position over term* Gröbner basis [29]. □

Example 1.3. For arbitrary r, let $\mathcal{I}$ be a zero-dimensional ideal in $\mathbb{K}[\boldsymbol{X}]$ and let $\mathcal{M} = \mathbb{K}[\boldsymbol{X}]/\mathcal{I}$ with the canonical $\mathbb{K}[\boldsymbol{X}]$-module structure. Then, taking $m = 1$ and $f_1 = 1 \in \mathcal{M}$, we have

$$\mathrm{Syz}_{\mathcal{M}}(f_1) = \{p \in \mathbb{K}[\boldsymbol{X}] \mid p\, f_1 = 0\} = \{p \in \mathbb{K}[\boldsymbol{X}] \mid p \in \mathcal{I}\} = \mathcal{I}.$$

Suppose we know a Gröbner basis of $\mathcal{I}$ for some monomial order $\prec_1$, together with the corresponding monomial basis of $\mathcal{M}$, and the multiplication matrices of $X_1, \ldots, X_r$ in $\mathcal{M}$. Then solving Problem 1 amounts to computing the Gröbner basis of $\mathcal{I}$ for the new order $\prec$.

More generally for a given $f_1 = f \in \mathcal{M}$, the case $m = 1$ corresponds to the computation of the *annihilator* of f in $\mathbb{K}[\boldsymbol{X}]$, often denoted by $\mathrm{Ann}_{\mathbb{K}[\boldsymbol{X}]}(\{f\})$. Indeed, the latter set is defined as $\{p \in \mathbb{K}[\boldsymbol{X}] \mid pf = 0\}$, which is precisely $\mathrm{Syz}_{\mathcal{M}}(f)$. □

Example 1.4. For an arbitrary r, let $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_D$ be pairwise distinct points in $\mathbb{K}^r$, with $\boldsymbol{\alpha}_i = (\alpha_{i,1}, \ldots, \alpha_{i,r})$ for all i. Let $\mathcal{I}$ be the vanishing ideal of $\{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_D\}$ and $\mathcal{M} = \mathbb{K}[\boldsymbol{X}]/\mathcal{I}$. As above, take $m = 1$ and $f_1 = 1$, so that $\mathrm{Syz}_{\mathcal{M}}(1) = \mathcal{I}$.

The Chinese Remainder Theorem gives an explicit isomorphism $\mathcal{M} \to \mathbb{K}^D$ that amounts to evaluation at $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_D$. The multiplication matrices induced by this module structure on $\mathbb{K}^D$ are diagonal, with $\boldsymbol{M}_j$ having diagonal $(\alpha_{1,j}, \ldots, \alpha_{D,j})$ for $1 \leqslant j \leqslant r$. Taking $\boldsymbol{F} = [1 \ \cdots \ 1] \in \mathbb{K}^{1\times D}$, solving Problem 1 allows us to compute the Gröbner basis of the vanishing ideal $\mathcal{I}$ for any given order $\prec$. This problem was introduced by Möller and Buchberger [36]; it may be extended to cases where vanishing multiplicities are prescribed [35]. □

Example 1.5. Now we consider an extension of the Möller-Buchberger problem due to Kehrein, Kreuzer and Robbiano [27]. Given r pairwise commuting $d \times d$ matrices $\boldsymbol{N}_1, \ldots, \boldsymbol{N}_r$, we look for their ideal of syzygies, that is, the ideal $\mathcal{I}$ of polynomials $p \in \mathbb{K}[\boldsymbol{X}]$ such that $p(\boldsymbol{N}_1, \ldots, \boldsymbol{N}_r) = \boldsymbol{0}$. When $r = 1$, this ideal is generated by the minimal polynomial of $\boldsymbol{N}_1$.

One may see this problem in our framework by considering $\mathcal{M} = \mathbb{K}^{d\times d}$ endowed with the $\mathbb{K}[\boldsymbol{X}]$-module structure given by $X_k \cdot \boldsymbol{A} = \boldsymbol{A}\boldsymbol{N}_k$ for all $1 \leqslant k \leqslant r$ and $\boldsymbol{A} \in \mathbb{K}^{d\times d}$. The ideal $\mathcal{I}$ defined above is the module of syzygies $\mathrm{Syz}_{\mathcal{M}}(f)$ of the identity matrix $f = \boldsymbol{I}_d \in \mathcal{M}$, so we have $m = d$ and $D = d^2$ here. To form the input of Problem 1, we choose as a basis of $\mathcal{M}$ the list of elementary matrices $\mathcal{F} = (\boldsymbol{c}_{1,1}, \ldots, \boldsymbol{c}_{1,d}, \ldots, \boldsymbol{c}_{d,1}, \ldots, \boldsymbol{c}_{d,d})$ where $\boldsymbol{c}_{i,j}$ is the matrix in $\mathbb{K}^{d\times d}$ whose only nonzero entry is a 1 at index (i, j). Then, for $k \in \{1, \ldots, r\}$, the multiplication matrix $\boldsymbol{M}_k$ is the Kronecker product $\boldsymbol{I}_d \otimes \boldsymbol{N}_k$, that is, the block diagonal matrix in $\mathbb{K}^{d^2\times d^2}$ with d diagonal blocks, each of them equal to $\boldsymbol{N}_k$. Besides, the input $\boldsymbol{F} \in \mathbb{K}^{1\times d^2}$ is the vector of coordinates of $f = \boldsymbol{I}_d$ on the basis $\mathcal{F}$, so that $\mathcal{I} = \mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$ where $\boldsymbol{M} = (\boldsymbol{M}_1, \ldots, \boldsymbol{M}_r)$. □

Example 1.6. Our last example is a multivariate extension of Hermite-Padé approximants and involves arbitrary parameters $r \geqslant 1$ and $m \geqslant 2$. For a positive integer d, consider the ideal $\mathcal{I} = \langle X_1^d, \ldots, X_r^d \rangle$, and let $\mathcal{M} = \mathbb{K}[\boldsymbol{X}]/\mathcal{I}$. Then for given $f_2, \ldots, f_m$ in $\mathcal{M}$, which may be seen as polynomials truncated in degree d in each variable, the syzygy module $\mathrm{Syz}_{\mathcal{M}}(-1, f_2, \ldots, f_m)$ is

$$\{(p, q_2, \ldots, q_m) \in \mathbb{K}[\boldsymbol{X}]^m \mid p = q_2 f_2 + \cdots + q_m f_m \bmod \langle X_1^d, \ldots, X_r^d \rangle\}.$$

It was showed in [19, Thm. 3.1] that this module is generated by

$$\{f_i \boldsymbol{c}_1 + \boldsymbol{c}_i \mid 2 \leqslant i \leqslant m\} \cup \{X_k^d \boldsymbol{c}_i \mid 1 \leqslant k \leqslant r, 2 \leqslant i \leqslant m\},$$

where $\boldsymbol{c}_i$ is the coordinate vector $(0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{K}[\boldsymbol{X}]^m$ with 1 at index i. In the same reference, two algorithms are given to find the Gröbner basis of this syzygy module for arbitrary

monomial orders. One algorithm uses the FGLM change of order (extended to modules), based on the fact that the above generating set is a Gröbner basis for a well-chosen order. The other one proceeds iteratively on the d^r vanishing conditions; this approach is similar to the above-mentioned algorithm of [35], and can be seen as a multivariate generalization of the classical iterative algorithm for univariate Hermite-Padé approximation [49, 5].

For the linear algebra viewpoint used in this paper, consider the $\mathbb{K}$-vector space basis $\mathcal{F}$ of $\mathcal{M}$ formed by all monomials $X_1^{e_1} \cdots X_r^{e_r}$ for $0 \leqslant e_1, \ldots, e_r < d$ ordered by the lexicographic order $\prec_{\text{lex}}$ with $X_r \prec_{\text{lex}} \cdots \prec_{\text{lex}} X_1$. Computing bases of the above syzygy module is an instance of Problem 1 with $D = d^r$, taking for $\boldsymbol{F}$ the matrix of the coefficients of $(-1, f_2, \ldots, f_m)$ on the basis $\mathcal{F}$, and for $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_r$ the multiplication matrices of $X_1, \ldots, X_r$ in $\mathcal{M}$ with respect to $\mathcal{F}$. These matrices are types of block upper shift matrices which are nilpotent of order d. Taking $r = 2$ for example, $\boldsymbol{M}_1$ is block-diagonal with all diagonal blocks equal to the $d \times d$ upper shift matrix, while $\boldsymbol{M}_2$ is a matrix formed by d rows and d columns of $d \times d$ blocks which are all zero except those on the (blockwise) superdiagonal which are equal to the $d \times d$ identity matrix. □

Main result. For r variables and an input module of vector space dimension D, we design an algorithm whose cost is essentially that of performing fast linear algebra operations with r scalar matrices of dimensions $D \times D$. In the rest of the paper, ω is a feasible exponent for matrix multiplication over the field $\mathbb{K}$; the best known bound is $\omega < 2.38$ [12, 30].

Theorem 1.7. *Let $\prec$ be a monomial order on $\mathbb{K}[\boldsymbol{X}]^m$, let $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_r$ be pairwise commuting matrices in $\mathbb{K}^{D \times D}$, and let $\boldsymbol{F} \in \mathbb{K}^{m \times D}$. Then there is an algorithm which solves Problem 1 using*

$$O\left(mD^{\omega-1} + D^{\omega}(r + \log(d_1 \cdots d_r))\right) \subset O\left(mD^{\omega-1} + rD^{\omega}\log(D)\right)$$

operations in $\mathbb{K}$, where $d_k \in \{1, \ldots, D\}$ is the degree of the minimal polynomial of $\boldsymbol{M}_k$, for $1 \leqslant k \leqslant r$.

This theorem is proved in Section 3, based on Algorithm 3 which is built upon Algorithm 1 (computing the monomial basis) and Algorithm 2 (computing normal forms). Commonly encountered situations involve $m \leqslant D$ (see Examples 1.3 to 1.5), in which case the cost bound can be simplified as $O(rD^{\omega}\log(D))$. The interest in the more precise cost bound involving $d_1, \ldots, d_r$ comes from situations such as that of Example 1.6, where $d_1 \cdots d_r = d^r = D$ since all the matrices $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_r$ have a minimal polynomial of degree d; in that case, the first cost bound in the theorem becomes $O(mD^{\omega-1} + rD^{\omega} + D^{\omega}\log(D))$. Another refinement of the cost bound is given in Remark 3.11 for a particular order $\prec$, namely the term over position lexicographic order.

Our algorithm deals with the multiplication matrices $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_r$ one after another, allowing us to rely on an approach inspired by that designed for the univariate case $r = 1$ in [24]. This also helps us to introduce fast matrix multiplication, by avoiding the computation of many vector-matrix products involving each time a different matrix $\boldsymbol{M}_k$, and instead grouping these operations into some matrix-matrix products involving $\boldsymbol{M}_1$, then some others involving $\boldsymbol{M}_2$, etc. To our knowledge, this is the first time a general answer is given to Problem 1. For problems such as Examples 1.3 to 1.6, ours is the first algorithm that relies on fast linear algebra techniques, with the partial exception of [14], as discussed below.

Our cost bound can be compared to the input and output size of the problem. The input matrices are represented using $mD + rD^2$ field elements, and we will see in Sections 2 and 3 that one can represent the output Gröbner basis using at most $mD + rD^2$ field elements as well.

Overview of the algorithm. To introduce matrix multiplication in our solution to Problem 1, we rely on a linearization into linear algebra problems over $\mathbb{K}$. From $\boldsymbol{M}$ and $\boldsymbol{F}$, we build a matrix over $\mathbb{K}$ whose nullspace corresponds to a set of syzygies in $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$. This matrix is called a *multi-Krylov matrix*, in reference to its structure which exhibits a collection of Krylov subspaces of $\mathbb{K}^D$.

The multi-Krylov matrix is a generalization to several variables and to an arbitrary monomial order of the (striped-)Krylov matrices considered for example in [26, 7]; already in [26, Chap. 6], Popov and Hermite bases of modules over the univariate polynomials are obtained by means of Krylov matrix computations. Its construction is similar to the Sylvester matrix and more generally to the Macaulay matrix [47, 31, 32], which are commonly used when adopting a linear algebra viewpoint while dealing with operations on univariate and multivariate polynomials.

In what follows, given $\mathbf{e} = (e_1, \ldots, e_r)$ in $\mathbb{N}^r$, we write $\boldsymbol{X}^{\mathbf{e}} = X_1^{e_1} \cdots X_r^{e_r}$ and $\boldsymbol{M}^{\mathbf{e}} = \boldsymbol{M}_1^{e_1} \cdots \boldsymbol{M}_r^{e_r}$; the ith coordinate vector in $\mathbb{K}^m$ is written $\boldsymbol{c}_i$. Then the construction of the multi-Krylov matrix is based on viewing the product $\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i \cdot \boldsymbol{F}$, for a monomial $\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i \in \mathbb{K}[\boldsymbol{X}]^m$ and $\boldsymbol{F}$ in $\mathbb{K}^{m\times D}$, as $\boldsymbol{f}_i\boldsymbol{M}^{\mathbf{e}}$, where $\boldsymbol{f}_i$ is the ith row of $\boldsymbol{F}$. Since a polynomial in $\mathbb{K}[\boldsymbol{X}]^m$ is a $\mathbb{K}$-linear combination of monomials, this identity means that a syzygy in $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$ may be interpreted as a $\mathbb{K}$-linear relation between row vectors of the form $\boldsymbol{f}_i\boldsymbol{M}^{\mathbf{e}}$.

Choosing some degree bounds $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_r) \in \mathbb{N}^r_{>0}$, our multi-Krylov matrix is then formed by all such rows $\boldsymbol{f}_i\boldsymbol{M}^{\mathbf{e}}$, for $1 \leqslant i \leqslant m$ and $\mathbf{0} \leqslant \mathbf{e} < \boldsymbol{\beta}$ entrywise, ordered according to the monomial order $\prec$. For sufficiently large $\boldsymbol{\beta}$ (taking $\beta_k = D$ for all k is enough, for instance), we show in Section 3.2 that the row rank profile of this matrix corresponds to the $\prec$-monomial basis of the quotient $\mathbb{K}[\boldsymbol{X}]^m / \mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$.

The main task of our algorithm is to compute this row rank profile. Adapting ideas in the algorithms of [16, 35, 19], one would iteratively consider the rows of the matrix, looking for a linear relation with the previous rows by Gaussian elimination. When such a linear relation is found, the corresponding row can be discarded. Now, the multi-Krylov structure further permits to discard all the rows that correspond to monomial multiples of the leading term of the discovered syzygy, even before computing these rows. At some point, the set of rows to be considered is exhausted, and we can deduce the row rank profile.

In this approach, a row of the multi-Krylov matrix is computed by multiplying one of the already computed rows by one of the multiplication matrices. This results in many vector-matrix products, with possibly different matrices each time: this is an obstacle towards incorporating fast matrix multiplication. We circumvent this by introducing the variables one after another, thus seemingly not respecting the order of the rows specified by the monomial order; yet, we will manage to ensure that this order is respected in the end. When dealing with one variable X_k, we process successive powers $\boldsymbol{M}_k^{2^e}$ in the style of Keller-Gehrig's algorithm [28], using a logarithmic number of iterations.

Finally, from the monomial basis, one can easily find the minimal generating set of the leading module of $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$. The union of the monomial basis and of these generators is a set of monomials, which thus corresponds to a submatrix of the multi-Krylov matrix; the left nullspace of this submatrix, computed in reduced row echelon form, yields the reduced $\prec$-Gröbner basis of syzygies.

Previous work. An immediate remark is that the number of field entries of the multi-Krylov matrix is $m\beta_1 \cdots \beta_r D \in O(mD^{r+1})$, which significantly exceeds our target cost. Exploiting the structure of this matrix is therefore a common thread in all efficient algorithms.

For the univariate case $r = 1$, first algorithms with cost quadratic in D were given in [45, 42] for Hermite-Padé approximation (Example 1.2); they returned a single syzygy of small degree. Later, work on this case [48, 5] showed how to compute a basis of the module of syzygies in time $O(mD^2)$ and the more general M-Padé approximation was handled in the same complexity in [49, 4, 6, 7], with the algorithm of [7] returning the reduced $\prec$-Gröbner basis (called the shifted Popov form in that context) of syzygies for an arbitrary monomial order $\prec$. Then a cost quasi-linear in D was achieved for M-Padé approximation by a divide and conquer approach based on fast polynomial matrix multiplication [5, 21, 24], with the most recent algorithms returning the reduced $\prec$-Gröbner basis of syzygies for an arbitrary $\prec$ at such a cost [23, 25]. M-Padé approximation exactly corresponds to instances of Problem 1 with $r = 1$ and a multiplication matrix in Jordan normal form (which is further nilpotent for Hermite-Padé approximation) [24]. For achieving such costs, that are better than quadratic in D, it is necessary that the multiplication matrix exhibits such a structure and that the algorithm takes advantage of it, since in general merely representing the input multiplication matrix already requires D^2 field elements.

Still for $r = 1$, the case of an upper triangular multiplication matrix $\boldsymbol{M}_1$ was handled in [7, Algo. FFFG] by relying on the kind of linearization discussed above. This algorithm exploits the triangular shape to iterate on the D leading principal submatrices of $\boldsymbol{M}_1$, which extends the iteration for M-Padé approximation in [49, 4, 6] and costs $O(mD^2 + D^3)$ operations (see [38, Prop. 6.5] for a complexity analysis). To take advantage of fast matrix multiplication, another approach was designed in [24, Sec. 7], considering the same Krylov matrix as in [7] but processing its structure in the style of Keller-Gehrig's algorithm [28]. The algorithm in [24] supports an arbitrary matrix $\boldsymbol{M}_1$ at the cost $O(mD^{\omega-1} + D^\omega \log(d))$, and the algorithm in this paper can be seen as a generalization of it to the multivariate case since both coincide when $r = 1$ (up to the conversion between shifted Popov forms and reduced $\prec$-Gröbner bases). This generalization is not straightforward: besides the obvious fact that the output basis usually has more than m elements because most submodules of $\mathbb{K}[\boldsymbol{X}]^m$ are not free (unlike in the univariate case), as highlighted above the multivariate case also involves a more complex management of the order in which rows in the multi-Krylov matrix are inserted and processed, in relation with the monomial order $\prec$ and the successive introductions of the different variables.

On the other hand, previous algorithms dealing with the case $r \geqslant 1$ were developed independently of this line of work, starting from Möller and Buchberger's algorithm [36] and Faugère *et al.*'s FGLM algorithm [16]. The former computes the ideal of a finite set of points (Example 1.4), while the latter is specialized to the change of monomial order for ideals (Example 1.3), with a cost bound in $O(rD^3)$. Note however that the input in the FGLM algorithm is not the same as in Problem 1; this is discussed below.

A first generalization of [36, 16] was presented in [35, Algo. 1], still in the case $m = 1$ and $f_1 = 1$: the input describes $\mathcal{M} = \mathbb{K}[\boldsymbol{X}]/\mathcal{I}$ for some zero-dimensional ideal $\mathcal{I}$ of degree D, and the algorithm outputs a $\prec$-Gröbner basis of $\mathcal{I}$. The cost bound for this algorithm involves a term $O(rD^3)$, but also a term related to the description of the input. This input description is different from ours: it consists of a set of linear functionals which defines the ideal $\mathcal{I}$; thus one should be careful when comparing this work to our results. Another related extension of [36] is the Buchberger-Möller algorithm for matrices given in [27, Sec. 4.1.2], which solves Example 1.5; the runtime is not specified in that reference.

Another type of generalizations of [36] was detailed in [35, Algo. 2], [19, Algo. 4.7], and [39, Algo. 3.2], with the last reference tackling syzygy modules with $m \geqslant 1$ and arbitrary $f_1, \ldots, f_m$ like in this paper, yet with assumptions on $\mathcal{M}$; the cost bounds given in [35] and [19] involve a term in $O(rD^3)$ whereas [39] does not report on complexity bounds. The assumptions on $\mathcal{M}$ are

specified in the input of [35, Algo. 2], in [19, Eqn. (4.1)], and in [39, Eqn. (5)], and they imply that one can solve the problem by finding iteratively Gröbner bases for a sequence of "approximating" modules of syzygies which decrease towards the actual solution. Such assumptions lead to instances of Problem 1 which generalize to $r \geqslant 1$ the above-mentioned cases of M-Padé approximation and of a triangular multiplication matrix when $r = 1$; [39, Sec. 5] explicitly mentions the link with Beckermann and Labahn's algorithm for M-Padé approximation [6, 7]. In both the univariate and multivariate settings, it seems that such an iterative approach cannot be applied to the general case of Problem 1, where the input module has no other property than being finite-dimensional, and is described through arbitrary commuting matrices.

For the particular case of the change of order for ideals (Example 1.3), so with $m = 1$, when the target order is the lexicographic order $\prec_{\text{lex}}$, and under the assumption that the ideal is in Shape Position, fast matrix multiplication was used for the first time in [15], yielding a sub-cubic complexity. Indeed, if for example X_r is the smallest variable, these assumptions ensure that only $\boldsymbol{M}_r$ is needed; with this matrix as input, [15, Prop. 3] gives a probabilistic algorithm to compute the $\prec_{\text{lex}}$-Gröbner basis of syzygies within the cost bound $O(D^\omega \log(D) + r\mathsf{M}(D)\log(D))$. Besides ideas from [17, 18], this uses repeating squaring as in [28]. In this paper, we manage to incorporate fast matrix multiplication without assumption on the module, and for an arbitrary order.

Still for the particular of Example 1.3, Faugère and Mou give in [17, 18] probabilistic algorithms based on sparse linear algebra. These papers do not consider the computation of the multiplication matrices, which are assumed to be known. While we do not make any sparsity assumption on the multiplication matrices, for the sake of comparison we still summarize this approach below. Noticing that the multiplication matrices arising from the context of polynomial system solving are often sparse, [17, 18] tackle Problem 1 from a point of view similar to the Wiedemann algorithm. Evaluating the monomials in $\mathbb{K}[\boldsymbol{X}]/\mathcal{I}$ at certain linear functionals allows one to build a multi-dimensional recurrent sequence which admits $\mathcal{I}$ as its ideal of syzygies (this is only true for some type of ideals $\mathcal{I}$). In terms of the multi-Krylov matrices we are considering in Section 3 concerning Problem 1, this is similar to introducing an additional projection on the right of the multiplication matrices to take advantage of the sparsity by using a black-box point of view. Then recovering a $\prec_{\text{lex}}$-Gröbner basis of this ideal of syzygies can be done via the Berlekamp-Massey-Sakata algorithm [44], or the recent improvements in [8, 9, 10, 11].

Application: change of order. The FGLM algorithm [16] solves the change of order problem for Gröbner bases of ideals in $O(rD^3)$ operations for arbitrary orders $\prec_1$ and $\prec_2$: starting *only* from a $\prec_1$-Gröbner basis $\mathcal{G}_1$ for the input order $\prec_1$, it computes the $\prec_2$-Gröbner basis $\mathcal{G}_2$ of the ideal $\mathcal{I} = \langle \mathcal{G}_1 \rangle$. Following [17, Sec. 2.1], one can view the algorithm as a two-step process: it first computes from $\mathcal{G}_1$ the multiplication matrices of $\mathcal{M} = \mathbb{K}[\boldsymbol{X}]/\mathcal{I}$ with respect to the $\prec_1$-monomial basis, and then finds $\mathcal{G}_2$ as a set of $\mathbb{K}$-linear relations between certain normal forms modulo $\mathcal{G}_1$. The algorithm extends to the case of submodules of $\mathbb{K}[\boldsymbol{X}]^n$ for $n \geqslant 1$ (see e.g. [19, Sec. 2]).

Our algorithm for Problem 1 incorporates fast linear algebra into the second step, so once the multiplication matrices are known, one can find the reduced $\prec_2$-Gröbner basis in $O(nD^{\omega-1} + rD^\omega \log(D))$ operations. We now discuss how fast linear algebra may be incorporated into the computation of the multiplication matrices (Problem 2).

Our solution to this problem finds its roots in [15, Sec. 4], which focuses on the case where $n = 1$ and $\mathcal{I} = \langle f_1, \ldots, f_s \rangle$ is an ideal in $\mathbb{K}[\boldsymbol{X}]$. In the context studied in this reference only the matrix $\boldsymbol{M}_r$ of the smallest variable X_r is needed, and it is showed that this matrix can be simply read off from the input Gröbner basis without arithmetic operations, under some structural

Problem 2 – *Computing the multiplication matrices*

Input:

- a monomial order $\prec$ on $\mathbb{K}[\boldsymbol{X}]^n$,
- a reduced $\prec$-Gröbner basis $\{f_1, \ldots, f_s\} \subset \mathbb{K}[\boldsymbol{X}]^n$ such that $\mathcal{M} = \mathbb{K}[\boldsymbol{X}]^n/\langle f_1, \ldots, f_s\rangle$ has finite dimension as a $\mathbb{K}$-vector space.

Output: the multiplication matrices $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_r$ of $X_1, \ldots, X_r$ in $\mathcal{M} = \mathbb{K}[\boldsymbol{X}]^n/\langle f_1, \ldots, f_s\rangle$ with respect to its $\prec$-monomial basis.

assumption on the ideal of leading terms of $\mathcal{I}$ described in [15, Prop. 7]. Here we consider the more general case of submodules $\mathcal{N} = \langle f_1, \ldots, f_s\rangle$ of $\mathbb{K}[\boldsymbol{X}]^n$, and we design an algorithm which computes all multiplication matrices $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_r$ in $\mathcal{M} = \mathbb{K}[\boldsymbol{X}]^n/\mathcal{N}$ using $O(rD^\omega \log(D))$ operations, under a structural assumption on the module of leading terms of $\mathcal{N}$. Situations where this assumption on $\langle \mathrm{lt}_\prec(\mathcal{N})\rangle$ holds typically involve a monomial order such that $X_r\boldsymbol{c}_i \prec \cdots \prec X_1\boldsymbol{c}_i$ for $1 \leqslant i \leqslant n$. The assumption, described below, naturally extends the one from [15] to the case of submodules.

Definition 1.8. *For a monomial submodule $\mathcal{T} \subseteq \mathbb{K}[\boldsymbol{X}]^n$, the assumption $\mathcal{H}(\mathcal{T})$ is: "for all monomials $\mu \in \mathcal{T}$, for all $j \in \{1, \ldots, r\}$ such that X_j divides μ, for all $i \in \{1, \ldots, j-1\}$, the monomial $\frac{X_i}{X_j}\mu$ belongs to $\mathcal{T}$".*

In fact, instead of considering all monomials in $\mathcal{T}$, one can observe that $\mathcal{H}(\mathcal{T})$ holds if and only if the property holds for each monomial in the minimal generating set of $\mathcal{T}$ (see Lemma 2.2).

Theorem 1.9. *For $n \geqslant 1$, let $\prec$ be a monomial order on $\mathbb{K}[\boldsymbol{X}]^n$ and let $\{f_1, \ldots, f_s\}$ be a reduced $\prec$-Gröbner basis defining a submodule $\mathcal{N} = \langle f_1, \ldots, f_s\rangle$ of $\mathbb{K}[\boldsymbol{X}]^n$ such that $\mathbb{K}[\boldsymbol{X}]^n/\mathcal{N}$ has dimension D as a $\mathbb{K}$-vector space. Assuming $\mathcal{H}(\langle \mathrm{lt}_\prec(\mathcal{N})\rangle)$,*

- *Problem 2 can be solved using $O(rD^\omega \log(D))$ operations in $\mathbb{K}$;*
- *the change of order problem, that is, computing the reduced $\prec_2$-Gröbner basis of $\mathcal{N}$ for a monomial order $\prec_2$, can be solved using $O(nD^{\omega-1} + rD^\omega \log(D))$ operations in $\mathbb{K}$.*

Concerning the first item, an overview of our approach is presented in Section 4.1 and the detailed algorithms and proofs are in Sections 4.2 and 4.3, with a slightly refined cost bound in Proposition 4.7. The second item is proved in Section 4.4, based on Algorithm 7 which essentially calls our algorithms to compute first the multiplication matrices (Problem 2) and then the $\prec_2$-Gröbner basis of $\mathcal{N}$ by considering a specific module of syzygies (Problem 1).

Our structural assumption has been considered before, in particular in the case where $n = 1$ and we work modulo an ideal $\mathcal{I} = \langle f_1, \ldots, f_s\rangle$ (and the monomial order is such that $X_r \prec \cdots \prec X_1$, which is always true up to renaming the variables). In this case, it holds assuming for instance that the coefficients of $f_1, \ldots, f_s$ are generic (that is, pairwise distinct indeterminates over a given ground field) and that the Moreno-Socias conjecture holds [15, Sec. 4.1]. Another important situation where the assumption holds is when the leading ideal $\langle \mathrm{lt}_\prec(\mathcal{I})\rangle$ is *Borel-fixed* and the characteristic of $\mathbb{K}$ is zero, see [13, Sec. 15.9] and [15, Sec. 4.2]. A theorem first proved by Galligo in power series rings [20], then by Bayer and Stillman [3, Prop. 1] for a homogeneous ideal $\mathcal{I}$ in $\mathbb{K}[\boldsymbol{X}]$ shows that after a generic change of coordinates, $\langle \mathrm{lt}_\prec(\mathcal{I})\rangle$ is Borel-fixed.

The most general version of this result we are aware of is due to Pardue [41]. It applies to $\mathbb{K}[\boldsymbol{X}]$-submodules $\mathcal{N} \subset \mathbb{K}[\boldsymbol{X}]^n$, for certain monomial orders $\prec$ on $\mathbb{K}[\boldsymbol{X}]^n$; the precise conditions on $\prec$ are too technical to be stated here, but they hold in particular for the term over position order induced by a monomial order on $\mathbb{K}[\boldsymbol{X}]$ which refines the (weighted) total degree. In such cases, Pardue shows that after a generic linear change of variables, $\langle \mathrm{lt}_{\prec}(\mathcal{N})\rangle$ satisfies a Borel-fixedness property on $\mathbb{K}[\boldsymbol{X}]^n$ which implies that $\mathcal{H}(\langle \mathrm{lt}_{\prec}(\mathcal{N})\rangle)$ holds, at least in characteristic zero.

For polynomial system solving, with $n = 1$, an interesting particular case of the change of order problem is that of $\prec_1 = \prec_{\mathrm{drl}}$ being the degree reverse lexicographic order and $\prec_2 = \prec_{\mathrm{lex}}$ being the lexicographic order (so that in characteristic zero, Pardue's result shows that in generic coordinates, our structural assumption holds for such inputs). Fast algorithms for this case have been studied in [15, 14]. The former assumes the ideal $\mathcal{I}$ is in Shape Position, whereas this assumption is not needed here. In [14], an algorithm is designed to compute the multiplication matrices from a $\prec_{\mathrm{drl}}$-Gröbner basis in time $O(\beta r^{\omega} D^{\omega})$, where β is the maximum total degree of the elements of the input Gröbner basis. This is obtained by iterating over the total degree: the normal forms of all monomials of the same degree are dealt with using only one call to Gaussian elimination. While this does not require an assumption on the leading ideal, it is unclear to us how to remove the dependency in β in general.

Outline. Section 2 gathers preliminary material used in the rest of the paper: some notation, as well as basic definitions and properties related to monomial orders, Gröbner bases, and monomial staircases. Then Section 3 gives algorithms and proofs concerning the computation of bases of syzygies, leading to the main result of this paper (Theorem 1.7), while Section 4 focuses on the computation of the multiplication matrices (Theorem 1.9); both sections are introduced with a more detailed outline of their content.

2. Notations and definitions

Monomial orders and Gröbner bases for modules. Hereafter, we consider a multivariate polynomial ring $\mathbb{K}[\boldsymbol{X}] = \mathbb{K}[X_1, \ldots, X_r]$, for some field $\mathbb{K}$. Recall that the coordinate vectors are denoted by $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_m$, that is,

$$\boldsymbol{c}_j = (0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{K}[\boldsymbol{X}]^m \text{ with 1 at index } j.$$

A monomial in $\mathbb{K}[\boldsymbol{X}]$ is defined from exponents $\mathbf{e} = (e_1, \ldots, e_r) \in \mathbb{N}^r$ as $\boldsymbol{X}^{\mathbf{e}} = X_1^{e_1} \cdots X_r^{e_r}$ and a monomial in $\mathbb{K}[\boldsymbol{X}]^m$ is $\mu \boldsymbol{c}_j = (0, \ldots, 0, \mu, 0, \ldots, 0)$ with $1 \leqslant j \leqslant m$ and where μ is any monomial of $\mathbb{K}[\boldsymbol{X}]$. A term in $\mathbb{K}[\boldsymbol{X}]$ or in $\mathbb{K}[\boldsymbol{X}]^m$ is a monomial multiplied by a nonzero scalar from $\mathbb{K}$. Given terms μ and ν in $\mathbb{K}[\boldsymbol{X}]$, we say that the term $\mu \boldsymbol{c}_j$ is divisible by the term $\nu \boldsymbol{c}_k$ if $j = k$ and μ is divisible by ν in $\mathbb{K}[\boldsymbol{X}]$.

A submodule of $\mathbb{K}[\boldsymbol{X}]^m$ generated by monomials of $\mathbb{K}[\boldsymbol{X}]^m$ is called a monomial submodule. A submodule of $\mathbb{K}[\boldsymbol{X}]^m$ generated by homogeneous polynomials of $\mathbb{K}[\boldsymbol{X}]^m$ is called a homogeneous submodule.

Following [13, Sec. 15.3], a monomial order on $\mathbb{K}[\boldsymbol{X}]^m$ is a total order $\prec$ on the monomials of $\mathbb{K}[\boldsymbol{X}]^m$ such that, for any monomials μ, ν of $\mathbb{K}[\boldsymbol{X}]^m$ and any monomial κ of $\mathbb{K}[\boldsymbol{X}]$, $\mu \prec \nu$ implies $\mu \prec \kappa\mu \prec \kappa\nu$. Examples of common monomial orders on $\mathbb{K}[\boldsymbol{X}]^m$ are so-called *term over position* (top) and *position over term* (pot). In both cases, we start from a monomial order on $\mathbb{K}[\boldsymbol{X}]$ written $\prec$. Then, given monomials $\mu \boldsymbol{c}_i$ and $\nu \boldsymbol{c}_j$, we say that $\mu \boldsymbol{c}_i \prec^{\mathrm{top}} \nu \boldsymbol{c}_j$ if $\mu \prec \nu$ or if $\mu = \nu$ and $i < j$. Similarly, we say that $\mu \boldsymbol{c}_i \prec^{\mathrm{pot}} \nu \boldsymbol{c}_j$ if $i < j$ or if $i = j$ and $\mu \prec \nu$.

For a given monomial order $\prec$ on $\mathbb{K}[\boldsymbol{X}]^m$ and an element $f \in \mathbb{K}[\boldsymbol{X}]^m$, the $\prec$-leading term of f, denoted by $\mathrm{lt}_\prec(f)$, is the term of f whose monomial is the greatest with respect to $\prec$. This extends to any collection $\mathcal{F} \subseteq \mathbb{K}[\boldsymbol{X}]^m$ of polynomials: $\mathrm{lt}_\prec(\mathcal{F})$ is the set of leading terms $\{\mathrm{lt}_\prec(f) \mid f \in \mathcal{F}\}$ of the elements of $\mathcal{F}$. In particular, for a module $\mathcal{N}$ in $\mathbb{K}[\boldsymbol{X}]^m$, $\langle \mathrm{lt}_\prec(\mathcal{N})\rangle$ is a monomial submodule of $\mathbb{K}[\boldsymbol{X}]^m$ which is called the $\prec$-leading module of $\mathcal{N}$.

Definition 2.1 (Gröbner basis)**.** *Let $\prec$ be a monomial order on $\mathbb{K}[\boldsymbol{X}]^m$ and let $\mathcal{N}$ be a $\mathbb{K}[\boldsymbol{X}]$-submodule of $\mathbb{K}[\boldsymbol{X}]^m$. A subset $\{f_1, \ldots, f_s\} \subset \mathcal{N}$ is said to be a $\prec$-Gröbner basis of $\mathcal{N}$ if the $\prec$-leading module of $\mathcal{N}$ is generated by $\{\mathrm{lt}_\prec(f_1), \ldots, \mathrm{lt}_\prec(f_s)\}$, i.e. $\langle \mathrm{lt}_\prec(\mathcal{N})\rangle = \langle \mathrm{lt}_\prec(f_1), \ldots, \mathrm{lt}_\prec(f_s)\rangle$.*

There is a specific $\prec$-Gröbner basis of $\mathcal{N}$, called the reduced $\prec$-Gröbner basis of $\mathcal{N}$, which is uniquely defined in terms of the module $\mathcal{N}$ and the monomial order $\prec$. Namely, this is the Gröbner basis $\{f_1, \ldots, f_s\}$ of $\mathcal{N}$ such that for $1 \leqslant i \leqslant s$, $\mathrm{lt}_\prec(f_i)$ is monic and does not divide any term of f_j for $j \neq i$.

Monomial basis and staircase monomials. In what follows, the submodules $\mathcal{N} \subseteq \mathbb{K}[\boldsymbol{X}]^m$ we consider are such that the quotient $\mathbb{K}[\boldsymbol{X}]^m/\mathcal{N}$ has finite dimension as a $\mathbb{K}$-vector space. We will often use its basis formed by the monomials not in $\mathrm{lt}_\prec(\mathcal{N})$ [13, Thm. 15.3]; this basis is denoted by $\mathcal{E} = (\varepsilon_1, \ldots, \varepsilon_D)$ and called the *$\prec$-monomial basis* of $\mathbb{K}[\boldsymbol{X}]^m/\mathcal{N}$. Any polynomial $f \in \mathbb{K}[\boldsymbol{X}]^m$ can be uniquely written $f = g + h$, where $g \in \mathcal{N}$ and $h \in \mathbb{K}[\boldsymbol{X}]^m$ is a $\mathbb{K}$-linear combination of the monomials in $\mathcal{E}$; this polynomial h is called the *$\prec$-normal form of f (with respect to $\mathcal{N}$)* and is denoted by $\mathrm{nf}_\prec(f)$. We extend the notation to sets of polynomials $\mathcal{F} \subseteq \mathbb{K}[\boldsymbol{X}]^m$, that is, $\mathrm{nf}_\prec(\mathcal{F}) = \{\mathrm{nf}_\prec(f) \mid f \in \mathcal{F}\}$.

As in [35, Sec. 3] (which focuses on the case of ideals), we will use other sets of monomials related to this monomial basis. First, we consider the monomials obtained by multiplying those of $\mathcal{E}$ by a variable:

$$\mathcal{S} = \{X_k \varepsilon_j \mid 1 \leqslant k \leqslant r, 1 \leqslant j \leqslant D\} \cup \{\boldsymbol{c}_i \mid 1 \leqslant i \leqslant m \text{ such that } \boldsymbol{c}_i \notin \mathcal{E}\}.$$

This allows us to define the border $\mathcal{B} = \mathcal{S} - \mathcal{E}$, which is a set of monomial generators of $\langle \mathrm{lt}_\prec(\mathcal{N})\rangle$. Then the polynomials $\{\mu - \mathrm{nf}_\prec(\mu) \mid \mu \in \mathcal{B}\}$ form a canonical generating set of $\mathcal{N}$, called the $\prec$-border basis of $\mathcal{N}$ [34, 35]. Finally, we consider the minimal generating set $\mathcal{L}$ of $\langle \mathrm{lt}_\prec(\mathcal{N})\rangle$: it is a subset of $\mathcal{B}$ such that the reduced $\prec$-Gröbner basis of $\mathcal{N}$ is $\mathcal{G} = \{\mu - \mathrm{nf}_\prec(\mu) \mid \mu \in \mathcal{L}\}$. In particular, we have $\mathcal{L} = \mathrm{lt}_\prec(\mathcal{G}) = \{\mathrm{lt}_\prec(f) \mid f \in \mathcal{G}\}$.

By construction, $\mathrm{Card}(\mathcal{L}) = \mathrm{Card}(\mathcal{G}) \leqslant \mathrm{Card}(\mathcal{B}) \leqslant \mathrm{Card}(\mathcal{S})$. Above, $\mathcal{S}$ is defined as the union of a set of cardinality at most rD and a set of cardinality at most m, hence $\mathrm{Card}(\mathcal{S}) \leqslant rD + m$. Besides, since the coordinate vectors in the second set are in the minimal generating set $\mathcal{L}$ of $\langle \mathrm{lt}_\prec(\mathcal{N})\rangle$, we have $\mathrm{Card}(\mathcal{B} - \mathcal{L}) \leqslant rD$. Note that if the upper bound on $\mathrm{Card}(\mathcal{S})$ is an equality, then all coordinate vectors $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_m$ are in $\mathcal{L}$, which holds only in the case $\mathcal{N} = \mathbb{K}[\boldsymbol{X}]^m$ (in particular $\mathcal{E} = \emptyset$ and $D = 0$).

Finally, we give a characterization of the structural assumption of monomial submodules described in Definition 1.8, showing that one can focus on the monomials in the minimal generating set instead of all monomials in the module.

Lemma 2.2. *Let $\mathcal{T}$ be a monomial submodule of $\mathbb{K}[\boldsymbol{X}]^m$ and let $\{\mu_1, \ldots, \mu_s\}$ be the minimal generating set of $\mathcal{T}$. Then $\mathcal{H}(\mathcal{T})$ holds if and only if for all $k \in \{1, \ldots, s\}$, for all $j \in \{1, \ldots, r\}$ such that X_j divides μ_k, for all $i \in \{1, \ldots, j-1\}$, we have $\frac{X_i}{X_j}\mu_k \in \mathcal{T}$.*

Proof. Obviously, $\mathcal{H}(\mathcal{T})$ implies the latter property since each μ_k is a monomial in $\mathcal{T}$. Conversely, we assume that for all $k \in \{1, \dots, s\}$, for all $j \in \{1, \dots, r\}$ such that X_j divides μ_k, for all $i \in \{1, \dots, j-1\}$, we have $\frac{X_i}{X_j}\mu_k \in \mathcal{T}$, and we want to prove that $\mathcal{H}(\mathcal{T})$ holds. Let μ be a monomial in $\mathcal{T}$, let $j \in \{1, \dots, r\}$ be such that X_j divides μ, and let $i \in \{1, \dots, j-1\}$; we want to prove that $\frac{X_i}{X_j}\mu \in \mathcal{T}$. Since $\mathcal{T}$ is a monomial module, $\mu = \nu\mu_k$ for some monomial $\nu \in \mathbb{K}[\boldsymbol{X}]$ and $1 \leqslant k \leqslant s$. By assumption, X_j divides $\nu\mu_k$, thus either X_j divides ν or X_j divides μ_k. In the first case $\frac{X_i}{X_j}\mu = (\frac{X_i}{X_j}\nu)\mu_k \in \mathcal{T}$, and in the second case $\frac{X_i}{X_j}\mu_k \in \mathcal{T}$ by assumption and therefore $\frac{X_i}{X_j}\mu = \nu(\frac{X_i}{X_j}\mu_k) \in \mathcal{T}$. □

3. Computing bases of syzygies via linear algebra

In this section, we focus on Problem 1 and we prove Theorem 1.7. Thus, we are given pairwise commuting matrices $\boldsymbol{M} = (\boldsymbol{M}_1, \dots, \boldsymbol{M}_r)$ in $\mathbb{K}^{D\times D}$, a matrix $\boldsymbol{F} \in \mathbb{K}^{m\times D}$ and a monomial order $\prec$ on $\mathbb{K}[\boldsymbol{X}]^m = \mathbb{K}[X_1, \dots, X_r]^m$; we compute the reduced $\prec$-Gröbner basis of $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$.

The basic ingredient is a *linearization* of the problem, meaning that we will interpret all operations on polynomials as operations of $\mathbb{K}$-linear algebra. In Section 3.1, we show a correspondence between syzygies of bounded degree and vectors in the nullspace of a matrix over $\mathbb{K}$ which exhibits a structure that we call *multi-Krylov*. The multi-Krylov matrix is formed by multiplications of $\boldsymbol{F}$ by powers of the multiplications matrices; its rows are ordered according to the monomial order $\prec$ given as input of Problem 1.

Then, in Section 3.2, we show that the row rank profile of this multi-Krylov matrix exactly corresponds to the $\prec$-monomial basis of the quotient $\mathbb{K}[\boldsymbol{X}]^m / \mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$. To compute this row rank profile efficiently, we use in particular an idea from [28] which we extend to our context, while also both exploiting the structure of the multi-Krylov matrix to always work on a small subset of its rows and ensuring that the rows are considered in the right order. Finally, in Section 3.3, we exploit the knowledge of the $\prec$-monomial basis to compute the reduced $\prec$-Gröbner basis of syzygies.

3.1. Monomial basis as the rank profile of a multi-Krylov matrix

We first describe *expansion* and *contraction* operations, which convert polynomials of bounded degrees into their coefficient vectors and vice versa (the bound is written $\boldsymbol{\beta}$ below). It will be convenient to rely on the following indexing function.

Definition 3.1. *Let $\boldsymbol{\beta} = (\beta_1, \dots, \beta_r) \in \mathbb{N}^r_{>0}$ and let $\prec$ be a monomial order on $\mathbb{K}[\boldsymbol{X}]^m$. Then we define the* $(\prec, \boldsymbol{\beta})$-indexing function $\phi_{\prec,\boldsymbol{\beta}}$ *as the unique bijection*

$$\phi_{\prec,\boldsymbol{\beta}} : \{\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i \mid \mathbf{0} \leqslant \mathbf{e} < \boldsymbol{\beta}, 1 \leqslant i \leqslant m\} \to \{1, \dots, m\beta_1 \cdots \beta_r\}$$

which is increasing for $\prec$, that is, such that $\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i \prec \boldsymbol{X}^{\mathbf{e}'}\boldsymbol{c}_{i'}$ if and only if $\phi_{\prec,\boldsymbol{\beta}}(\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i) < \phi_{\prec,\boldsymbol{\beta}}(\boldsymbol{X}^{\mathbf{e}'}\boldsymbol{c}_{i'})$.

In other words, take the sequence of monomials $(\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i, \mathbf{0} \leqslant \mathbf{e} < \boldsymbol{\beta}, 1 \leqslant i \leqslant m)$ and sort it according to $\prec$; then $\phi_{\prec,\boldsymbol{\beta}}(\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i)$ is the index of $\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i$ in the sorted sequence (assuming indices start at 1).

Hereafter, $\mathbb{K}[\boldsymbol{X}]_{\prec\boldsymbol{\beta}}$ stands for the set of polynomials $p \in \mathbb{K}[\boldsymbol{X}]$ such that $\deg_{X_k}(p) < \beta_k$ for $1 \leqslant k \leqslant r$. This yields a $\mathbb{K}$-linear correspondence between bounded-degree polynomials and

vectors

$$\begin{array}{rccc} \mathcal{E}_{\prec,\boldsymbol{\beta}}: & \mathbb{K}[\boldsymbol{X}]^m_{<\boldsymbol{\beta}} & \to & \mathbb{K}^{1\times m\beta_1\cdots\beta_r} \\ \boldsymbol{p} = & \displaystyle\sum_{\substack{\boldsymbol{f}=\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i \\ \mathbf{0}\leqslant\mathbf{e}<\boldsymbol{\beta}, 1\leqslant i\leqslant m}} u_{\boldsymbol{f}}\,\boldsymbol{f} & \mapsto & \boldsymbol{v} = [u_{\phi^{-1}_{\prec,\boldsymbol{\beta}}(k)} \mid 1 \leqslant k \leqslant m\beta_1\cdots\beta_r] \end{array}$$

called *expansion*, with inverse $C_{\prec,\boldsymbol{\beta}}$ called *contraction*. For a polynomial $\boldsymbol{p} \in \mathbb{K}[\boldsymbol{X}]^m_{<\boldsymbol{\beta}}$, $\mathcal{E}_{\prec,\boldsymbol{\beta}}(\boldsymbol{p})$ is the vector in $\mathbb{K}^{1\times m\beta_1\cdots\beta_r}$ whose entry at index $\phi_{\prec,\boldsymbol{\beta}}(\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i)$ is the coefficient of the term involving $\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i$ in $\boldsymbol{p}$.

Example 3.2. Consider the case with $r = 2$ variables and $m = 2$, using the $\prec_{\text{lex}}$-term over position order $\prec^{\text{top}}_{\text{lex}}$ on $\mathbb{K}[X, Y]^2$, with $Y \prec_{\text{lex}} X$. Choosing the degree bounds $\boldsymbol{\beta} = (2, 3)$, the monomials

$$\{X^jY^k\boldsymbol{c}_i \mid \mathbf{0} \leqslant (j,k) < (2,3), 1 \leqslant i \leqslant 2\}$$

are indexed as follows, according to Definition 3.1:

Monomial $X^jY^k\boldsymbol{c}_i$	Index $\phi_{\prec^{\text{top}}_{\text{lex}},(2,3)}(X^jY^k\boldsymbol{c}_i)$
$\begin{bmatrix}1 & 0\end{bmatrix}$	1
$\begin{bmatrix}0 & 1\end{bmatrix}$	2
$\begin{bmatrix}Y & 0\end{bmatrix}$	3
$\begin{bmatrix}0 & Y\end{bmatrix}$	4
$\begin{bmatrix}Y^2 & 0\end{bmatrix}$	5
$\begin{bmatrix}0 & Y^2\end{bmatrix}$	6
$\begin{bmatrix}X & 0\end{bmatrix}$	7
$\begin{bmatrix}0 & X\end{bmatrix}$	8
$\begin{bmatrix}XY & 0\end{bmatrix}$	9
$\begin{bmatrix}0 & XY\end{bmatrix}$	10
$\begin{bmatrix}XY^2 & 0\end{bmatrix}$	11
$\begin{bmatrix}0 & XY^2\end{bmatrix}$	12

Let $\boldsymbol{p}$ be the polynomial in $\mathbb{K}[X, Y]^2_{<(2,3)}$ and $\boldsymbol{v}$ be the vector in $\mathbb{K}^{1\times 12}$ defined by

$$\begin{aligned} \boldsymbol{p} &= \begin{bmatrix}46 + 95Y + 75X + 10XY & 36 + 18Y + 38Y^2 + 77X + 83XY + 35XY^2\end{bmatrix}, \\ \boldsymbol{v} &= \begin{bmatrix}86 & 0 & 32 & 83 & 54 & 26 & 0 & 68 & 86 & 0 & 54 & 22\end{bmatrix}. \end{aligned}$$

In this case, the expansion of $\boldsymbol{p}$ and the contraction of $\boldsymbol{v}$ are given by

$$\begin{aligned} \mathcal{E}_{\prec,\boldsymbol{\beta}}(\boldsymbol{p}) &= \begin{bmatrix}46 & 36 & 95 & 18 & 0 & 38 & 75 & 77 & 10 & 83 & 0 & 35\end{bmatrix}, \\ C_{\prec,\boldsymbol{\beta}}(\boldsymbol{v}) &= \begin{bmatrix}86 + 32Y + 54Y^2 + 86XY + 54XY^2 & 83Y + 26Y^2 + 68X + 22XY^2\end{bmatrix}. \end{aligned}$$ □

Now, we detail the construction of the multi-Krylov matrix. Let $\boldsymbol{M} = (\boldsymbol{M}_1, \ldots, \boldsymbol{M}_r)$ be pairwise commuting matrices in $\mathbb{K}^{D\times D}$ that define a $\mathbb{K}[\boldsymbol{X}]$-module structure on $\mathbb{K}^{1\times D}$, and let

$\boldsymbol{F}$ be in $\mathbb{K}^{m\times D}$, with rows $\boldsymbol{f}_1, \dots, \boldsymbol{f}_m$. As mentioned in Definition 1.1, for a polynomial $\boldsymbol{p} = [p_1, \dots, p_m] \in \mathbb{K}[\boldsymbol{X}]^m$ we write

$$\boldsymbol{p} \cdot \boldsymbol{F} = p_1 \cdot \boldsymbol{f}_1 + \cdots + p_m \cdot \boldsymbol{f}_m = \boldsymbol{f}_1\, p_1(\boldsymbol{M}) + \cdots + \boldsymbol{f}_m\, p_m(\boldsymbol{M}).$$

As a result, a polynomial $\boldsymbol{p}$ is in $\operatorname{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$ if its coefficients form a $\mathbb{K}$-linear combination of vectors of the form $\boldsymbol{f}_i\boldsymbol{M}^{\mathbf{e}}$ which is zero. If furthermore $\boldsymbol{p}$ is nonzero and has its degrees in each variable bounded by $(\beta_1, \dots, \beta_r)$, then it corresponds to a nontrivial $\mathbb{K}$-linear relation between the row vectors

$$\{\boldsymbol{f}_i\boldsymbol{M}^{\mathbf{e}} \mid \mathbf{0} \leqslant \mathbf{e} < \boldsymbol{\beta}, 1 \leqslant i \leqslant m\}.$$

This leads us to consider the matrices formed by these vectors, ordered according to $\phi_{\prec,\boldsymbol{\beta}}$.

Definition 3.3. *Let $\prec$ be a monomial order on $\mathbb{K}[\boldsymbol{X}]^m$, let $\boldsymbol{M} = (\boldsymbol{M}_1, \dots, \boldsymbol{M}_r) \in \mathbb{K}^{D\times D}$ be pairwise commuting matrices, let $\boldsymbol{F} \in \mathbb{K}^{m\times D}$ whose rows are $\boldsymbol{f}_1, \dots, \boldsymbol{f}_m$, and let $\boldsymbol{\beta} = (\beta_1, \dots, \beta_r) \in \mathbb{N}^r_{>0}$. The $(\prec, \boldsymbol{\beta})$-*multi-Krylov matrix for $(\boldsymbol{M}, \boldsymbol{F})$, *denoted by $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$, is the matrix in $\mathbb{K}^{m\beta_1\cdots\beta_r\times D}$ whose row at index $\phi_{\prec,\boldsymbol{\beta}}(\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i)$ is $\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i \cdot \boldsymbol{F} = \boldsymbol{f}_i\boldsymbol{M}^{\mathbf{e}}$, for $\mathbf{0} \leqslant \mathbf{e} < \boldsymbol{\beta}$ and $1 \leqslant i \leqslant m$.*

Example 3.4. Following on from Example 3.2, we consider the vector space dimension $D = 3$ and matrices $\boldsymbol{F}$ in $\mathbb{K}^{2\times 3}$ and $\boldsymbol{M} = (\boldsymbol{M}_X, \boldsymbol{M}_Y)$ in $\mathbb{K}^{3\times 3}$ such that $\boldsymbol{M}_X\boldsymbol{M}_Y = \boldsymbol{M}_Y\boldsymbol{M}_X$. Then from the indexing function $\phi_{\prec^{\text{top}}_{\text{lex}},(2,3)}$ described above we obtain

$$\mathcal{K}_{\prec^{\text{top}}_{\text{lex}},(2,3)}(\boldsymbol{M}, \boldsymbol{F}) = \begin{bmatrix} \boldsymbol{F} \\ \boldsymbol{F}\boldsymbol{M}_Y \\ \boldsymbol{F}\boldsymbol{M}_Y^2 \\ \boldsymbol{F}\boldsymbol{M}_X \\ \boldsymbol{F}\boldsymbol{M}_X\boldsymbol{M}_Y \\ \boldsymbol{F}\boldsymbol{M}_X\boldsymbol{M}_Y^2 \end{bmatrix} \in \mathbb{K}^{12\times 3}. \qquad \square$$

By construction, we have the following result, which relates the left nullspace of the multi-Krylov matrix with the set of bounded-degree syzygies.

Lemma 3.5. *If $\boldsymbol{v} \in \mathbb{K}^{1\times m\beta_1\cdots\beta_r}$ is in the left nullspace of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$, then its contraction $\mathcal{C}_{\prec,\boldsymbol{\beta}}(\boldsymbol{v}) \in \mathbb{K}[\boldsymbol{X}]^m_{\prec\boldsymbol{\beta}}$ is in $\operatorname{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$. Conversely, if $\boldsymbol{p} \in \mathbb{K}[\boldsymbol{X}]^m_{\prec\boldsymbol{\beta}}$ is in $\operatorname{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$, then its expansion $\mathcal{E}_{\prec,\boldsymbol{\beta}}(\boldsymbol{p}) \in \mathbb{K}^{1\times m\beta_1\cdots\beta_r}$ is in the left nullspace of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$.*

Our first step towards finding a $\prec$-Gröbner basis $\mathcal{G}$ of $\operatorname{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$ consists in computing the $\prec$-monomial basis $\mathcal{E}$ of the quotient $\mathbb{K}[\boldsymbol{X}]^m / \operatorname{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$; we are going to prove that it corresponds to the row rank profile of the multi-Krylov matrix. From the above discussion, we know that considering this matrix only gives us access to syzygies which satisfy degree constraints. In what follows, we will choose $\boldsymbol{\beta} = (\beta_1, \dots, \beta_r)$, with $\beta_i \geqslant D$ for all i. In particular, for this choice, all elements in the monomial basis of $\mathbb{K}[\boldsymbol{X}]^m / \operatorname{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$ are in $\mathbb{K}[\boldsymbol{X}]^m_{\prec\boldsymbol{\beta}}$.

We recall that, for a matrix $\boldsymbol{A}$ in $\mathbb{K}^{\mu\times\nu}$, the *row rank profile* of $\boldsymbol{A}$ is the lexicographically smallest subtuple $(\rho_1, \dots, \rho_\Delta)$ of $(1, \dots, \mu)$ such that Δ is the rank of $\boldsymbol{A}$ and the rows $(\rho_1, \dots, \rho_\Delta)$ of $\boldsymbol{A}$ are linearly independent.

Theorem 3.6. *Let $\prec$ be a monomial order on $\mathbb{K}[\boldsymbol{X}]^m$, let $\boldsymbol{M} = (\boldsymbol{M}_1, \dots, \boldsymbol{M}_r)$ be pairwise commuting matrices in $\mathbb{K}^{D\times D}$, let $\boldsymbol{F} \in \mathbb{K}^{m\times D}$, and let $\boldsymbol{\beta} = (\beta_1, \dots, \beta_r)$, with $\beta_i \geqslant D$ for all i. Let further $(\rho_1, \dots, \rho_\Delta) \in \mathbb{N}^\Delta_{>0}$ be the row rank profile of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$. Then the $\prec$-monomial basis $\mathcal{E}$ of $\mathbb{K}[\boldsymbol{X}]^m / \operatorname{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$ is equal to $\{\phi^{-1}_{\prec,\boldsymbol{\beta}}(\rho_1), \dots, \phi^{-1}_{\prec,\boldsymbol{\beta}}(\rho_\Delta)\}$.*

Proof. Write $\rho_j = \phi_{\prec,\boldsymbol{\beta}}(\boldsymbol{X}^{\mathbf{e}_j}\boldsymbol{c}_{i_j})$, so that $\{\phi_{\prec,\boldsymbol{\beta}}^{-1}(\rho_1), \ldots, \phi_{\prec,\boldsymbol{\beta}}^{-1}(\rho_\Delta)\} = \{\boldsymbol{X}^{\mathbf{e}_1}\boldsymbol{c}_{i_1}, \ldots, \boldsymbol{X}^{\mathbf{e}_\Delta}\boldsymbol{c}_{i_\Delta}\}$. We want to prove that $\mathrm{lt}_\prec(\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F}))$ is the set of monomials not in $\{\boldsymbol{X}^{\mathbf{e}_1}\boldsymbol{c}_{i_1}, \ldots, \boldsymbol{X}^{\mathbf{e}_\Delta}\boldsymbol{c}_{i_\Delta}\}$.

First, consider any monomial $\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i$ for $1 \leqslant i \leqslant m$ and $\mathbf{e} \in \mathbb{N}^r$ such that $\mathbf{e} \not\prec \boldsymbol{\beta}$. Such a monomial cannot be in $\{\boldsymbol{X}^{\mathbf{e}_1}\boldsymbol{c}_{i_1}, \ldots, \boldsymbol{X}^{\mathbf{e}_\Delta}\boldsymbol{c}_{i_\Delta}\}$ since by construction of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$ we have $\mathbf{e}_j < \boldsymbol{\beta}$ for all j. On the other hand, $\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i$ cannot be in the $\prec$-monomial basis $\mathcal{E}$ either. Indeed, writing $\boldsymbol{X}^{\mathbf{e}} = X_1^{e_1} \cdots X_r^{e_r}$, $\mathbf{e} \not\prec \boldsymbol{\beta}$ means that $e_k \geqslant \beta_k \geqslant D$ for some index k; if $\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i$ is in $\mathcal{E}$ then $X_k^{e_k}\boldsymbol{c}_i$ is also in $\mathcal{E}$ and thus $\boldsymbol{c}_i, X_k\boldsymbol{c}_i, \ldots, X_k^D\boldsymbol{c}_i$ are in $\mathcal{E}$, which is a contradiction since linear independence would imply that the minimal polynomial of $\boldsymbol{M}_k$ has degree greater than D. Hence $\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i \in \mathrm{lt}_\prec(\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F}))$.

Now, let $\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i \in \mathrm{lt}_\prec(\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F}))$ be such that $\mathbf{e} < \boldsymbol{\beta}$. Then there is a polynomial $\boldsymbol{p}$ in $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$ such that $\mathrm{lt}_\prec(\boldsymbol{p}) = \boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i$, and Lemma 3.5 implies that $\mathcal{E}_{\prec,\boldsymbol{\beta}}(\boldsymbol{p})$ is in the left nullspace of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$. Since by construction the rightmost nonzero entry of $\mathcal{E}_{\prec,\boldsymbol{\beta}}(\boldsymbol{p})$ is 1 at index $\phi_{\prec,\boldsymbol{\beta}}(\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i)$, the vector $\mathcal{E}_{\prec,\boldsymbol{\beta}}(\boldsymbol{p})$ expresses the row of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$ with index $\phi_{\prec,\boldsymbol{\beta}}(\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i)$ as a $\mathbb{K}$-linear combination of the rows with smaller indices. By definition of the row rank profile, this implies $\phi_{\prec,\boldsymbol{\beta}}(\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i) \notin \{\rho_1, \ldots, \rho_\Delta\}$, and therefore $\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i \notin \{\boldsymbol{X}^{\mathbf{e}_1}\boldsymbol{c}_{i_1}, \ldots, \boldsymbol{X}^{\mathbf{e}_\Delta}\boldsymbol{c}_{i_\Delta}\}$.

Conversely, let $\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i \notin \{\boldsymbol{X}^{\mathbf{e}_1}\boldsymbol{c}_{i_1}, \ldots, \boldsymbol{X}^{\mathbf{e}_\Delta}\boldsymbol{c}_{i_\Delta}\}$ be a monomial such that $\mathbf{e} < \boldsymbol{\beta}$. Then $\phi_{\prec,\boldsymbol{\beta}}(\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i) \notin \{\rho_1, \ldots, \rho_\Delta\}$. Thus, by definition of the row rank profile, there is a nonzero vector $\boldsymbol{v} \in \mathbb{K}^{1\times mD^r}$ such that $\boldsymbol{v}$ is in the left nullspace of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$ and the rightmost nonzero entry of $\boldsymbol{v}$ is 1 at index $\phi_{\prec,\boldsymbol{\beta}}(\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i)$. Then $\mathrm{lt}_\prec(C_{\prec,\boldsymbol{\beta}}(\boldsymbol{v})) = \boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i$, and according to Lemma 3.5, $C_{\prec,\boldsymbol{\beta}}(\boldsymbol{v})$ is in $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$, hence $\mathrm{lt}_\prec(C_{\prec,\boldsymbol{\beta}}(\boldsymbol{v})) \in \mathrm{lt}_\prec(\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F}))$. □

In particular, we see that the dimension Δ of $\mathbb{K}[\boldsymbol{X}]^m / \mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$ as a $\mathbb{K}$-vector space is equal to the rank of the multi-Krylov matrix $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$. In particular this implies $\Delta \leqslant D$, since $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$ has D columns.

3.2. Computing the monomial basis

We now show how to exploit the structure of the multi-Krylov matrix so as to efficiently compute its row rank profile, yielding the $\prec$-monomial basis $\mathcal{E}$ of $\mathbb{K}[\boldsymbol{X}]^m / \mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$.

For $\boldsymbol{\beta}$ as in Theorem 3.6, the dense representation of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$ uses at least mD^{r+1} field elements, which is well beyond our target cost $\tilde{O}(mD^{\omega-1} + rD^\omega)$. On the other hand, this matrix is succinctly described by the data $(\prec, \boldsymbol{M}, \boldsymbol{F})$, which requires $O(mD + rD^2)$ field elements. Like previous related algorithms [36, 16, 35], we will never compute the full dense representation of this matrix, but rather always store and use a minimal amount of data that allows the algorithm to progress. The main property behind this is that once some monomial is found not to be in the sought monomial basis, then all multiples of that monomial can be discarded from the rest of the computation.

Our algorithm for computing the monomial basis $\mathcal{E} = (\varepsilon_1, \ldots, \varepsilon_\Delta)$ also returns the matrix $\boldsymbol{B} \in \mathbb{K}^{\Delta\times D}$ whose rows are $\varepsilon_1 \cdot \boldsymbol{F}, \ldots, \varepsilon_\Delta \cdot \boldsymbol{F}$; it will be needed later on. The algorithm exploits the structure of this matrix, and uses fast arithmetic for matrices over $\mathbb{K}$ through

- a procedure RowRankProfile which computes the row rank profile of any matrix in $\mathbb{K}^{\mu\times\nu}$ of rank ρ in $O(\rho^{\omega-2}\mu\nu)$ operations in $\mathbb{K}$, as described in [46, Sec. 2.2];
- matrix multiplication which is incorporated by following a strategy in the style of Keller-Gehrig [28].

In short, the latter strategy can be thought of as precomputing powers of the form $\boldsymbol{M}_j^{2^e}$ of the multiplication matrices, which then allows us to group many vector-matrix products into a small number of matrix-matrix products. To achieve this we work iteratively on the variables, thus first focusing on all operations involving $\boldsymbol{M}_1$, then on those involving $\boldsymbol{M}_2$, etc. The order of

the rows specified by $\prec$ is not respected in this process, since at a fixed stage of the algorithm we will only have considered a submatrix of the multi-Krylov matrix which does not involve the last variables. To fix this we constantly re-order, according to $\prec$, the rows that have been processed and the ones that we introduce.

Algorithm 1 – MonomialBasis

Input:

- monomial order $\prec$ on $\mathbb{K}[\boldsymbol{X}]^m = \mathbb{K}[X_1, \ldots, X_r]^m$,
- pairwise commuting matrices $\boldsymbol{M} = (\boldsymbol{M}_1, \ldots, \boldsymbol{M}_r)$ in $\mathbb{K}^{D\times D}$,
- matrix $\boldsymbol{F} \in \mathbb{K}^{m\times D}$.

Output:

- the $\prec$-monomial basis $\mathcal{E} = (\varepsilon_1, \ldots, \varepsilon_\Delta)$ of $\mathbb{K}[\boldsymbol{X}]^m / \operatorname{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$,
- the matrix $\boldsymbol{B} \in \mathbb{K}^{\Delta\times D}$ whose rows are $\varepsilon_1 \cdot \boldsymbol{F}, \ldots, \varepsilon_\Delta \cdot \boldsymbol{F}$.

1. $\beta \leftarrow 2^{\lceil \log_2(D)\rceil+1}$; $\boldsymbol{\beta} \leftarrow (\beta, \ldots, \beta) \in \mathbb{N}^r_{>0}$

2. $\phi_{\prec,\boldsymbol{\beta}} \leftarrow$ the indexing function in Definition 3.1

3. $\pi \leftarrow$ the permutation matrix in $\{0,1\}^{m\times m}$ such that the tuple $\mathbf{t} = \pi[\phi_{\prec,\boldsymbol{\beta}}(\boldsymbol{c}_1), \ldots, \phi_{\prec,\boldsymbol{\beta}}(\boldsymbol{c}_m)]^\mathsf{T}$ is increasing

4. $\boldsymbol{B} \leftarrow \pi\boldsymbol{F}$

5. $\hat{\delta}, (i_1, \ldots, i_{\hat{\delta}}) \leftarrow$ RowRankProfile($\boldsymbol{B}$)

6. For k from 1 to r `// iterate over the variables`

a. $\boldsymbol{P} \leftarrow \boldsymbol{M}_k$; $e \leftarrow 0$

b. Do

(i) $\delta \leftarrow \hat{\delta}$

(ii) $(\rho_1, \ldots, \rho_\delta) \leftarrow$ subtuple of $\mathbf{t}$ with entries $i_1, \ldots, i_\delta$

(iii) $\boldsymbol{B} \leftarrow$ the submatrix of $\boldsymbol{B}$ with rows $i_1, \ldots, i_\delta$

(iv) $\hat{\rho}_j \leftarrow \phi_{\prec,\boldsymbol{\beta}}(X_k^{2^e}\phi^{-1}_{\prec,\boldsymbol{\beta}}(\rho_j))$ for $1 \leqslant j \leqslant \delta$

(v) $\pi \leftarrow$ the permutation matrix in $\{0,1\}^{2\delta\times 2\delta}$ such that the tuple $\mathbf{t} = \pi[\rho_1, \ldots, \rho_\delta, \hat{\rho}_1, \ldots, \hat{\rho}_\delta]^\mathsf{T}$ is increasing

(vi) $\boldsymbol{B} \leftarrow \pi \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{BP} \end{bmatrix}$

(vii) $\hat{\delta}, (i_1, \ldots, i_{\hat{\delta}}) \leftarrow$ RowRankProfile($\boldsymbol{B}$)

(viii) $\boldsymbol{P} \leftarrow \boldsymbol{P}^2$; $e \leftarrow e+1$

Until $\hat{\delta} = \delta$ and $(\rho_1, \ldots, \rho_\delta) =$ subtuple of $\mathbf{t}$ with entries $i_1, \ldots, i_\delta$

7. Return $\mathcal{E} = (\phi^{-1}_{\prec,\boldsymbol{\beta}}(\rho_1), \ldots, \phi^{-1}_{\prec,\boldsymbol{\beta}}(\rho_\delta))$ and the submatrix of $\boldsymbol{B}$ with rows $i_1, \ldots, i_\delta$

Proposition 3.7. *Algorithm 1 returns the $\prec$-monomial basis $\mathcal{E} = (\varepsilon_1, \ldots, \varepsilon_\Delta)$ of $\mathbb{K}[\boldsymbol{X}]^m / \operatorname{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$ and the matrix $\boldsymbol{B} \in \mathbb{K}^{\Delta\times D}$ whose rows are $\varepsilon_1 \cdot \boldsymbol{F}, \ldots, \varepsilon_\Delta \cdot \boldsymbol{F}$. It uses*

$$O\left(mD^{\omega-1} + D^\omega(r + \log(d_1 \cdots d_r))\right) \subset O\left(mD^{\omega-1} + rD^\omega \log(D)\right)$$

operations in $\mathbb{K}$, where $d_k \in \{1, \ldots, D\}$ is the degree of the minimal polynomial of $\boldsymbol{M}_k$, for $1 \leqslant k \leqslant r$.

Proof. Let $\beta = 2^{\lceil \log_2(D) \rceil + 1}$ as in the algorithm; for $1 \leqslant k \leqslant r$ and $0 \leqslant e \leqslant \log_2(\beta)$, let us consider the set of monomials

$$\mathcal{S}_{k,e} = \{\boldsymbol{X}^{\mathbf{e}} \boldsymbol{c}_i \mid 1 \leqslant i \leqslant m,\ 0 \leqslant \mathbf{e} < (\beta, \ldots, \beta, 2^e, 1, \ldots, 1)\},$$

where 2^e is the kth entry of the tuple. Then we denote by $\boldsymbol{C}_{k,e} \in \mathbb{K}^{(m\beta^{k-1}2^e) \times D}$ the submatrix of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$ formed by its rows in $\phi_{\prec,\boldsymbol{\beta}}(\mathcal{S}_{k,e})$. For $0 \leqslant e \leqslant e' \leqslant \log_2(\beta)$, $\boldsymbol{C}_{k,e}$ is a submatrix of $\boldsymbol{C}_{k,e'}$, but not necessarily a top submatrix of it. Remark also that for $k < r$, $\boldsymbol{C}_{k,\log_2(\beta)} = \boldsymbol{C}_{k+1,0}$.

The correctness of the algorithm is proved by an induction that involves k and e. For $1 \leqslant k \leqslant r$, denote by $\ell_k \geqslant 0$ the last value of the index e for which we enter the body of the Do-Until loop (Step **6.b**). Then, for $1 \leqslant k \leqslant r$ and $0 \leqslant e \leqslant \ell_k$, define the following assertions, that we consider at the beginning iteration (k, e) of the Do-Until loop:

A_1 : the rows of indices $i_1, \ldots, i_{\hat{\delta}}$ in $\boldsymbol{B}$ are the rows defining the row rank profile of $\boldsymbol{C}_{k,e}$;
A_2 : the entries of indices $i_1, \ldots, i_{\hat{\delta}}$ in $\mathbf{t}$ are the indices of these rows in $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$.

We will prove by induction that these properties hold for all values of k and e considered above. For $k = 1$ and $e = 0$, $\boldsymbol{C}_{1,0}$ is the submatrix of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$ with rows in $\{\phi_{\prec,\boldsymbol{\beta}}(\boldsymbol{c}_1), \ldots, \phi_{\prec,\boldsymbol{\beta}}(\boldsymbol{c}_m)\}$; therefore, by choice of the permutation π at Step **3**, we have $\boldsymbol{C}_{1,0} = \pi\boldsymbol{F} = \boldsymbol{B}$ at Step **4**. Thus, upon entering the For loop for the first time, with $k = 1$ and $e = 0$, we see that A_1 and A_2 hold.

Then let k be in $\{1, \ldots, r\}$ and e in $\{0, \ldots, \ell_k\}$; we assume that A_1 and A_2 hold at indices k and e. Let us denote $\boldsymbol{\rho} = \{\rho_1, \ldots, \rho_\delta\}$ as defined in Step **6.b.**(ii), and let $\hat{\boldsymbol{\rho}} = \{\hat{\rho}_1, \ldots, \hat{\rho}_\delta\}$, where $\hat{\rho}_j = \phi_{\prec,\boldsymbol{\beta}}(X_k^{2^e} \phi^{-1}_{\prec,\boldsymbol{\beta}}(\rho_j))$ for $1 \leqslant j \leqslant \delta$ are the indices computed at Step **6.b.**(iv). Let also $(\gamma_1, \ldots, \gamma_\nu)$ be the indices of the rows of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$ corresponding to the row rank profile of its submatrix $\boldsymbol{C}_{k,e+1}$. To complete this proof, we will use three intermediate lemmas; first, we claim that the following holds:

Lemma 3.8. $(\gamma_1, \ldots, \gamma_\nu)$ *is a subsequence of the tuple* $\mathbf{t}$.

Proof of Lemma 3.8. Let j be in $\{1, \ldots, \nu\}$ and let us prove that γ_j is in $\boldsymbol{\rho} \cup \hat{\boldsymbol{\rho}}$. By assumption, the row of index γ_j in $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$ is not a linear combination of the rows of smaller indices in the submatrix $\boldsymbol{C}_{k,e+1}$ of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$.

Suppose first that $\phi^{-1}_{\prec,\boldsymbol{\beta}}(\gamma_j)$ is in $\mathcal{S}_{k,e}$, so that γ_j actually corresponds to a row in $\boldsymbol{C}_{k,e}$. Since $\boldsymbol{C}_{k,e}$ is a submatrix of $\boldsymbol{C}_{k,e+1}$, the remark above implies that the row of index γ_j in $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$ is not a linear combination of the rows of smaller indices in $\boldsymbol{C}_{k,e}$. This means that this row belongs to the row rank profile of $\boldsymbol{C}_{k,e}$, and so (by A_2) γ_j is in $\boldsymbol{\rho}$.

Now, we assume that $\phi^{-1}_{\prec,\boldsymbol{\beta}}(\gamma_j) \in \mathcal{S}_{k,e+1} - \mathcal{S}_{k,e}$, and we prove that $\gamma_j \in \hat{\boldsymbol{\rho}}$, or in other words, that $\phi^{-1}_{\prec,\boldsymbol{\beta}}(\gamma_j) \in \{X_k^{2^e} \phi^{-1}_{\prec,\boldsymbol{\beta}}(\rho_j) \mid 1 \leqslant j \leqslant \delta\}$. Since $\phi^{-1}_{\prec,\boldsymbol{\beta}}(\gamma_j) \in \mathcal{S}_{k,e+1} - \mathcal{S}_{k,e}$, we can write $\phi^{-1}_{\prec,\boldsymbol{\beta}}(\gamma_j) = X_k^{2^e} \boldsymbol{X}^{\boldsymbol{f}} \boldsymbol{c}_i$, with $\boldsymbol{X}^{\boldsymbol{f}} \boldsymbol{c}_i$ in $\mathcal{S}_{k,e}$. Suppose that $\boldsymbol{X}^{\boldsymbol{f}} \boldsymbol{c}_i$ is not in $\phi^{-1}_{\prec,\boldsymbol{\beta}}(\boldsymbol{\rho})$, so that by A_2, the row of index $\phi_{\prec,\boldsymbol{\beta}}(\boldsymbol{X}^{\boldsymbol{f}} \boldsymbol{c}_i)$ in $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$ is a linear combination of the previous rows in $\boldsymbol{C}_{k,e}$. Right-multiply all these rows by $\boldsymbol{M}_k^{2^e}$; this shows that the row indexed by γ_j in $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$ is a linear combination of rows of smaller indices in its submatrix $\boldsymbol{C}_{k,e+1}$, a contradiction. Our claim is proved. □

Now, by A_1 and A_2, after the update at Steps **6.b.**(vi), the rows of $\boldsymbol{B}$ are precisely the rows of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$ of indices in $\boldsymbol{\rho} \cup \hat{\boldsymbol{\rho}}$ sorted in increasing order. Thus, after the row rank profile computation at Step **6.b.**(vii), the rows in $\boldsymbol{B}$ of indices $i_1, \ldots, i_{\hat{\delta}}$ are the rows of $\boldsymbol{C}_{k,e+1}$ corresponding to its row rank profile, and the subtuple of $\mathbf{t}$ with entries $i_1, \ldots, i_{\hat{\delta}}$ is precisely $(\gamma_1, \ldots, \gamma_\nu)$.

If $e < \ell_k$, this implies that A_1 and A_2 still hold at step $(k, e+1)$. Suppose next that instead, $e = \ell_k$. We claim the following.

Lemma 3.9. *We have $\ell_k \leqslant \lceil \log_2(d_k) \rceil$; equivalently, if we exit the* Do-Until *loop at the end of iteration e, then $e \leqslant \lceil \log_2(d_k) \rceil$.*

Proof of Lemma 3.9. First, we observe that the indices of the rows in $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M},\boldsymbol{F})$ corresponding to the row rank profile of $\boldsymbol{C}_{k,e-1}$, and of those corresponding to the row rank profile of $\boldsymbol{C}_{k,e}$, are different. Indeed, A_1 and A_2 show that the former correspond to indices $(\rho_1,\dots,\rho_\delta)$ obtained at Step **6.b.**(ii) at iteration $e-1$, the latter to the same indices at iteration e, and the fact that we did not exit the loop at step $e-1$ implies that they are different.

Suppose then, by means of contradiction, that $e \geqslant \lceil \log_2(d_k) \rceil + 1$, so that $e \geqslant \log_2(d_k) + 1$. Consider a row $\boldsymbol{r}$ in $\boldsymbol{C}_{k,e}$ that is not in $\boldsymbol{C}_{k,e-1}$; then, $\boldsymbol{r} = \boldsymbol{s}\boldsymbol{M}_k^{2^{e-1}}$ for some row $\boldsymbol{s}$ in $\boldsymbol{C}_{k,e-1}$. By assumption, 2^{e-1} is at least equal to the degree d_k of the minimal polynomial of $\boldsymbol{M}_k$. In particular, $\boldsymbol{M}_k^{2^{e-1}}$ is a linear combination of powers of $\boldsymbol{M}_k$ of exponent less than 2^{e-1}. Now, all rows $\boldsymbol{s}\boldsymbol{M}_k^i$, for $i < 2^{e-1}$, are in $\boldsymbol{C}_{k,e}$, and have lower indices than $\boldsymbol{r}$. This implies that $\boldsymbol{r}$ is not in the row rank profile of $\boldsymbol{C}_{k,e}$, and thus $\boldsymbol{C}_{k,e-1}$ and $\boldsymbol{C}_{k,e}$ have the same row rank profile. This contradicts the property in the previous paragraph, hence $e \leqslant \lceil \log_2(d_k) \rceil$. □

Using this property, we prove that A_1 and A_2 now hold for indices $k+1$ and $e = 0$ (this will be enough to conclude our induction proof).

Lemma 3.10. *If we exit the* Do-Until *loop after step e (equivalently, if $e = \ell_k$), then the rows in $\boldsymbol{B}$ of indices $i_1,\dots,i_{\hat{\delta}}$ are the rows of $\boldsymbol{C}_{k,\log_2(\beta)}$ corresponding to its row rank profile, and the subtuple of $\mathbf{t}$ with entries $i_1,\dots,i_{\hat{\delta}}$ is the indices of these rows in $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M},\boldsymbol{F})$.*

Proof of Lemma 3.10. Our assumption means that $(\gamma_1,\dots,\gamma_\nu) = (\rho_1,\dots,\rho_\delta)$; this is equivalent to saying that the indices in $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M},\boldsymbol{F})$ of the row rank profiles of its submatrices $\boldsymbol{C}_{k,e}$ and $\boldsymbol{C}_{k,e+1}$ are the same. In particular, any row in $\boldsymbol{C}_{k,e+1}$ is a linear combination of rows of lower indices in $\boldsymbol{C}_{k,e}$.

We will prove the following below: *any row in $\boldsymbol{C}_{k,\log_2(\beta)}$ is a linear combination of rows of lower indices in $\boldsymbol{C}_{k,e}$*. In particular, this implies that the indices in $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M},\boldsymbol{F})$ of the row rank profiles of its submatrices $\boldsymbol{C}_{k,e+1}$ and $\boldsymbol{C}_{k,\log_2(\beta)}$ are all the same. Since we saw that after Step **6.b.**(vii), the rows in $\boldsymbol{B}$ of indices $i_1,\dots,i_{\hat{\delta}}$ are the rows of $\boldsymbol{C}_{k,e+1}$ corresponding to its row rank profile, and the subtuple of $\mathbf{t}$ with entries $i_1,\dots,i_{\hat{\delta}}$ are the indices of these rows in $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M},\boldsymbol{F})$, this is enough to prove the lemma.

We prove our claim by induction on the rows of $\boldsymbol{C}_{k,\log_2(\beta)}$. Let thus $\boldsymbol{r}$ be a row in $\boldsymbol{C}_{k,\log_2(\beta)}$, and assume the claim holds for all previous rows.

If $\boldsymbol{r}$ is from its submatrix $\boldsymbol{C}_{k,e}$, we are done. Else, since $e+1 \leqslant \log_2(\beta)$ holds (from the previous lemma), $\boldsymbol{r}$ can be written as $\boldsymbol{r} = \boldsymbol{s}\boldsymbol{M}_k^c$, for some $c \geqslant 0$, where $\boldsymbol{s}$ is a row in $\boldsymbol{C}_{k,e+1}$. We know that $\boldsymbol{s}$ is a linear combination of rows $\boldsymbol{s}_1,\dots,\boldsymbol{s}_\iota$ of lower indices in $\boldsymbol{C}_{k,e}$, so that $\boldsymbol{r}$ is a linear combination of $\boldsymbol{s}_1\boldsymbol{M}_k^c,\dots,\boldsymbol{s}_\iota\boldsymbol{M}_k^c$. All these rows are in $\boldsymbol{C}_{k,\log_2(\beta)}$ and have lower indices than $\boldsymbol{r}$. By our induction assumption, they are linear combinations of rows of lower indices in $\boldsymbol{C}_{k,e}$, and thus so is $\boldsymbol{r}$. □

(Continuing proof of Proposition 3.7.) If $k < r$ then we can turn to the next variable X_{k+1}, since the former lemma, together with the equality $\boldsymbol{C}_{k,\log_2(\beta)} = \boldsymbol{C}_{k+1,0}$, shows that A_1 and A_2 hold for indices $(k+1,0)$. For $k = r$, since $\boldsymbol{C}_{r,\log_2(\beta)} = \mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M},\boldsymbol{F})$, the lemma shows that the output of the algorithm is indeed the row rank profile of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M},\boldsymbol{F})$ and the submatrix of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M},\boldsymbol{F})$ formed by the corresponding rows. According to Theorem 3.6, the $\prec$-monomial basis can directly be deduced from the rank profile of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M},\boldsymbol{F})$. This concludes the proof of correctness.

Concerning the cost bound, according to [46, Thm. 2.10], the row rank profile computation at Step **5** can be computed in $O(\rho^{\omega-2}mD)$ operations, where $\rho \leqslant D$ is the rank of $\boldsymbol{F}$. In particular, this is $O(mD^{\omega-1})$.

Let us now focus on the iteration (k, e) and we show that it uses $O(D^\omega)$ operations. First, note that $\delta \leqslant D$ holds throughout (since δ is the rank of a matrix with D columns). Since upon entering Step **6.b.**(vi), $\boldsymbol{B}$ has $\delta \leqslant D$ rows and D columns, its update and the row rank profile of the latter can be computed in $O(D^\omega)$ operations. Finally, squaring the $D \times D$ matrix $\boldsymbol{P}$ at Step **6.b.**(viii) is also done in $O(D^\omega)$ operations.

To conclude the proof of the cost bound, we recall from Lemma 3.9 that in iteration k of the For loop, we pass $\ell_k + 1 \leqslant \lceil \log_2(d_k) \rceil + 1$ times through the body of the Do-Until loop. □

Remark 3.11. We may slightly refine the analysis for some particular monomial orders. Indeed, the order in which the For and Do-Until loops introduce the new monomials to be processed corresponds to the $\prec_{\text{lex}}$-term over position order $\prec_{\text{lex}}^{\text{top}}$ over $\mathbb{K}[\boldsymbol{X}]^m$, with $X_1 \prec_{\text{lex}} \cdots \prec_{\text{lex}} X_r$. As a result, the behaviour and the cost bound of the algorithm can be described with more precision if the input monomial order is $\prec = \prec_{\text{lex}}^{\text{top}}$.

In this case, we are processing the rows of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$ in the order they are in the matrix. In particular, the permutation π at Steps **3** and **6.b.**(v) is always the identity matrix, and the tuple $(\rho_1, \ldots, \rho_\delta)$ inside the loops consists of the first δ entries of the actual row rank profile of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$.

Furthermore, the fact that we are processing the rows in their actual order has a small impact on the cost bound, as follows. Let us denote by $\mathcal{G}$ the reduced $\prec$-Gröbner basis of $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$, and let $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \ldots, \hat{\beta}_r)$ be the tuple of maximum degrees in $\mathcal{G}$, that is, $\hat{\beta}_k = \max_{\boldsymbol{p} \in \mathcal{G}} \deg_{X_k}(\boldsymbol{p})$ for $1 \leqslant k \leqslant r$.

Then, at iteration k of the For loop, the Do-Until loop does at most $\lceil \log_2(\hat{\beta}_k + 1) \rceil + 1$ iterations; indeed, when reaching the iteration that introduces powers of the variable X_k all greater than $\hat{\beta}_k$, the partial row rank profile $(\rho_1, \ldots, \rho_\delta)$ is not modified anymore, and the Do-Until loop exits. Now, considering the total number of iterations, we claim that

$$\sum_{1 \leqslant k \leqslant r} \log(\hat{\beta}_k + 1) \;\leqslant\; r \log\left(\frac{\hat{\beta}_1 + \cdots + \hat{\beta}_r}{r} + 1\right) \;\leqslant\; r \log\left(\frac{D}{r} + 2\right). \tag{1}$$

As a result, when the input order is $\prec = \prec_{\text{lex}}^{\text{top}}$, Algorithm 1 uses

$$O\left(mD^{\omega-1} + rD^\omega \log\left(\frac{D}{r} + 2\right)\right)$$

operations in $\mathbb{K}$.

We now prove our claim. The first inequality in Eq. (1) is a direct application of the arithmetic mean-geometric mean inequality. The second inequality follows from the bound $\hat{\beta}_1 + \cdots + \hat{\beta}_r \leqslant D + r - 1$, which holds since the $\prec$-monomial basis, whose cardinality Δ is at most D, contains the $1 + \hat{\beta}_1 + \cdots + \hat{\beta}_r - r$ distinct monomials specified hereafter. By definition of $\hat{\boldsymbol{\beta}}$, for each $k \in \{1, \ldots, r\}$ such that $\hat{\beta}_k > 0$, there is a monomial appearing in some element of $\mathcal{G}$ which is a multiple of $X_k^{\hat{\beta}_k} \boldsymbol{c}_{i_k}$ for some $1 \leqslant i_k \leqslant m$. Thus $X_k^{\hat{\beta}_k} \boldsymbol{c}_{i_k}$ is either in the $\prec$-monomial basis or in its border, which implies that the $\prec$-monomial basis contains $\{\boldsymbol{c}_{i_k}, X_k \boldsymbol{c}_{i_k}, \ldots, X_k^{\hat{\beta}_k - 1} \boldsymbol{c}_{i_k}\}$. Considering the union of all such sets for $1 \leqslant k \leqslant r$ (with the empty set if $\hat{\beta}_k = 0$) yields at least $1 + (\hat{\beta}_1 - 1) + \cdots + (\hat{\beta}_r - 1)$ distinct monomials, since each intersection of a pair of such sets has at most one element (the coordinate vector). □

3.3. *Fast computation of the basis of syzygies*

Next, we present our algorithm to compute the reduced Gröbner basis of $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$. By definition, it can be described by the minimal generators of the leading module of $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$ along with the associated normal forms. We first show how to use the knowledge of the monomial basis to compute such normal forms efficiently. In all this section, we let $\boldsymbol{M} = (\boldsymbol{M}_1, \dots, \boldsymbol{M}_r)$ be pairwise commuting matrices in $\mathbb{K}^{D\times D}$, $\boldsymbol{F}$ be in $\mathbb{K}^{m\times D}$, $\prec$ be a monomial order on $\mathbb{K}[\boldsymbol{X}]^m$, and $\mathcal{E}$ be the $\prec$-monomial basis of $\mathbb{K}[\boldsymbol{X}]^m / \mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$.

3.3.1. *Simultaneous computation of normal forms of monomials*

First, for some arbitrary monomials $\{\boldsymbol{X}^{\mathbf{e}_1}\boldsymbol{c}_{i_1}, \dots, \boldsymbol{X}^{\mathbf{e}_s}\boldsymbol{c}_{i_s}\}$, we give an algorithm that computes their $\prec$-normal forms with respect to $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$. Each of these $\prec$-normal forms is a uniquely defined $\mathbb{K}$-linear combination of the monomials in the $\prec$-monomial basis $\mathcal{E}$; the main task of our algorithm is to find the coefficients of these s combinations, which we gather below in an $s \times \Delta$ matrix $\boldsymbol{N}$.

In the linearized viewpoint, we associate the monomials $\{\boldsymbol{X}^{\mathbf{e}_1}\boldsymbol{c}_{i_1}, \dots, \boldsymbol{X}^{\mathbf{e}_s}\boldsymbol{c}_{i_s}\}$ with a matrix $\boldsymbol{T} \in \mathbb{K}^{s\times D}$, whose jth row is $\boldsymbol{X}^{\mathbf{e}_j}\boldsymbol{c}_{i_j} \cdot \boldsymbol{F} = \boldsymbol{f}_{i_j}\boldsymbol{M}^{\mathbf{e}_j}$, where $\boldsymbol{f}_{i_j}$ is the i_jth row of $\boldsymbol{F}$ (we are using the notation of Definition 1.1). Similarly, we associate the monomial basis $\mathcal{E}$ with a matrix $\boldsymbol{B} \in \mathbb{K}^{\Delta\times D}$, where $\Delta = \dim_{\mathbb{K}}(\mathbb{K}[\boldsymbol{X}]^m / \mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F}))$ as above; if we write $\mathcal{E} = (\varepsilon_1 \prec \cdots \prec \varepsilon_\Delta)$, then the jth row of $\boldsymbol{B}$ is $\varepsilon_j \cdot \boldsymbol{F}$.

Then the rows of $\boldsymbol{T}$ are $\mathbb{K}$-linear combinations of the rows of $\boldsymbol{B}$: there is a matrix $\boldsymbol{N} \in \mathbb{K}^{s\times\Delta}$ such that $\boldsymbol{T} = \boldsymbol{N}\boldsymbol{B}$. (The notation $\boldsymbol{T}$ stands for *terms*, while $\boldsymbol{B}$ stands for *basis*, and $\boldsymbol{N}$ for *normal forms*.) To compute this matrix $\boldsymbol{N}$, one can directly use $\boldsymbol{N} = \boldsymbol{T}\boldsymbol{B}^{-1}$ if $\boldsymbol{B}$ is square, and more generally one can use a similar identity $\boldsymbol{N} = \bar{\boldsymbol{T}}\bar{\boldsymbol{B}}^{-1}$ where $\bar{\boldsymbol{B}}$ is a $\Delta \times \Delta$ invertible submatrix of $\boldsymbol{B}$, as detailed in Step **1** of Algorithm 2.

Proposition 3.12. *Algorithm 2 is correct and uses $O(\Delta^{\omega-1}(D + s))$ operations in $\mathbb{K}$.*

Proof. For the correctness, we focus on the case $s = 1$; to prove the general case $s \geqslant 1$ it suffices to apply the following arguments for each $j \in \{1, \dots, s\}$, to the jth row of $\boldsymbol{T}$ and the corresponding output element $\boldsymbol{\nu}_j$. Thus, we consider $\boldsymbol{T} = \boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i \cdot \boldsymbol{F} = \boldsymbol{f}_i\boldsymbol{M}^{\mathbf{e}}$ for some $\mathbf{e} \in \mathbb{N}^r$ and $1 \leqslant i \leqslant m$, and our goal is to prove that $\mathrm{nf}_\prec(\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i) = \nu_1\varepsilon_1 + \cdots + \nu_\Delta\varepsilon_\Delta$ where $\boldsymbol{N} = [\nu_1 \quad \cdots \quad \nu_\Delta]$ is the unique vector in $\mathbb{K}^{1\times\Delta}$ such that $\boldsymbol{T} = \boldsymbol{N}\boldsymbol{B}$. Choosing large enough exponent bounds $\boldsymbol{\beta} \in \mathbb{N}^r_{>0}$, such as $\boldsymbol{\beta} = (\max(\mathbf{e}, D)+1, \dots, \max(\mathbf{e}, D)+1)$, we recall from Definition 3.3 that $\boldsymbol{T}$ is a row of the multi-Krylov matrix $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$, and from Theorem 3.6 that $\boldsymbol{B}$ is the submatrix of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$ formed by the rows corresponding to its row rank profile. This proves the existence and uniqueness of $\boldsymbol{N}$ such that $\boldsymbol{T} = \boldsymbol{N}\boldsymbol{B}$.

We now explain the computation of $\boldsymbol{N}$ in Step **1**. We use the column rank profile of $\boldsymbol{B}$ as a specific set of column indices $\bar{\rho}_1 < \cdots < \bar{\rho}_\Delta$ such that the corresponding $\Delta \times \Delta$ submatrix $\bar{\boldsymbol{B}}$ of $\boldsymbol{B}$ is invertible. Then, writing $\bar{\boldsymbol{T}} \in \mathbb{K}^{1\times\Delta}$ for the subvector of $\boldsymbol{T}$ formed by its entries $\{\bar{\rho}_1, \dots, \bar{\rho}_\Delta\}$, the identity $\boldsymbol{T} = \boldsymbol{N}\boldsymbol{B}$ yields $\bar{\boldsymbol{T}} = \boldsymbol{N}\bar{\boldsymbol{B}}$, hence $\boldsymbol{N} = \bar{\boldsymbol{T}}\bar{\boldsymbol{B}}^{-1}$.

Since the jth row of $\boldsymbol{B}$ is $\varepsilon_j \cdot \boldsymbol{F}$, we have $\mathbf{0} = \boldsymbol{T} - \boldsymbol{N}\boldsymbol{B} = (\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i - \nu_1\varepsilon_1 + \cdots + \nu_\Delta\varepsilon_\Delta) \cdot \boldsymbol{F}$, hence $\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i - \nu_1\varepsilon_1 + \cdots + \nu_\Delta\varepsilon_\Delta$ is in $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$. Thus

$$\mathrm{nf}_\prec(\boldsymbol{X}^{\mathbf{e}}\boldsymbol{c}_i) = \mathrm{nf}_\prec(\nu_1\varepsilon_1 + \cdots + \nu_\Delta\varepsilon_\Delta) = \nu_1\varepsilon_1 + \cdots + \nu_\Delta\varepsilon_\Delta;$$

indeed $\nu_1\varepsilon_1 + \cdots + \nu_\Delta\varepsilon_\Delta$ is already in $\prec$-normal form as it is a combination of the $\prec$-monomial basis $\mathcal{E}$ of $\mathbb{K}[\boldsymbol{X}]^m / \mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$. This concludes the proof of correctness.

Algorithm 2 – NormalForm

Input:

- matrix $\boldsymbol{T} \in \mathbb{K}^{s\times D}$,
- list of monomials $\mathcal{E} = (\varepsilon_1, \dots, \varepsilon_\Delta)$ in $\mathbb{K}[\boldsymbol{X}]^m$,
- matrix $\boldsymbol{B} \in \mathbb{K}^{\Delta\times D}$ with full row rank.

Output: list $(\boldsymbol{v}_1, \dots, \boldsymbol{v}_s)$ of elements of $\mathbb{K}[\boldsymbol{X}]^m$

Ensures: assuming the following holds:

- $\boldsymbol{M} = (\boldsymbol{M}_1, \dots, \boldsymbol{M}_r)$ are pairwise commuting matrices in $\mathbb{K}^{D\times D}$,
- $\boldsymbol{F}$ is a matrix in $\mathbb{K}^{m\times D}$ with rows $\boldsymbol{f}_1, \dots, \boldsymbol{f}_m$,
- the jth row of $\boldsymbol{T}$ is $\boldsymbol{X}^{\mathbf{e}_j}\boldsymbol{c}_{i_j} \cdot \boldsymbol{F} = \boldsymbol{f}_{i_j}\boldsymbol{M}^{\mathbf{e}_j}$ for some monomial $\boldsymbol{X}^{\mathbf{e}_j}\boldsymbol{c}_{i_j}$,
- $\mathcal{E}$ is the $\prec$-monomial basis of $\mathbb{K}[\boldsymbol{X}]^m / \operatorname{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$ for some monomial order $\prec$ on $\mathbb{K}[\boldsymbol{X}]^m$,
- the jth row of $\boldsymbol{B}$ is $\varepsilon_j \cdot \boldsymbol{F}$,

then $\boldsymbol{v}_j = \operatorname{nf}_\prec(\boldsymbol{X}^{\mathbf{e}_j}\boldsymbol{c}_{i_j})$ for $1 \leqslant j \leqslant s$.

1. `/* Compute the matrix` $\boldsymbol{N} \in \mathbb{K}^{s\times\Delta}$ `such that` $\boldsymbol{T} = \boldsymbol{N}\boldsymbol{B}$ `*/`
   $(\bar\rho_1, \dots, \bar\rho_\Delta) \leftarrow$ the column rank profile of $\boldsymbol{B}$
   $\bar{\boldsymbol{B}}$ and $\bar{\boldsymbol{T}} \leftarrow$ submatrices of $\boldsymbol{B}$ and $\boldsymbol{T}$ formed by the columns $\{\bar\rho_1, \dots, \bar\rho_\Delta\}$
   $\boldsymbol{N} = [v_{i,j}]_{1\leqslant i\leqslant s, 1\leqslant j\leqslant\Delta} \leftarrow \bar{\boldsymbol{T}}\bar{\boldsymbol{B}}^{-1}$
2. `/* Deduce normal forms */`
   For i from 1 to s: $\boldsymbol{v}_i \leftarrow v_{i,1}\varepsilon_1 + \cdots + v_{i,\Delta}\varepsilon_\Delta \in \mathbb{K}[\boldsymbol{X}]^m$
3. Return $(\boldsymbol{v}_1, \dots, \boldsymbol{v}_s)$

Concerning the cost bound, Steps **2** and **3** do not require operations in $\mathbb{K}$. In Step **1**, the column rank profile is obtained in $O(\Delta^{\omega-1}D)$ operations according to [46, Thm. 2.10]; the inversion of $\bar{\boldsymbol{B}}$ costs $O(\Delta^\omega)$; and the multiplication $\bar{\boldsymbol{T}}\bar{\boldsymbol{B}}^{-1}$ uses $O(s\Delta^{\omega-1})$ operations if $s \geqslant \Delta$, and $O(\Delta^\omega)$ otherwise. Since $\Delta \leqslant D$ we obtain the announced bound. □

Remark 3.13. One may observe that Algorithm 2 works in the more general case where $\mathcal{E}$ is a basis of the $\mathbb{K}$-vector space $\mathbb{K}[\boldsymbol{X}]^m / \operatorname{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$, and thus in particular when $\mathcal{E}$ is the monomial basis associated to a border basis of $\operatorname{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$ which is not necessarily related to a monomial order. In that case, each output element $\boldsymbol{v}_j \in \mathbb{K}[\boldsymbol{X}]^m$ is the unique polynomial which is equal to $\boldsymbol{X}^{\mathbf{e}_j}\boldsymbol{c}_{i_j}$ modulo $\operatorname{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$ and whose monomials are in $\mathcal{E}$. Besides, the argument referring to Theorem 3.6 in the proof above can be replaced by the fact that, assuming $\boldsymbol{\beta}$ large enough, $\boldsymbol{B}$ is a submatrix of $\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F})$ formed by $\Delta = \operatorname{rank}(\mathcal{K}_{\prec,\boldsymbol{\beta}}(\boldsymbol{M}, \boldsymbol{F}))$ linearly independent rows. The above cost bound thus also holds in this more general case, yet one should note that using Algorithm 2 requires the knowledge of the input matrix $\boldsymbol{B}$, which might be expensive to compute from $\mathcal{E}$ and $\boldsymbol{F}$ depending on the choice of $\mathcal{E}$. In our specific case, Algorithm 1 outputs both $\mathcal{E}$ and $\boldsymbol{B}$. □

3.3.2. Computing reduced Gröbner bases of syzygies

To compute the reduced $\prec$-Gröbner basis $\mathcal{G}$ of $\operatorname{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$, we start by using Algorithm 1 to find the $\prec$-monomial basis $\mathcal{E} = (\varepsilon_1, \dots, \varepsilon_\Delta)$, together with the matrix $\boldsymbol{B}$ giving all $\varepsilon_k \cdot \boldsymbol{F}$. From $\mathcal{E}$, we deduce the set $\mathcal{L} = \{\operatorname{lt}_\prec(\boldsymbol{p}) \mid \boldsymbol{p} \in \mathcal{G}\}$ formed by the $\prec$-leading terms of the polynomials in $\mathcal{G}$, as explained in the next paragraph. Finally, having $\mathcal{L}$, we compute $\prec$-normal forms modulo $\operatorname{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$ using Algorithm 2 so as to obtain $\mathcal{G} = \{f - \operatorname{nf}_\prec(f) \mid f \in \mathcal{L}\}$. We refer to Section 2 for

more details concerning the latter identity and the sets of monomials $\mathcal{S}$ and $\mathcal{B}$ used in the next paragraph.

To find $\mathcal{L}$, we start from $\mathcal{E} = (\varepsilon_1, \dots, \varepsilon_\Delta)$ and consider the set of multiples

$$\mathcal{S} = \{X_k \varepsilon_j \mid 1 \leqslant k \leqslant r, 1 \leqslant j \leqslant \Delta\} \cup \{\boldsymbol{c}_i \mid 1 \leqslant i \leqslant m \text{ such that } \boldsymbol{c}_i \notin \mathcal{E}\}.$$

It gives us the border, as $\mathcal{B} = \mathcal{S} - \mathcal{E}$. The latter is a set of monomial generators for the monomial submodule $\langle \mathrm{lt}_\prec(\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F}))\rangle$, while $\mathcal{L}$ is the minimal generating set of the same submodule. Thus $\mathcal{L}$ can be found from $\mathcal{B}$ by discarding all monomials in $\mathcal{B}$ which are divisible by another monomial in $\mathcal{B}$. The number of generators $\mathrm{Card}(\mathcal{G})$ is not known in advance; it is at least m, and at most $r\Delta + m$ as explained in Section 2. In particular, the output Gröbner basis is represented using $rD^2 + mD$ field elements, as claimed in the introduction.

The $\prec$-normal forms of the monomials in $\mathcal{L}$ will be computed using Algorithm 2; for this, we need to know $f \cdot \boldsymbol{F}$, for f in $\mathcal{L}$. The matrix $\boldsymbol{B}$ describes all $\varepsilon_j \cdot \boldsymbol{F}$, for $1 \leqslant j \leqslant \Delta$; on the other hand, we know that any f in $\mathcal{L}$ which is not among $\{\boldsymbol{c}_1, \dots, \boldsymbol{c}_m\}$ is a product of the form $f = X_k \varepsilon_j$, for some k in $\{1, \dots, r\}$ and j in $\{1, \dots, \Delta\}$. In such a case, $f \cdot \boldsymbol{F}$ can be computed as $(\varepsilon_j \cdot \boldsymbol{F})\boldsymbol{M}_k$; in the algorithm, we use fast matrix multiplication to compute several $f \cdot \boldsymbol{F}$ at once. Altogether, Algorithm 3 and Proposition 3.14 prove Theorem 1.7.

Proposition 3.14. *Algorithm 3 is correct and uses*

$$O\left(mD^{\omega-1} + D^\omega(r + \log(d_1 \cdots d_r))\right) \subset O\left(mD^{\omega-1} + rD^\omega \log(D)\right)$$

operations in $\mathbb{K}$, where $d_k \in \{1, \dots, D\}$ is the degree of the minimal polynomial of $\boldsymbol{M}_k$, for $1 \leqslant k \leqslant r$.

Proof. Concerning correctness, the construction of $\boldsymbol{B}$ ensures that after Step **2.d**, the rows of $\boldsymbol{T}_k$ are the rows $X_k^{-1}\boldsymbol{X}^{\mathbf{e}_{k,j}}\boldsymbol{c}_{i_{k,j}} \cdot \boldsymbol{F}$. Therefore, after Step **2.e** the rows of $\boldsymbol{T}_k$ are the rows $\boldsymbol{X}^{\mathbf{e}_{k,j}}\boldsymbol{c}_{i_{k,j}} \cdot \boldsymbol{F} = \boldsymbol{f}_{i_{k,j}}\boldsymbol{M}^{\mathbf{e}_{k,j}}$. Then Proposition 3.12 implies that $\boldsymbol{v}_{k,j}$ computed at Step **3** is the normal form $\mathrm{nf}_\prec(\boldsymbol{X}^{\mathbf{e}_{k,j}}\boldsymbol{c}_{i_{k,j}})$, for $1 \leqslant j \leqslant s_k$ and $0 \leqslant k \leqslant r$. This shows the correctness of the algorithm since, as explained above, the reduced $\prec$-Gröbner basis of syzygies is $\{f - \mathrm{nf}_\prec(f) \mid f \in \mathcal{L}\}$.

Concerning the cost bound, the cost of Step **1** is given in Proposition 3.7 and is precisely the cost bound in the present proposition. Then, at the iteration k of the For loop, the multiplication at Step **2.e** involves the $s_k \times D$ matrix $\boldsymbol{T}_k$ and the $D \times D$ matrix $\boldsymbol{M}_k$. For k in $\{1, \dots, r\}$, since we have by definition $s_k = \mathrm{Card}(\mathcal{L}_k) \leqslant \mathrm{Card}(\mathcal{E}) = \Delta \leqslant D$, this multiplication is performed in $O(D^\omega)$ operations; over the r iterations, this leads to a total of $O(rD^\omega)$ operations. For $k = 0$, we have $s_0 = m$, so the cost is $O(mD^{\omega-1} + D^\omega)$. Finally, we have $\mathcal{L}_0 \cup \cdots \cup \mathcal{L}_r = \mathcal{L}$ and therefore $s_0 + \cdots + s_r \leqslant \mathrm{Card}(\mathcal{L}) \leqslant r\Delta + m \leqslant rD + m$; hence the cost for computing normal forms at Step **3** is in $O(\Delta^{\omega-1}(D + s_0 + \cdots + s_r)) \subseteq O(mD^{\omega-1} + rD^\omega)$ according to Proposition 3.12. □

Remark 3.15. By considering $\mathcal{B}$ instead of $\mathcal{L}$ at the second step, one could slightly modify Algorithm 3 so that, instead of the reduced $\prec$-Gröbner basis, it returns the border basis with respect to the $\prec$-monomial basis computed at the first step. One can verify that the computation of that monomial basis remains the most expensive step of the modified algorithm, and thus that the overall cost bound is the same as the one in Proposition 3.14. □

4. Computing multiplication matrices from the Gröbner basis

In this section, we tackle Problem 2 and prove Theorem 1.9. In what follows, $\mathcal{N}$ is a submodule of $\mathbb{K}[\boldsymbol{X}]^n$ such that $\mathbb{K}[\boldsymbol{X}]^n/\mathcal{N}$ has finite dimension D as a $\mathbb{K}$-vector space, $\prec$ is a monomial

Algorithm 3 – SyzygyModuleBasis

Input:

- monomial order $\prec$ on $\mathbb{K}[X_1, \ldots, X_r]^{1\times m}$,
- pairwise commuting matrices $\boldsymbol{M} = (\boldsymbol{M}_1, \ldots, \boldsymbol{M}_r)$ in $\mathbb{K}^{D\times D}$,
- matrix $\boldsymbol{F} \in \mathbb{K}^{m\times D}$.

Output: the reduced $\prec$-Gröbner basis of $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$.

1. `/* Compute monomial basis */`
$\mathcal{E} = (\varepsilon_1, \ldots, \varepsilon_\Delta), \boldsymbol{B} \leftarrow$ MonomialBasis$(\prec, \boldsymbol{M}, \boldsymbol{F})$

2. `/* Compute leading monomials and their linearizations */`
$\mathcal{L} \leftarrow$ minimal generating set of $\langle \mathrm{lt}_\prec(\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F}))\rangle$, deduced from $\mathcal{E}$

$\mathcal{L}_0 \leftarrow \mathcal{L} \cap \{\boldsymbol{c}_i \mid 1 \leqslant i \leqslant m\}$

write $\mathcal{L}_0$ as $\{\boldsymbol{c}_{i_{0,j}} \mid 1 \leqslant j \leqslant s_0\}$ for some indices $i_{0,1}, \ldots, i_{0,s_0}$

$(\mathbf{e}_{0,1}, \ldots, \mathbf{e}_{0,s_0}) \leftarrow (0, \ldots, 0)$

For k from 1 to r

a. $\mathcal{L}_k \leftarrow \{f \in \mathcal{L} - (\mathcal{L}_0 \cup \cdots \cup \mathcal{L}_{k-1}) \mid X_k \text{ divides } f \text{ and } X_k^{-1} f \in \mathcal{E}\}$

b. write $\mathcal{L}_k$ as $\{\boldsymbol{X}^{\mathbf{e}_{k,j}} \boldsymbol{c}_{i_{k,j}} \mid 1 \leqslant j \leqslant s_k\}$ for some exponents and indices $\mathbf{e}_{k,j}, i_{k,j}$

c. For j from 1 to s_k: $\mu_j \leftarrow$ index such that $X_k^{-1} \boldsymbol{X}^{\mathbf{e}_{k,j}} \boldsymbol{c}_{i_{k,j}} = \varepsilon_{\mu_j}$

d. $\boldsymbol{T}_k \leftarrow$ matrix formed by the rows $\mu_1, \ldots, \mu_{s_k}$ of $\boldsymbol{B}$, in this order

e. $\boldsymbol{T}_k \leftarrow \boldsymbol{T}_k \boldsymbol{M}_k$

$\boldsymbol{T} \leftarrow \begin{bmatrix} \boldsymbol{T}_0 \\ \boldsymbol{T}_1 \\ \vdots \\ \boldsymbol{T}_r \end{bmatrix}$

3. `/* Compute normal forms` $\boldsymbol{\nu}_{k,j} = \mathrm{nf}_\prec(\boldsymbol{X}^{\mathbf{e}_{k,j}} \boldsymbol{c}_{i_{k,j}})$ `and return */`
$(\boldsymbol{\nu}_{0,1}, \ldots, \boldsymbol{\nu}_{0,s_0}, \ldots, \boldsymbol{\nu}_{r,1}, \ldots, \boldsymbol{\nu}_{r,s_r}) \leftarrow$ NormalForm$(\boldsymbol{T}, \mathcal{E}, \boldsymbol{B})$

Return $\{\boldsymbol{X}^{\mathbf{e}_{k,j}} \boldsymbol{c}_{i_{k,j}} - \boldsymbol{\nu}_{k,j} \mid 1 \leqslant j \leqslant s_k, 0 \leqslant k \leqslant r\}$

order on $\mathbb{K}[\boldsymbol{X}]^n$, and $\mathcal{G}$ is the reduced $\prec$-Gröbner basis of $\mathcal{N}$. Having as input $\mathcal{G}$, we give an algorithm to compute the multiplication matrices for this quotient with respect to the $\prec$-monomial basis, under the assumption on the leading module of $\mathcal{N}$ described in Definition 1.8. For conciseness, hereafter this assumption is called the *structural assumption*.

4.1. Overview of the algorithm

We first discuss the shape of the $\prec$-monomial basis of $\mathbb{K}[\boldsymbol{X}]^n/\mathcal{N}$ (see [16] and [35, Sec. 3] for similar observations in the case of ideals), and then we present an overview of our approach for computing the multiplication matrices.

Let $\mathcal{E} = (\varepsilon_1, \ldots, \varepsilon_D)$ be the $\prec$-monomial basis of $\mathbb{K}[\boldsymbol{X}]^n/\mathcal{N}$. Below, when discussing multiplication matrices (with respect to $\mathcal{E}$) and elements of $\mathbb{K}[\boldsymbol{X}]^n/\mathcal{N}$ represented on the basis $\mathcal{E}$, we assume that one has fixed an order on the elements of this basis, for example $\varepsilon_1 \prec \cdots \prec \varepsilon_D$. Then the sought multiplication matrices $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_r \in \mathbb{K}^{D\times D}$ are such that the row j of $\boldsymbol{M}_k$ is the coefficient vector of the normal form $\mathrm{nf}_\prec(X_k\varepsilon_j)$ represented in the basis $\mathcal{E}$, for $1 \leqslant k \leqslant r$ and $1 \leqslant j \leqslant D$. Thus, we consider the set of these monomials obtained by multiplying those of $\mathcal{E}$ by a variable:

$$\mathcal{S} = \{X_k\varepsilon_j \mid 1 \leqslant k \leqslant r, 1 \leqslant j \leqslant D\} \cup \{\boldsymbol{c}_i \mid 1 \leqslant i \leqslant n \text{ such that } \boldsymbol{c}_i \in \mathrm{lt}_\prec(\mathcal{N})\}.$$

Note that we have added the coordinate vectors $\boldsymbol{c}_i$ that are in the leading module $\mathrm{lt}_\prec(\mathcal{N})$, or equivalently that are not in $\mathcal{E}$. This is because the normal forms of these coordinate vectors will also be computed by our algorithm, for a negligible cost since they will be directly obtained from the $\prec$-Gröbner basis.

Regarding the computation of the normal forms of the monomials in $\mathcal{S}$, we can divide them into three disjoint categories:

$$\mathcal{S} = (\mathcal{S} - \mathcal{B}) \cup \mathcal{L} \cup (\mathcal{B} - \mathcal{L}),$$

where $\mathcal{B} = \mathcal{S} - \mathcal{E}$ is the border and $\mathcal{L} = \mathrm{lt}_\prec(\mathcal{G}) \subseteq \mathcal{B}$ is the minimal generating set of $\langle \mathrm{lt}_\prec(\mathcal{N})\rangle$ (see Section 2 for more details, and Fig. 1 for an example).

The first set $\mathcal{S} - \mathcal{B}$ is contained in $\mathcal{E}$; precisely,

$$\mathcal{E} \;=\; (\mathcal{S} - \mathcal{B}) \;\cup\; \{\boldsymbol{c}_i \mid 1 \leqslant i \leqslant n \text{ such that } \boldsymbol{c}_i \notin \mathrm{lt}_\prec(\mathcal{N})\}.$$

As a result, each monomial in $\mathcal{S} - \mathcal{B}$ is its own $\prec$-normal form, and the corresponding rows of the multiplication matrices are coordinate vectors of length D which are obtained for free.

The monomials in the second set $\mathcal{L}$ are the $\prec$-leading terms of the elements of $\mathcal{G}$, so that $\mathcal{G} = \{f - \mathrm{nf}_\prec(f) \mid f \in \mathcal{L}\}$. Thus, from the knowledge of $\mathcal{G}$, one can obtain $\mathrm{nf}_\prec(\mathcal{L})$ using at most sD computations of opposites in $\mathbb{K}$, where $s = \mathrm{Card}(\mathcal{G})$; by opposite, we mean having on input $\alpha \in \mathbb{K}$ and computing $-\alpha$. We recall from Section 2 that $s = \mathrm{Card}(\mathcal{L}) \leqslant \mathrm{Card}(\mathcal{S}) < rD + n$ (the bound is strict here since $D > 0$).

Thus, to obtain the multiplication matrices, the main task is to compute the normal forms of the third set $\mathcal{B} - \mathcal{L}$. As discussed above, our algorithm works under the structural assumption $\mathcal{H}(\langle \mathrm{lt}_\prec(\mathcal{N})\rangle)$ from Definition 1.8. The next lemma summarizes the above discussion about the computation of $\mathrm{nf}_\prec(\mathcal{E}\cup\mathcal{L})$, and also highlights one example of how one can exploit the structural assumption; note that this result appears in [14, Sec. 7] in the case of ideals, under slightly different assumptions.

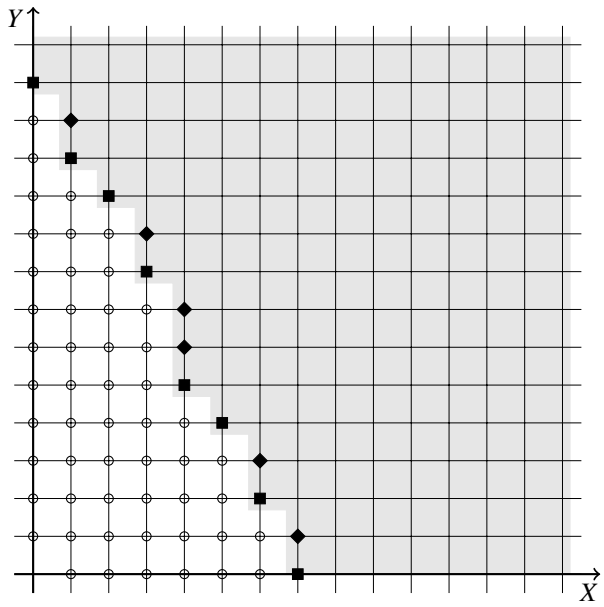

Figure 1: Illustration of the sets of exponents in the case of the bivariate monomial ideal generated by $\mathcal{L} = \{Y^{13}, Y^{11}X, Y^{10}X^2, Y^8X^3, Y^5X^4, Y^4X^5, Y^2X^6, X^7\}$. The elements of $\mathcal{L}$ are represented by squares, those of $\mathcal{B} - \mathcal{L}$ by diamonds, and those of $\mathcal{S} - \mathcal{B}$ by circles. Here, the monomial basis is $\mathcal{E} = \{1\} \cup (\mathcal{S} - \mathcal{B})$. Monomials in the greyed area are those in $\langle \mathcal{L} \rangle$, or in other words, those not in $\mathcal{E}$.

Lemma 4.1. *Given the reduced $\prec$-Gröbner basis $\mathcal{G}$ of $\mathcal{N}$, one can compute* $\mathrm{nf}_\prec(\mathcal{E} \cup \mathcal{L})$ *using at most sD operations in $\mathbb{K}$, where s is the cardinality of $\mathcal{G}$. Assuming $\mathcal{H}(\langle \mathrm{lt}_\prec(\mathcal{N}) \rangle)$, we have $\{X_r\varepsilon_j \mid 1 \leqslant j \leqslant D\} \subset \mathcal{E} \cup \mathcal{L}$ and thus $\boldsymbol{M}_r$ can be read off from* $\mathrm{nf}_\prec(\mathcal{E} \cup \mathcal{L})$.

Proof. The first claim follows from the above discussion, recalling that s is also the cardinality of $\mathcal{L}$. Indeed, $\mathrm{nf}_\prec(\mathcal{E})$ is obtained for free, and for each monomial f in $\mathcal{L}$, its normal form $\mathrm{nf}_\prec(f)$ is computed using at most D computations of opposites of elements of $\mathbb{K}$.

Now, assume $\mathcal{H}(\langle \mathrm{lt}_\prec(\mathcal{N}) \rangle)$ and suppose by contradiction that $X_r\varepsilon_j \notin \mathcal{E} \cup \mathcal{L}$ for some j. Since $X_r\varepsilon_j$ is not in $\mathcal{E}$, it is in $\mathrm{lt}_\prec(\mathcal{N})$, and thus it is a multiple $X_r\varepsilon_j = X_1^{\alpha_1} \cdots X_r^{\alpha_r} f$ of some $f \in \mathcal{L}$, for some exponents $\alpha_1, \ldots, \alpha_r$. Since $X_r\varepsilon_j \notin \mathcal{L}$, we have $\alpha_k > 0$ for some $1 \leqslant k \leqslant r$. If $\alpha_r > 0$, then $\varepsilon_j = X_1^{\alpha_1} \cdots X_r^{\alpha_r - 1} f \in \mathrm{lt}_\prec(\mathcal{N})$, which is absurd since $\varepsilon_j \in \mathcal{E}$; hence $k < r$ and $X_r\varepsilon_j = X_1^{\alpha_1} \cdots X_{r-1}^{\alpha_{r-1}} f$. But then $\frac{1}{X_k} X_r\varepsilon_j \in \mathrm{lt}_\prec(\mathcal{N})$, and using $\mathcal{H}(\langle \mathrm{lt}_\prec(\mathcal{N}) \rangle)$ we arrive at the same contradiction: $\frac{X_k}{X_r} \frac{1}{X_k} X_r\varepsilon_j = \varepsilon_j \in \mathrm{lt}_\prec(\mathcal{N})$. Therefore $X_r\varepsilon_j \in \mathcal{E} \cup \mathcal{L}$ for all j, which proves the inclusion in the lemma.

The last claim follows, since each row of $\boldsymbol{M}_r$ is the $\prec$-normal form of a monomial $X_r\varepsilon_j$, for some $1 \leqslant j \leqslant D$. □

Computing the remaining multiplication matrices requires to compute normal forms of monomials in the third set $\mathcal{B} - \mathcal{L}$, which is more involved. Our main algorithmic ingredient to compute those efficiently is a procedure which computes a collection of vector-matrix products of the form $\boldsymbol{v}\boldsymbol{M}^e$, where $\boldsymbol{M}$ is some $D \times D$ matrix; in our context $\boldsymbol{M}$ is one of the multiplication matrices that are already known at some point of the algorithm. We call this operation *Krylov evaluation* and

we give an algorithm for it in Section 4.2. The next example gives a simple illustration of how Krylov evaluation occurs in the computation of multiplication matrices.

Remark 4.2. Assume $\mathcal{N}$ is an ideal of $\mathbb{K}[\boldsymbol{X}] = \mathbb{K}[X_1, X_2]$, that is, $r = 2$ and $n = 1$. The above lemma shows how to compute $\boldsymbol{M}_2$ under the structural assumption. Having $\boldsymbol{M}_2$, we will now see how Krylov evaluation allows us to compute $\boldsymbol{M}_1$ using $O(D^\omega \log(D))$ operations in $\mathbb{K}$; hence, in this context, both multiplication matrices are obtained in this cost bound.

As explained above, the rows of $\boldsymbol{M}_1$ that correspond to normal forms in $\mathrm{nf}_\prec(\mathcal{E} \cup \mathcal{L})$ are found using $O(D^2)$ operations, since here $s \leqslant D$. Thus, it remains to compute its rows that are in $\mathrm{nf}_\prec(\mathcal{B}_1)$, where $\mathcal{B}_1 = \{X_1\varepsilon_j \mid 1 \leqslant j \leqslant D\} - (\mathcal{E} \cup \mathcal{L})$. Write $\mathcal{L} = \{X_1^{\alpha_j} X_2^{\beta_j} \mid 1 \leqslant j \leqslant s\}$ with $(\alpha_{j+1}, \beta_{j+1}) \prec_{\mathrm{lex}} (\alpha_j, \beta_j)$ for $1 \leqslant j < s$; since $\mathcal{L}$ is the minimal generating set of $\langle \mathrm{lt}_\prec(\mathcal{N}) \rangle$, (α_j) is decreasing with $\alpha_s = 0$ and (β_j) is increasing with $\beta_1 = 0$. Then $\mathcal{B}_1 = \{X_1^{\alpha_j} X_2^{\beta_j + k} \mid 1 \leqslant k < \beta_{j+1} - \beta_j, 1 \leqslant j < s\}$.

Now let $\boldsymbol{v}_j \in \mathbb{K}^{1 \times D}$ be the vector which represents $\mathrm{nf}_\prec(X_1^{\alpha_j} X_2^{\beta_j})$ in the basis $\mathcal{E}$; since $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_s\}$ represent $\mathrm{nf}_\prec(\mathcal{L})$, these vectors are among the rows of $\boldsymbol{M}_1$ that have already been computed. Then the vectors representing $\mathrm{nf}_\prec(\mathcal{B}_1)$ are

$$\{\boldsymbol{v}_j \boldsymbol{M}_2^k \mid 1 \leqslant k < \beta_{j+1} - \beta_j, 1 \leqslant j < s\}.$$

Performing this Krylov evaluation using the algorithm in Section 4.2 takes $O(D^\omega \log(D))$ operations in $\mathbb{K}$, as stated in Lemma 4.3 which involves parameters that are here $\sigma = \beta_s \leqslant D$ and $\mu \leqslant D$. □

More generally, for r variables and $n \geqslant 1$, one may similarly obtain $\boldsymbol{M}_{r-1}$ by computing such Krylov evaluations, assuming $\mathcal{H}(\langle \mathrm{lt}_\prec(\mathcal{N}) \rangle)$ (this is a consequence of the more general Lemmas 4.4 and 4.6). However, when $r > 2$ this does not extend into an iterative computation of the multiplication matrices: $\boldsymbol{M}_{r-2}$ cannot be obtained simply by Krylov evaluation with the matrix $\boldsymbol{M}_{r-1}$ and the normal forms in $\mathrm{nf}_\prec(\mathcal{E} \cup \mathcal{L})$ and those given by the rows of $\boldsymbol{M}_{r-1}$. The reason is that some of the normal forms which constitute the rows of $\boldsymbol{M}_{r-2}$ are actually obtained by Krylov evaluation with the matrix $\boldsymbol{M}_r$ and the normal forms in $\mathrm{nf}_\prec(\mathcal{L})$.

Thus we change our focus, from the computation of the multiplication matrices to that of the normal forms which we can obtained by Krylov evaluation with the known multiplication matrices and known normal forms. Roughly, our algorithm is as follows. The first iteration is for $i = r$ and considers $\mathcal{S}_r = \mathcal{E} \cup \mathcal{L}$, for which we have seen how to efficiently compute $\mathrm{nf}_\prec(\mathcal{S}_r)$. Our structural assumption ensures that these normal forms contain those giving $\boldsymbol{M}_r$. Then the iteration $i = r - 1$ considers the monomials $\mathcal{S}_{r-1}$ that can be obtained from $\mathcal{S}_r = \mathcal{E} \cup \mathcal{L}$ by multiplication by X_r, and their normal forms $\mathrm{nf}_\prec(\mathcal{S}_{r-1})$ are computed using Krylov evaluation with $\boldsymbol{M}_r$ and the vectors representing $\mathrm{nf}_\prec(\mathcal{S}_r)$. Again, our assumption ensures that $\mathrm{nf}_\prec(\mathcal{S}_{r-1})$ gives $\boldsymbol{M}_{r-1}$, but it also contains other normal forms which correspond to rows of multiplication matrices $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_{r-2}$, whose computation is not complete yet. Then we continue this process until $i = 1$: at this stage, we have covered the whole set of monomials $\mathcal{S}$ and we thus have all the normal forms in $\mathrm{nf}_\prec(\mathcal{S})$, from which we read the rows of the multiplication matrices. The algorithm is described in detail in Section 4.3.

4.2. Algorithm for Krylov evaluation

Now we give a simple method for the computation of a collection of vector-matrix products of the form $\boldsymbol{v}\boldsymbol{M}^e$, obtaining efficiency via repeated squaring of the matrix $\boldsymbol{M}$. This is detailed in Algorithm 4, in which we use the following conventions. When specified, instead of indexing

the rows of a matrix $\boldsymbol{K} \in \mathbb{K}^{\sigma \times D}$ using the integers $(1, \ldots, \sigma)$, we index them by a given totally ordered set $(\mathcal{P}, \leqslant)$ of cardinality σ. Explicitly, if $\mathcal{P} = \{\mathbf{e}_1, \ldots, \mathbf{e}_\sigma\}$ with $\mathbf{e}_1 \leqslant \cdots \leqslant \mathbf{e}_\sigma$, then the ith row of $\boldsymbol{K}$ has index $\mathbf{e}_i$. Then, for any subset $\mathcal{P}' \subseteq \mathcal{P}$, we write $\mathrm{Rows}(\boldsymbol{K}, \mathcal{P}')$ for the submatrix of $\boldsymbol{K}$ formed by its rows with indices in $\mathcal{P}'$. An assignment operation such as $\mathrm{Rows}(\boldsymbol{K}, \mathcal{P}') \leftarrow \boldsymbol{A}$ for some $\boldsymbol{A} \in \mathbb{K}^{\mathrm{Card}(\mathcal{P}') \times D}$ does modify the corresponding entries of $\boldsymbol{K}$.

Algorithm 4 – KrylovEval

Input:

- a matrix $\boldsymbol{M} \in \mathbb{K}^{D \times D}$ for some $D \in \mathbb{N}_{>0}$,
- row vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_t \in \mathbb{K}^{1 \times D}$ for some $t \in \mathbb{N}_{>0}$,
- bounds $e_1, \ldots, e_t \in \mathbb{N}_{>0}$.

Output: the matrix $\boldsymbol{K} \in \mathbb{K}^{(e_1 + \cdots + e_t) \times D}$ whose row $e_1 + \cdots + e_{j-1} + e$ is equal to $\boldsymbol{v}_j \boldsymbol{M}^e$, for $1 \leqslant e \leqslant e_j$ and $1 \leqslant j \leqslant t$.

1. `/* Initialize set of indices and output matrix */`
   $\mathcal{P} \leftarrow \{(e, j) \mid 1 \leqslant e \leqslant e_j, 1 \leqslant j \leqslant t\}$
   $\boldsymbol{K} \leftarrow \mathbf{0} \in \mathbb{K}^{(e_1 + \cdots + e_t) \times D}$ with its rows indexed by $(\mathcal{P}, \prec_{\mathrm{lex}})$
2. `/* Case` $e = 1$`: compute` $\boldsymbol{v}_j \boldsymbol{M}$ `for` $1 \leqslant j \leqslant t$ `*/`
   $\mathcal{P}' \leftarrow \{(1, j) \mid 1 \leqslant j \leqslant t\}$ // $\mathcal{P}' \subseteq \mathcal{P}$
   $\mathrm{Rows}(\boldsymbol{K}, \mathcal{P}') \leftarrow \begin{bmatrix} \boldsymbol{v}_1^\mathsf{T} & \cdots & \boldsymbol{v}_t^\mathsf{T} \end{bmatrix}^\mathsf{T} \boldsymbol{M}$
3. `/* Repeated squaring: handle` $e \in \{2^{i-1} + 1, \ldots, 2^i\}$ `for` $i > 1$ `*/`
   $\boldsymbol{N} \leftarrow \boldsymbol{M}$
   For i from 1 to $\lceil \log_2(\max_i e_i) \rceil$:
   - If $i > 1$ then $\boldsymbol{N} \leftarrow \boldsymbol{N}^2$ // $\boldsymbol{N} = \boldsymbol{M}^{2^{i-1}}$
   - $\mathcal{P}' \leftarrow \{(e, j) \mid 2^{i-1} < e \leqslant \min(e_j, 2^i), 1 \leqslant j \leqslant t\}$ // $\mathcal{P}' \subseteq \mathcal{P}$
   - $\mathcal{P}'' = \{(e - 2^{i-1}, j) \mid (e, j) \in \mathcal{P}'\}$ // $\mathcal{P}'' \subseteq \mathcal{P}$
   - $\mathrm{Rows}(\boldsymbol{K}, \mathcal{P}') \leftarrow \mathrm{Rows}(\boldsymbol{K}, \mathcal{P}'') \cdot \boldsymbol{N}$
4. Return $\boldsymbol{K}$

Lemma 4.3. *Given $\boldsymbol{M} \in \mathbb{K}^{D \times D}$, let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_t \in \mathbb{K}^{1 \times D}$, and let $e_1, \ldots, e_t \in \mathbb{N}_{>0}$, Algorithm 4 computes the row vectors $\{\boldsymbol{v}_j \boldsymbol{M}^e \mid 1 \leqslant e \leqslant e_j, 1 \leqslant j \leqslant t\}$ using*

$$O\left(D^\omega(1 + \log(\mu)) + D^{\omega-1}\sigma(1 + \log(\mu t/\sigma))\right) \subseteq O\left(D^{\omega-1}(D + \sigma)(1 + \log(\mu))\right)$$

operations in $\mathbb{K}$, where $\sigma = e_1 + \cdots + e_t$ and $\mu = \max(e_1, \ldots, e_t)$.

Proof. We want to prove that after Step 3, the row $e_1 + \cdots + e_{j-1} + e$ of $\boldsymbol{K}$ is $\boldsymbol{v}_j \boldsymbol{M}^e$, for $1 \leqslant e \leqslant e_j$ and $1 \leqslant j \leqslant t$. Indexing the rows of $\boldsymbol{K}$ by $(\mathcal{P}, \prec_{\mathrm{lex}})$ as in the algorithm, this means that the row of $\boldsymbol{K}$ at index (e, j) is $\boldsymbol{v}_j \boldsymbol{M}^e$, for all $(e, j) \in \mathcal{P}$. To prove this, we show that at the end of the ith iteration of the loop at Step 3, the following assertion holds:

$$\begin{aligned} \mathcal{A}_i : &\text{ the row of } \boldsymbol{K} \text{ at index } (e, j) \text{ is equal to } \boldsymbol{v}_j \boldsymbol{M}^e, \\ &\text{for all } (e, j) \in \mathcal{P} \text{ such that } e \leqslant 2^i. \end{aligned}$$

This gives the conclusion, since when the algorithm completes the loop, we have $i = \lceil \log_2(\mu) \rceil$ and $2^i \geqslant \mu$, and all $(e, j) \in \mathcal{P}$ are such that $e \leqslant \mu$ by definition of μ.

First, $(1, j) \in \mathcal{P}$ for $1 \leqslant j \leqslant t$, and after Step 2, the row of $\boldsymbol{K}$ at index $(1, j)$ is equal to $\boldsymbol{v}_j\boldsymbol{M}$. So $\mathcal{A}_0$ holds before the first iteration $i = 1$. Now, assume that $\mathcal{A}_{i-1}$ holds before the ith iteration: we want to prove that $\mathcal{A}_i$ holds at the end of this iteration. After the squaring at the beginning of the iteration, we have $\boldsymbol{N} = \boldsymbol{M}^{2^{i-1}}$. By construction, $\mathcal{P}'$ is the set of indices such that we have the equality

$$\mathcal{P}' \cup (\mathcal{P} \cap \{(e, j) \mid 1 \leqslant e \leqslant 2^{i-1}, 1 \leqslant j \leqslant t\}) = \mathcal{P} \cap \{(e, j) \mid 1 \leqslant e \leqslant 2^i, 1 \leqslant j \leqslant t\}.$$

Thus our goal is to show that for each $(e, j) \in \mathcal{P}'$, at the end of the iteration the row of $\boldsymbol{K}$ at index (e, j) is $\boldsymbol{v}_j\boldsymbol{M}^e$. By assumption, the row of $\boldsymbol{K}$ at index $(e - 2^{i-1}, j)$ is $\boldsymbol{v}_j\boldsymbol{M}^{e-2^{i-1}}$. Then the last step of the iteration ensures that the row of $\boldsymbol{K}$ at index (e, j) is $\boldsymbol{v}_j\boldsymbol{M}^{e-2^{i-1}}\boldsymbol{N} = \boldsymbol{v}_j\boldsymbol{M}^{e-2^{i-1}}\boldsymbol{M}^{2^{i-1}} = \boldsymbol{v}_j\boldsymbol{M}^e$. Thus, $\mathcal{A}_i$ holds, and this concludes the proof of correctness.

At Step 2, we multiply an $t \times D$ matrix and a $D \times D$ matrix, using $O(D^\omega + D^{\omega-1}t)$ operations in $\mathbb{K}$; this is within the bound in the lemma since $t \leqslant \sigma$. Over all iterations of the loop at Step 3, the squarings of $\boldsymbol{N}$ use a total of $O(D^\omega(\lceil \log_2(\mu) \rceil - 1)) \subseteq O(D^\omega \log_2(\mu))$ operations. The product $\mathrm{Rows}(\boldsymbol{K}, \mathcal{P}'') \cdot \boldsymbol{N}$ is computed using $O(D^\omega + D^{\omega-1}\mathrm{Card}(\mathcal{P}''))$ operations in $\mathbb{K}$ since the matrices have size $\mathrm{Card}(\mathcal{P}'') \times D$ and $D \times D$, respectively. By definition, $\mathrm{Card}(\mathcal{P}'') = \mathrm{Card}(\mathcal{P}')$, and since $\mathcal{P}' = \mathcal{P} \cap \{(e, j) \mid 2^{i-1} < e \leqslant 2^i, 1 \leqslant j \leqslant t\}$, we obtain $\mathrm{Card}(\mathcal{P}'') \leqslant \mathrm{Card}(\mathcal{P}) = \sigma$ and $\mathrm{Card}(\mathcal{P}'') \leqslant 2^{i-1}t$. Then the total number of operations in $\mathbb{K}$ used for the computation of $\mathrm{Rows}(\boldsymbol{K}, \mathcal{P}'') \cdot \boldsymbol{N}$ over all iterations of the loop at Step 3 is

$$O\left(\sum_{i=1}^{\lceil \log_2(\mu) \rceil} (D^\omega + D^{\omega-1} \min(2^{i-1}t, \sigma))\right) \subseteq O\left(D^\omega \lceil \log_2(\mu) \rceil + D^{\omega-1} \sum_{i=1}^{\lceil \log_2(\mu) \rceil} \min(2^{i-1}t, \sigma)\right),$$

which is within the cost bound in the lemma. Indeed, $\lceil \log_2(\mu) \rceil \leqslant 1 + \log_2(\mu)$ and

$$\begin{aligned}\sum_{i=1}^{\lceil \log_2(\mu) \rceil} \min(2^{i-1}t, \sigma) \leqslant \sum_{i=1}^{\lfloor \log_2(\sigma/t) \rfloor} 2^{i-1}t + \sum_{i=\lfloor \log_2(\sigma/t) \rfloor + 1}^{\lceil \log_2(\mu) \rceil} \sigma &\leqslant \sigma(1 + \lceil \log_2(\mu) \rceil - \lfloor \log_2(\sigma/t) \rfloor) \\ &\leqslant \sigma(3 + \log_2(\mu t/\sigma)).\end{aligned}$$

□

4.3. *Computing the multiplication matrices*

Now, we describe our algorithm for computing the multiplication matrices and give a complexity analysis. We follow on from notation in Section 4.1.

We are going to show how to compute the normal forms of all monomials in

$$\mathcal{E} \cup \mathcal{B} = \mathcal{S} \cup \{\boldsymbol{c}_i \mid 1 \leqslant i \leqslant n \text{ such that } \boldsymbol{c}_i \notin \mathrm{lt}_\prec(\mathcal{N})\};$$

since this set contains $\mathcal{S}$, this directly yields the multiplication matrices. We design an iteration on the r variables which computes the normal forms of r sets $\mathcal{E} \cup \mathcal{L} = \hat{\mathcal{S}}_r \subseteq \hat{\mathcal{S}}_{r-1} \subseteq \cdots \subseteq \hat{\mathcal{S}}_1 = \mathcal{E} \cup \mathcal{B}$; at the end, $\mathrm{nf}_\prec(\hat{\mathcal{S}}_1)$ gives the sought normal forms.

Thus, we start with the normal forms of the monomials in $\hat{\mathcal{S}}_r = \mathcal{E} \cup \mathcal{L}$, which are easily found from $\mathcal{G}$ (see Lemma 4.1). Then, for $1 \leqslant i < r$, we consider the monomials in $\mathcal{E} \cup \mathcal{B}$ which are obtained from $\mathcal{E} \cup \mathcal{L}$ through multiplication by $X_{i+1}, \ldots, X_r$:

$$\hat{\mathcal{S}}_i = \{X_{i+1}^{e_{i+1}} \cdots X_r^{e_r} f \mid e_{i+1}, \ldots, e_r \geqslant 0, f \in \mathcal{E} \cup \mathcal{L}\} \cap (\mathcal{E} \cup \mathcal{B}).$$

The normal forms $\mathrm{nf}_\prec(\hat{\mathcal{S}}_i)$ can be obtained from those in $\mathrm{nf}_\prec(\mathcal{E} \cup \mathcal{L})$ through multiplication by $\boldsymbol{M}_{i+1}, \ldots, \boldsymbol{M}_r$, if these matrices are known.

From these sets, we define the sets mentioned at the end of Section 4.1: $\mathcal{S}_r = \hat{\mathcal{S}}_r = \mathcal{E} \cup \mathcal{L}$ for $i = r$, and $\mathcal{S}_i = \hat{\mathcal{S}}_i - \hat{\mathcal{S}}_{i+1}$ for $1 \leqslant i < r$. Therefore $\hat{\mathcal{S}}_i$ is the disjoint union $\mathcal{S}_i \cup \cdots \cup \mathcal{S}_r$, and $\mathcal{S}_i$ is the set of monomials in $\mathcal{B} - \hat{\mathcal{S}}_{i+1}$ which can be obtained from $\mathcal{E} \cup \mathcal{L}$ through multiplication by a monomial in $X_{i+1}, \ldots, X_r$ which does involve the variable X_{i+1}. That is,

$$\begin{aligned}\mathcal{S}_i &= \{X_{i+1}^{e_{i+1}} \cdots X_r^{e_r} f \mid e_{i+1} > 0, e_{i+2}, \ldots, e_r \geqslant 0, f \in \mathcal{E} \cup \mathcal{L}\} \cap (\mathcal{B} - \hat{\mathcal{S}}_{i+1}) \\ &= \{X_{i+1}^{e} f \mid e > 0, f \in \hat{\mathcal{S}}_{i+1}\} \cap (\mathcal{B} - \hat{\mathcal{S}}_{i+1}).\end{aligned}$$

In particular, if $\boldsymbol{M}_{i+1}$ and $\mathrm{nf}_{\prec}(\hat{\mathcal{S}}_{i+1})$ are known, then $\mathrm{nf}_{\prec}(\mathcal{S}_i)$ can be computed via Krylov evaluation with the matrix $\boldsymbol{M}_{i+1}$ and the vectors representing $\mathrm{nf}_{\prec}(\hat{\mathcal{S}}_{i+1})$. Having $\mathrm{nf}_{\prec}(\mathcal{S}_i)$ gives us $\mathrm{nf}_{\prec}(\hat{\mathcal{S}}_i) = \mathrm{nf}_{\prec}(\mathcal{S}_i) \cup \mathrm{nf}_{\prec}(\hat{\mathcal{S}}_{i+1})$, and we will prove in Lemma 4.4 that $\boldsymbol{M}_i$ can be read off from $\mathrm{nf}_{\prec}(\hat{\mathcal{S}}_i)$, under the structural assumption. Thus we can proceed iteratively, since then, from $\boldsymbol{M}_i$ and $\mathrm{nf}_{\prec}(\hat{\mathcal{S}}_i)$ we can use Krylov evaluation to find $\mathrm{nf}_{\prec}(\mathcal{S}_{i-1})$, from which we deduce $\boldsymbol{M}_{i-1}$, etc. At the end of this process we have computed $\mathrm{nf}_{\prec}(\hat{\mathcal{S}}_1) \supseteq \mathrm{nf}_{\prec}(\mathcal{S})$ and deduced all the multiplication matrices.

Lemma 4.4. *Assuming $\mathcal{H}(\langle \mathrm{lt}_{\prec}(\mathcal{N}) \rangle)$, we have*

$$\{X_i \varepsilon_j \mid 1 \leqslant j \leqslant D\} \quad \subseteq \quad \hat{\mathcal{S}}_i \quad \textit{for all} \;\; 1 \leqslant i \leqslant r;$$

in particular, the multiplication matrices $\boldsymbol{M}_i, \ldots, \boldsymbol{M}_r$ can be read off from $\mathrm{nf}_{\prec}(\hat{\mathcal{S}}_i)$.

Proof. Note that for $i = r$, this result was already proved in Lemma 4.1 (and we will use similar arguments in the proof below); besides, for $i = 1$ it is straightforward since $\hat{\mathcal{S}}_1 = \mathcal{E} \cup \mathcal{B}$.

Let $i \in \{1, \ldots, r\}$. First, $\boldsymbol{M}_{i+1}, \ldots, \boldsymbol{M}_r$ can be read off from $\mathrm{nf}_{\prec}(\hat{\mathcal{S}}_i)$ since for $k \in \{i+1, \ldots, r\}$ we have $\{X_k \varepsilon_j, 1 \leqslant j \leqslant D\} \subseteq \hat{\mathcal{S}}_{k-1} \subseteq \hat{\mathcal{S}}_i$. The fact that $\boldsymbol{M}_i$ can be read off from $\mathrm{nf}_{\prec}(\hat{\mathcal{S}}_i)$ follows from the inclusion $\{X_i \varepsilon_j \mid 1 \leqslant j \leqslant D\} \subseteq \hat{\mathcal{S}}_i$ in the lemma; it remains to prove that inclusion.

Thus, we want to prove $X_i \varepsilon_j \in \hat{\mathcal{S}}_i$ for any $j \in \{1, \ldots, D\}$; for this we will use the structural assumption. The particular case $X_i \varepsilon_j \in \mathcal{E} \cup \mathcal{L}$ is obvious since $\mathcal{E} \cup \mathcal{L} = \mathcal{S}_r \subseteq \hat{\mathcal{S}}_i$. Now we consider $X_i \varepsilon_j \notin \mathcal{E} \cup \mathcal{L}$. Then $X_i \varepsilon_j \in \mathrm{lt}_{\prec}(\mathcal{N})$ and there exist exponents $\alpha_1, \ldots, \alpha_r$ not all zero and a monomial $f \in \mathcal{L}$ such that $X_i \varepsilon_j = X_1^{\alpha_1} \cdots X_r^{\alpha_r} f$.

Suppose by contradiction that there exists $k \in \{1, \ldots, i\}$ such that $\alpha_k > 0$. If $k = i$, then $\alpha_i > 0$ implies that ε_j is a multiple of f, hence $\varepsilon_j \in \mathrm{lt}_{\prec}(\mathcal{N})$, which is not the case. If $1 \leqslant k < i$, $\alpha_k > 0$ implies $\frac{1}{X_k} X_i \varepsilon_j \in \mathrm{lt}_{\prec}(\mathcal{N})$, and using the structural assumption we obtain the same contradiction: $\frac{X_k}{X_i} \frac{1}{X_k} X_i \varepsilon_j = \varepsilon_j \in \mathrm{lt}_{\prec}(\mathcal{N})$. Thus there is no such k, and $\alpha_1 = \cdots = \alpha_i = 0$. As a result, $X_i \varepsilon_j = X_{i+1}^{\alpha_{i+1}} \cdots X_r^{\alpha_r} f$, which is in $\hat{\mathcal{S}}_i$. □

For completeness, in Algorithm 5 we describe a straightforward subroutine for determining the sets of monomials $(\mathcal{S}_i)_{1 \leqslant i \leqslant r}$, which is directly based on the description of these sets in Lemma 4.5. Note that this computation does not involve field operations but only comparisons of exponents of monomials, so that here the time for finding these sets is not taken into account in our cost bounds. In an efficient implementation of our algorithm for finding the multiplication matrices, one would rather compute these sets while building $\mathcal{B}$ from $\mathcal{G}$, and we believe that finding these sets should indeed be a negligible part of the running time of such an implementation.

Lemma 4.5. *Assume $\mathcal{H}(\langle \mathrm{lt}_{\prec}(\mathcal{N}) \rangle)$ and let $1 \leqslant i < r$. Let*

$$\{f_1, \ldots, f_t\} = \{f \in \hat{\mathcal{S}}_{i+1} \cap \mathcal{B} \mid X_{i+1} f \in \mathcal{B} - \hat{\mathcal{S}}_{i+1}\},$$

and for $1 \leqslant j \leqslant t$*, let* $e_j \in \mathbb{N}_{>0}$ *be the largest integer such that* $X_{i+1}^{e_j} f_j \in \mathcal{B} - \hat{\mathcal{S}}_{i+1}$*. Then*

$$\mathcal{S}_i = \{X_{i+1}^e f_j \mid 1 \leqslant e \leqslant e_j, 1 \leqslant j \leqslant t\}.$$

Proof. First, we prove that $\mathcal{S}_i$ contains the latter set. Let $1 \leqslant j \leqslant t$ and $1 \leqslant e \leqslant e_j$. Then f_j is in $\hat{\mathcal{S}}_{i+1}$ and $X_{i+1}^e f_j$ is in $\mathcal{B}$. Since $e > 0$, $X_{i+1}^e f_j$ is in $\hat{\mathcal{S}}_i$. Furthermore, $X_{i+1}^e f_j$ is not in $\hat{\mathcal{S}}_{i+1}$, so that it is in $\hat{\mathcal{S}}_i - \hat{\mathcal{S}}_{i+1} = \mathcal{S}_i$.

Now, let $g \in \mathcal{S}_i$. By definition, this means that g is in $\hat{\mathcal{S}}_i \subseteq \mathcal{E} \cup \mathcal{B}$, and g is not in $\hat{\mathcal{S}}_{i+1} \supseteq \mathcal{E} \cup \mathcal{L}$. In particular, g is in $\mathcal{B} - \hat{\mathcal{S}}_{i+1}$. Furthermore, we can write $g = X_{i+1}^e f$ for some $e > 0$ and $f \in \hat{\mathcal{S}}_{i+1}$. Then let e' be the smallest exponent such that $X_{i+1}^{e'} f \notin \hat{\mathcal{S}}_{i+1}$; we have $1 \leqslant e' \leqslant e$ since $f \in \hat{\mathcal{S}}_{i+1}$ and $g \notin \hat{\mathcal{S}}_{i+1}$. Let $f' = X_{i+1}^{e'-1} f \in \hat{\mathcal{S}}_{i+1}$; thus f' is in $\mathcal{B}$: if this was not the case, then f' would be in $\mathcal{E}$ and $X_{i+1} f'$ would be in $\hat{\mathcal{S}}_{i+1}$ according to Lemma 4.4. Furthermore, it is a property of the border that, since the multiple $g = X_{i+1}^{e-e'+1} f'$ is in $\mathcal{B}$, then $X_{i+1} f'$ is in $\mathcal{E} \cup \mathcal{B}$; yet $X_{i+1} f'$ is not in $\hat{\mathcal{S}}_{i+1}$ which contains $\mathcal{E}$, hence $X_{i+1} f' \in \mathcal{B} - \hat{\mathcal{S}}_{i+1}$. It follows that $f' = f_j$ for some $1 \leqslant j \leqslant t$. Thus we have $g = X_{i+1}^{e-e'+1} f_j$, with $e - e' + 1 \leqslant e_j$ by definition of e_j, which concludes the proof. □

Algorithm 5 – NextMonomials

Input:

- the border $\mathcal{B}$,
- the set of monomials $\hat{\mathcal{S}}_{i+1} = \mathcal{S}_{i+1} \cup \cdots \cup \mathcal{S}_r$ for some $1 \leqslant i < r$.

Output: the set of monomials $\mathcal{S}_i$, in the form $\mathcal{S}_i = \{X_{i+1}^e f_j \mid 1 \leqslant e \leqslant e_j, 1 \leqslant j \leqslant t\}$ for some $f_1, \ldots, f_t \in \hat{\mathcal{S}}_{i+1} \cap \mathcal{B}$ and $e_1, \ldots, e_t \in \mathbb{N}_{>0}$.

1. $\mathcal{S}_i \leftarrow \emptyset; n \leftarrow 0$
2. For each $f \in \hat{\mathcal{S}}_{i+1} \cap \mathcal{B}$ such that $X_{i+1} f \in \mathcal{B} - \hat{\mathcal{S}}_{i+1}$:
   $e \leftarrow 1$; While $X_{i+1}^{e+1} f \in \mathcal{B} - \hat{\mathcal{S}}_{i+1}$: $e \leftarrow e + 1$
   $\mathcal{S}_i \leftarrow \mathcal{S}_i \cup \{X_{i+1} f, \ldots, X_{i+1}^e f\}$
   $t \leftarrow t + 1; e_t \leftarrow e; f_t \leftarrow f$
3. Return $\mathcal{S}_i = \{X_{i+1}^e f_j \mid 1 \leqslant e \leqslant e_j, 1 \leqslant j \leqslant t\}$

Next, we show how to compute $\mathrm{nf}_{\prec}(\mathcal{S}_i)$ from $\mathrm{nf}_{\prec}(\hat{\mathcal{S}}_{i+1})$ and $\boldsymbol{M}_{i+1}$ using Krylov evaluation.

Lemma 4.6. *Let* $i \in \{1, \ldots, r-1\}$*. Given* $(\mathcal{B}, \hat{\mathcal{S}}_{i+1}, \mathrm{nf}_{\prec}(\hat{\mathcal{S}}_{i+1}), \boldsymbol{M}_{i+1})$*, one can compute* $\mathcal{S}_i$ *and* $\mathrm{nf}_{\prec}(\mathcal{S}_i)$ *as follows:*

- $\mathcal{S}_i = \{X_{i+1}^e f_j \mid 1 \leqslant e \leqslant e_j, 1 \leqslant j \leqslant t_i\} \leftarrow$ NextMonomials$(\mathcal{B}, \hat{\mathcal{S}}_{i+1})$*, for some* $f_1, \ldots, f_{t_i} \in \hat{\mathcal{S}}_{i+1} \cap \mathcal{B}$ *and* $e_1, \ldots, e_{t_i} \in \mathbb{N}_{>0}$
- $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{t_i}\} \subseteq \mathbb{K}^{1 \times D} \leftarrow \mathrm{nf}_{\prec}(\{f_1, \ldots, f_{t_i}\})$*, retrieved from* $\mathrm{nf}_{\prec}(\hat{\mathcal{S}}_{i+1})$
  $\mathrm{nf}_{\prec}(\mathcal{S}_i) \leftarrow$ *rows of* KrylovEval$(\boldsymbol{M}_{i+1}, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{t_i}, e_1, \ldots, e_{t_i})$

This uses

$$O\left(D^\omega(1 + \log(\mu_i)) + D^{\omega-1}\sigma_i(1 + \log(\mu_i t_i/\sigma_i))\right) \subseteq O\left(D^{\omega-1}(D + \sigma_i)(1 + \log(\mu_i))\right)$$

operations in $\mathbb{K}$*, where* $\sigma_i = e_1 + \cdots + e_{t_i}$ *is the cardinality of* $\mathcal{S}_i$ *and* $\mu_i = \max(e_1, \ldots, e_{t_i})$*. We have* $\mu_i \leqslant \max\{e \in \mathbb{N} \mid X_{i+1}^e f \in \mathcal{B}$ *for some* $f \in \mathcal{B}\}$*.*

Proof. In the first step, $\mathcal{S}_i$ is determined from $\hat{\mathcal{S}}_{i+1}$ without field operation as shown in Algorithm 5, and it is obtained in the form $\mathcal{S}_i = \{X_{i+1}^e f_j \mid 1 \leqslant e \leqslant e_j, 1 \leqslant j \leqslant t_i\}$, where $f_1, \ldots, f_{t_i}$ are elements of $\hat{\mathcal{S}}_{i+1}$ and $e_1, \ldots, e_{t_i}$ are positive integers; in particular, $e_1 + \cdots + e_{t_i} = \mathrm{Card}(\mathcal{S}_i) = \sigma_i$. The upper bound on μ_i holds since, by construction of $\mathcal{S}_i$ as in Algorithm 5 (see also Lemma 4.5), $f_j \in \mathcal{B}$ and $X_{i+1}^{e_j} f_j \in \mathcal{B}$ for $1 \leqslant j \leqslant t_i$.

Going to the normal forms, we get

$$\mathrm{nf}_{\prec}(\mathcal{S}_i) = \{\boldsymbol{v}_j \boldsymbol{M}_{i+1}^e \mid 1 \leqslant e \leqslant e_j, 1 \leqslant j \leqslant t_i\} \tag{2}$$

where $\boldsymbol{v}_j = \mathrm{nf}_{\prec}(f_j) \in \mathbb{K}^{1\times D}$ for $1 \leqslant j \leqslant t_i$; these normal forms are already known since they are in $\mathrm{nf}_{\prec}(\hat{\mathcal{S}}_{i+1})$. This shows that the second item correctly computes $\mathrm{nf}_{\prec}(\mathcal{S}_i)$. The cost bound is a consequence of Lemma 4.3. □

Algorithm 6 – MultiplicationMatrices

Input:

- a monomial order $\prec$ on $\mathbb{K}[\boldsymbol{X}]^n$,
- a reduced $\prec$-Gröbner basis $\mathcal{G} \subseteq \mathbb{K}[\boldsymbol{X}]^n$.

Requirements:

- $\mathbb{K}[\boldsymbol{X}]^n/\mathcal{N}$ has finite dimension D as a $\mathbb{K}$-vector space, where $\mathcal{N} = \langle \mathcal{G} \rangle$,
- $\mathcal{H}(\langle \mathrm{lt}_{\prec}(\mathcal{N}) \rangle)$ holds.

Output:

- the $\prec$-monomial basis $\mathcal{E}$ of $\mathbb{K}[\boldsymbol{X}]^n/\mathcal{N}$,
- the multiplication matrices of $X_1, \ldots, X_r$ in $\mathbb{K}[\boldsymbol{X}]^n/\mathcal{N}$ with respect to $\mathcal{E}$.

1. `/* Build main sets of exponents */`
Read $\mathcal{L}$ and $\mathcal{E} = (\varepsilon_1, \ldots, \varepsilon_D)$ from $\mathcal{G}$, with $\varepsilon_1 \prec \cdots \prec \varepsilon_D$
$\mathcal{S} \leftarrow \{X_k \varepsilon_j \mid 1 \leqslant k \leqslant r, 1 \leqslant j \leqslant D\} \cup \{\boldsymbol{c}_i \mid 1 \leqslant i \leqslant n \text{ such that } \boldsymbol{c}_i \in \mathcal{L}\}$
$\mathcal{B} \leftarrow \mathcal{S} - \mathcal{E}$
`/* Below, normal forms are represented in` $\mathcal{E}$`, as vectors in` $\mathbb{K}^{1\times D}$ `*/`

2. `/* Initialize the iteration:` $\hat{\mathcal{S}} = \hat{\mathcal{S}}_r = \mathcal{E} \cup \mathcal{L}$`,` $\mathcal{Q} = \mathrm{nf}_{\prec}(\hat{\mathcal{S}}_{i+1})$`, find` $\boldsymbol{M}_r$ `*/`
$\hat{\mathcal{S}} \leftarrow \mathcal{E} \cup \mathcal{L}$; $\mathcal{Q} \leftarrow \mathrm{nf}_{\prec}(\hat{\mathcal{S}})$; read $\boldsymbol{M}_r$ from $\mathcal{Q}$

3. For i from $r-1$ to 1:
`/* Before iteration` i`:` $\hat{\mathcal{S}} = \hat{\mathcal{S}}_{i+1}$`,` $\mathcal{Q} = \mathrm{nf}_{\prec}(\hat{\mathcal{S}}_{i+1})$`,` $\boldsymbol{M}_{i+1}, \ldots, \boldsymbol{M}_r$ `known */`
`/* After iteration` i`:` $\hat{\mathcal{S}} = \hat{\mathcal{S}}_i$`,` $\mathcal{Q} = \mathrm{nf}_{\prec}(\hat{\mathcal{S}}_i)$`,` $\boldsymbol{M}_i, \ldots, \boldsymbol{M}_r$ `known */`

a. $\widetilde{\mathcal{S}} = \{X_{i+1}^e f_j \mid 1 \leqslant e \leqslant e_j, 1 \leqslant j \leqslant t\} \leftarrow$ NextMonomials$(\mathcal{B}, \hat{\mathcal{S}})$, for some $f_1, \ldots, f_t \in \hat{\mathcal{S}}$ and $e_1, \ldots, e_t \in \mathbb{N}_{>0}$ // $\widetilde{\mathcal{S}} = \mathcal{S}_i$

b. $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_t\} \subseteq \mathbb{K}^{1\times D} \leftarrow \mathrm{nf}_{\prec}(\{f_1, \ldots, f_t\})$, retrieved from $\mathcal{Q} = \mathrm{nf}_{\prec}(\hat{\mathcal{S}})$
$\widetilde{\mathcal{Q}} \leftarrow$ rows of KrylovEval$(\boldsymbol{M}_{i+1}, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_t, e_1, \ldots, e_t)$ // $\widetilde{\mathcal{Q}} = \mathrm{nf}_{\prec}(\mathcal{S}_i)$

c. $\hat{\mathcal{S}} \leftarrow \widetilde{\mathcal{S}} \cup \hat{\mathcal{S}}$; $\mathcal{Q} \leftarrow \widetilde{\mathcal{Q}} \cup \mathcal{Q}$; read $\boldsymbol{M}_i$ from $\hat{\mathcal{S}}$

4. Return $\mathcal{E}, \boldsymbol{M}_1, \ldots, \boldsymbol{M}_r$

The correctness of Algorithm 6 follows from the results and discussions in this section. The next proposition implies the first item of Theorem 1.9, and gives a more precise cost bound. It uses notation from Lemma 4.6. We remark that one could easily verify that the requirements

of Algorithm 6 hold while building the $\prec$-monomial basis $\mathcal{E}$ at the first step, relying on the characterization of $\mathcal{H}(\langle \mathrm{lt}_\prec(\mathcal{N})\rangle)$ described in Lemma 2.2.

Proposition 4.7. *Let $\prec$ be a monomial order on $\mathbb{K}[\boldsymbol{X}]^n$ and let $\mathcal{G}$ be a reduced $\prec$-Gröbner basis such that $\mathbb{K}[\boldsymbol{X}]^n/\mathcal{N}$ has dimension D, where $\mathcal{N} = \langle \mathcal{G}\rangle$. Assume $\mathcal{H}(\langle \mathrm{lt}_\prec(\mathcal{N})\rangle)$, and using notation above, let $\mu = \max(\mu_1, \ldots, \mu_{r-1})$. Thus*

$$\mu \leqslant \max\{e \in \mathbb{N} \mid X_i^e f \in \mathcal{B} \text{ for some } f \in \mathcal{B} \text{ and some } 2 \leqslant i \leqslant r\},$$

and in particular $\mu \leqslant D$. Then Algorithm 6 solves Problem 2 using

$$\begin{aligned} &O\left(D^\omega\,(r-1+\log(\mu_1\cdots\mu_{r-1})) + D^{\omega-1}\left(\mathrm{Card}(\mathcal{B}-\mathcal{L}) + \textstyle\sum_{1\leqslant i\leqslant r}\sigma_i\log(\mu_i t_i/\sigma_i)\right)\right)\\ &\subseteq\ O\left(rD^\omega(1+\log(\mu))\right)\ \subseteq\ O\left(rD^\omega\log(D)\right)\end{aligned}$$

operations in $\mathbb{K}$.

Proof. First, Step 2 computes $\hat{\mathcal{S}}_r = \mathcal{S}_r = \mathcal{E}\cup\mathcal{L}$ and $\mathrm{nf}_\prec(\hat{\mathcal{S}}_r)$ from $\mathcal{G}$, using $O(rD^2)$ computations of opposites of field elements (see Lemma 4.1). Then the For loop iteratively applies Lemma 4.6 to obtain the remaining matrices. Using notation from Lemma 4.6, the overall cost bound is

$$\begin{aligned} &O\left(\sum_{1\leqslant i\leqslant r-1} D^\omega(1+\log(\mu_i)) + D^{\omega-1}\sigma_i(1+\log(\mu_i t_i/\sigma_i))\right)\\ &\subseteq O\left(D^\omega(r-1+\log(\mu_1\cdots\mu_{r-1})) + D^{\omega-1}\left(\mathrm{Card}(\mathcal{B}-\mathcal{L}) + \sum_{1\leqslant i\leqslant r-1}\sigma_i\log(\mu_i t_i/\sigma_i)\right)\right).\end{aligned}$$

Indeed, we have $\sigma_1+\cdots+\sigma_{r-1} = \mathrm{Card}(\mathcal{B}-\mathcal{L})$, since $\sigma_i = \mathrm{Card}(\mathcal{S}_i)$ and $\mathcal{S}_1\cup\cdots\cup\mathcal{S}_{r-1} = \hat{\mathcal{S}}_1 - \mathcal{S}_r = (\mathcal{E}\cup\mathcal{B}) - (\mathcal{E}\cup\mathcal{L}) = \mathcal{B}-\mathcal{L}$. Using the bounds $\mathrm{Card}(\mathcal{B}-\mathcal{L}) \leqslant rD$ (see Section 2) as well as $\mu_i t_i/\sigma_i \leqslant \mu_i \leqslant \mu$ for all i, we obtain the simplified cost bound $O(rD^\omega(1+\log(\mu)))$. □

4.4. Change of order

Combining the above algorithms leads to an efficient change of order procedure, detailed in Algorithm 7.

As above concerning the computation of multiplication matrices, one may easily verify from the input of Algorithm 7 whether the requirements hold. For simplicity, here we only use the simplified cost bounds of the above results; better bounds may be obtained in particular cases.

Proposition 4.8. *Algorithm 7 is correct and uses $O(nD^{\omega-1} + rD^\omega\log(D))$ operations in $\mathbb{K}$.*

Proof. According to Proposition 4.7, Step 1 uses $O(rD^\omega\log(D))$ operations in $\mathbb{K}$ and returns the $\prec$-monomial basis $\mathcal{E}$ of $\mathbb{K}[\boldsymbol{X}]^n/\mathcal{N}$ and the multiplication matrices $\boldsymbol{M}$ with respect to $\mathcal{E}$. To build the matrix $\boldsymbol{F} \in \mathbb{K}^{n\times D}$, Step 2 uses $O(nD)$ operations; precisely, each normal form of a coordinate vector in the Else statement costs at most D computations of the opposite of an element in $\mathbb{K}$. By Proposition 3.14, Step 3 uses $O(nD^{\omega-1} + rD^\omega\log(D))$ operations to compute the reduced $\prec_2$-Gröbner basis $\mathcal{G}_2$ of $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F})$. Hence the overall cost bound. Proving correctness amounts to showing that $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F}) = \mathcal{N}$, which directly follows from the construction of $\boldsymbol{F}$ and the fact that

Algorithm 7 – ChangeOrder

Input:

- a monomial order $\prec_1$ on $\mathbb{K}[\boldsymbol{X}]^n$,
- a reduced $\prec_1$-Gröbner basis $\mathcal{G}_1 \subseteq \mathbb{K}[\boldsymbol{X}]^n$,
- a monomial order $\prec_2$ on $\mathbb{K}[\boldsymbol{X}]^n$.

Requirements:

- $\mathbb{K}[\boldsymbol{X}]^n/\mathcal{N}$ has finite dimension D as a $\mathbb{K}$-vector space, where $\mathcal{N} = \langle \mathcal{G}_1 \rangle$,
- $\mathcal{H}(\langle \mathrm{lt}_{\prec_1}(\mathcal{N}) \rangle)$ holds.

Output:

- the reduced $\prec_2$-Gröbner basis of $\mathcal{N}$.

1. $\mathcal{E} = (\varepsilon_1, \ldots, \varepsilon_D)$, $\boldsymbol{M} = (\boldsymbol{M}_1, \ldots, \boldsymbol{M}_r) \leftarrow$ MultiplicationMatrices$(\prec_1, \mathcal{G}_1)$

2. `/* Build a matrix` $\boldsymbol{F}$ `such that` $\mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F}) = \mathcal{N}$ `*/`

$\boldsymbol{F} \leftarrow$ matrix in $\mathbb{K}^{n \times D}$

For $1 \leqslant i \leqslant n$:

If $\boldsymbol{c}_i = \varepsilon_j$ for some $1 \leqslant j \leqslant D$:

ith row of $\boldsymbol{F} \leftarrow [0 \ \cdots \ 0 \ 1 \ 0 \ \cdots \ 0] \in \mathbb{K}^{1 \times D}$ with 1 at index j

Else: `/* in this case` $\boldsymbol{c}_i - \mathrm{nf}_{\prec_1}(\boldsymbol{c}_i) \in \mathcal{G}_1$ `*/`

ith row of $\boldsymbol{F} \leftarrow$ vector in $\mathbb{K}^{1 \times D}$ representing $\mathrm{nf}_{\prec_1}(\boldsymbol{c}_i)$ on the basis $\mathcal{E}$

3. $\mathcal{G}_2 \leftarrow$ SyzygyModuleBasis$(\prec_2, \boldsymbol{M}, \boldsymbol{F})$

4. Return $\mathcal{G}_2$

$\boldsymbol{c}_i - \mathrm{nf}_{\prec_1}(\boldsymbol{c}_i)$ is in $\mathcal{N}$:

$$\begin{aligned}(p_1, \ldots, p_n) \in \mathrm{Syz}_{\boldsymbol{M}}(\boldsymbol{F}) \ &\Leftrightarrow \ \sum_{\substack{1 \leqslant i \leqslant n \\ \boldsymbol{c}_i \in \mathcal{E}}} p_i \boldsymbol{c}_i + \sum_{\substack{1 \leqslant i \leqslant n \\ \boldsymbol{c}_i \notin \mathcal{E}}} p_i \mathrm{nf}_{\prec_1}(\boldsymbol{c}_i) \ \in \mathcal{N} \\ &\Leftrightarrow \ \sum_{1 \leqslant i \leqslant n} p_i \boldsymbol{c}_i = (p_1, \ldots, p_n) \in \mathcal{N}. \qquad \square\end{aligned}$$

References

[1] Alonso, M.E., Marinari, M.G., Mora, T., 2003. The Big Mother of all Dualities: Möller Algorithm. Communications in Algebra 31, 783–818. doi:10.1081/AGB-120017343.

[2] Auzinger, W., Stetter, H.J., 1988. An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations, in: Proceedings Numerical Mathematics 1988, Birkhäuser Basel, Basel. pp. 11–30. doi:10.1007/978-3-0348-6303-2_2.

[3] Bayer, D., Stillman, M., 1987. A theorem on refining division orders by the reverse lexicographic order. Duke Mathematical Journal 55, 321–328. doi:10.1215/S0012-7094-87-05517-7.

[4] Beckermann, B., 1992. A reliable method for computing M-Padé approximants on arbitrary staircases. J. Comput. Appl. Math. 40, 19–42. doi:10.1016/0377-0427(92)90039-Z.

[5] Beckermann, B., Labahn, G., 1994. A uniform approach for the fast computation of matrix-type Padé approximants. SIAM J. Matrix Anal. Appl. 15, 804–823. doi:10.1137/S0895479892230031.

[6] Beckermann, B., Labahn, G., 1997. Recursiveness in matrix rational interpolation problems. J. Comput. Appl. Math. 77, 5–34. doi:10.1016/S0377-0427(96)00120-3.

[7] Beckermann, B., Labahn, G., 2000. Fraction-free computation of matrix rational interpolants and matrix gcds. SIAM J. Matrix Anal. Appl. 22, 114–144. doi:10.1137/S0895479897326912.

[8] Berthomieu, J., Boyer, B., Faugère, J.C., 2015. Linear algebra for computing gröbner bases of linear recursive multidimensional sequences, in: ISSAC'15, ACM, New York, NY, USA. pp. 61–68. doi:10.1145/2755996.2756673.

[9] Berthomieu, J., Boyer, B., Faugère, J.C., 2017. Linear Algebra for Computing Gröbner Bases of Linear Recursive Multidimensional Sequences. J. Symbolic Comput. 83, 36–67. doi:10.1016/j.jsc.2016.11.005.

[10] Berthomieu, J., Faugère, J.C., 2016. Guessing linear recurrence relations of sequence tuples and p-recursive sequences with linear algebra, in: ISSAC'16, ACM, New York, NY, USA. pp. 95–102. doi:10.1145/2930889.2930926.

[11] Berthomieu, J., Faugère, J.C., 2018. A polynomial-division-based algorithm for computing linear recurrence relations, ACM, New York, NY, USA. pp. 79–86. doi:10.1145/3208976.3209017.

[12] Coppersmith, D., Winograd, S., 1990. Matrix multiplication via arithmetic progressions. J. Symbolic Comput. 9, 251–280. doi:10.1016/S0747-7171(08)80013-2.

[13] Eisenbud, D., 1995. Commutative Algebra: with a View Toward Algebraic Geometry. Graduate Texts in Mathematics, Springer, New York, Berlin, Heildelberg. doi:10.1007/978-1-4612-5350-1.

[14] Faugère, J., Gaudry, P., Huot, L., Renault, G., 2013. Polynomial systems solving by fast linear algebra. CoRR abs/1304.6039. URL: http://arxiv.org/abs/1304.6039.

[15] Faugère, J.C., Gaudry, P., Huot, L., Renault, G., 2014. Sub-cubic change of ordering for Gröbner basis: a probabilistic approach, in: ISSAC'14, ACM, New York, NY, USA. pp. 170–177. doi:10.1145/2608628.2608669.

[16] Faugère, J.C., Gianni, P., Lazard, D., Mora, T., 1993. Efficient computation of zero-dimensional Gröbner bases by change of ordering. Journal of Symbolic Computation 16, 329–344. doi:10.1006/jsco.1993.1051.

[17] Faugère, J.C., Mou, C., 2011. Fast algorithm for change of ordering of zero-dimensional gröbner bases with sparse multiplication matrices, in: ISSAC'11, ACM, New York, NY, USA. pp. 115–122. doi:10.1145/1993886.1993908.

[18] Faugère, J.C., Mou, C., 2017. Sparse FGLM algorithms. J. Symbolic Comput. 80, Part 3, 538–569. doi:10.1016/j.jsc.2016.07.025.

[19] Fitzpatrick, P., 1997. Solving a Multivariable Congruence by Change of Term Order. J. Symbolic Comput. 24, 575–589. doi:10.1006/jsco.1997.0153.

[20] Galligo, A., 1974. A propos du théorème de préparation de Weierstrass, in: Fonctions de Plusieurs Variables Complexes, Springer. pp. 543–579. doi:10.1007/BFb0068121.

[21] Giorgi, P., Jeannerod, C.P., Villard, G., 2003. On the complexity of polynomial matrix computations, in: ISSAC'03, ACM. pp. 135–142. doi:10.1145/860854.860889.

[22] Hermite, C., 1893. Sur la généralisation des fractions continues algébriques. Annali di Matematica Pura ed Applicata (1867-1897) 21, 289–308. doi:10.1007/BF02420446.

[23] Jeannerod, C.P., Neiger, V., Schost, E., Villard, G., 2016. Fast computation of minimal interpolation bases in Popov form for arbitrary shifts, in: ISSAC'16, ACM. pp. 295–302. doi:10.1145/2930889.2930928.

[24] Jeannerod, C.P., Neiger, V., Schost, E., Villard, G., 2017. Computing minimal interpolation bases. J. Symbolic Comput. 83, 272–314. doi:10.1016/j.jsc.2016.11.015.

[25] Jeannerod, C.P., Neiger, V., Villard, G., 2019. Fast computation of approximant bases in canonical form. J. Symbolic Comput. In press. doi:10.1016/j.jsc.2019.07.011.

[26] Kailath, T., 1980. Linear Systems. Prentice-Hall.

[27] Kehrein, A., Kreuzer, M., Robbiano, L., 2005. An algebraist's view on border bases, in: and Dickenstein, A., Emiris, I.Z. (Eds.), Solving Polynomial Equations: Foundations, Algorithms, and Applications. Springer, pp. 169–202. doi:10.1007/b138957.

[28] Keller-Gehrig, W., 1985. Fast algorithms for the characteristic polynomial. Theoretical Computer Science 36, 309–317. doi:10.1016/0304-3975(85)90049-0.

[29] Kojima, C., Rapisarda, P., Takaba, K., 2007. Canonical forms for polynomial and quadratic differential operators. Systems and Control Letters 56, 678–684. doi:10.1016/j.sysconle.2007.06.004.

[30] Le Gall, F., 2014. Powers of tensors and fast matrix multiplication, in: ISSAC'14, ACM. pp. 296–303. doi:10.1145/2608628.2608664.

[31] Macaulay, F.S., 1902. Some formulae in elimination. Proceedings of the London Mathematical Society s1-35, 3–27. doi:10.1112/plms/s1-35.1.3.

[32] Macaulay, F.S., 1916. The Algebraic Theory of Modular Systems. Cambridge Tracts in Mathematics and Mathematical Physics, Cambridge University Press.

[33] Mahler, K., 1968. Perfect systems. Composit. Math. 19, 95–166.

[34] Marinari, M.G., Möller, H.M., Mora, T., 1991. Gröbner bases of ideals given by dual bases, in: ISSAC'91, ACM, New York, NY, USA. pp. 55–63. doi:10.1145/120694.120702.

[35] Marinari, M.G., Möller, H.M., Mora, T., 1993. Gröbner bases of ideals defined by functionals with an application to ideals of projective points. Appl. Algebra Engrg. Comm. Comput. 4, 103–145. doi:10.1007/BF01386834.

[36] Möller, H.M., Buchberger, B., 1982. The construction of multivariate polynomials with preassigned zeros, in:

EUROCAM'82, Springer. pp. 24–31. doi:10.1007/3-540-11607-9_3.
[37] Mourrain, B., 1999. A new criterion for normal form algorithms, in: Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 430–442. doi:10.1007/3-540-46796-3_41.
[38] Neiger, V., 2016. Bases of relations in one or several variables: fast algorithms and applications. Ph.D. thesis. École Normale Supérieure de Lyon. URL: https://tel.archives-ouvertes.fr/tel-01431413/.
[39] O'Keeffe, H., Fitzpatrick, P., 2002. Gröbner basis solutions of constrained interpolation problems. Linear Algebra Appl. 351, 533–551. doi:10.1016/S0024-3795(01)00509-2.
[40] Padé, H., 1894. Sur la généralisation des fractions continues algébriques. Journal de Mathématiques Pures et Appliquées , 291–330.
[41] Pardue, K., 1994. Nonstandard Borel-Fixed Ideals. Ph.D. thesis. Brandeis University.
[42] Paszkowski, S., 1987. Recurrence relations in Padé-Hermite approximation. J. Comput. Appl. Math. 19, 99–107. doi:10.1016/0377-0427(87)90177-4.
[43] Popov, V.M., 1972. Invariant description of linear, time-invariant controllable systems. SIAM Journal on Control 10, 252–264. doi:10.1137/0310020.
[44] Sakata, S., 1990. Extension of the berlekamp-massey algorithm to n dimensions. Inform. and Comput. 84, 207–239. doi:10.1016/0890-5401(90)90039-K.
[45] Sergeyev, A.V., 1987. A recursive algorithm for Padé-Hermite approximations. USSR Comput. Math. Math. Phys. 26, 17–22. doi:10.1016/0041-5553(86)90003-0.
[46] Storjohann, A., 2000. Algorithms for Matrix Canonical Forms. Ph.D. thesis. Swiss Federal Institute of Technology – ETH. URL: https://cs.uwaterloo.ca/~astorjoh/diss2up.pdf.
[47] Sylvester, J.J., 1853. On a Theory of the Syzygetic Relations of Two Rational Integral Functions, Comprising an Application to the Theory of Sturm's Functions, and That of the Greatest Algebraical Common Measure. Philosophical Transactions of the Royal Society of London 143, 407–548. doi:10.1098/rstl.1853.0018.
[48] Van Barel, M., Bultheel, A., 1991. The computation of non-perfect Padé-Hermite approximants. Numer. Algorithms 1, 285–304. doi:10.1007/BF02142327.
[49] Van Barel, M., Bultheel, A., 1992. A general module theoretic framework for vector M-Padé and matrix rational interpolation. Numer. Algorithms 3, 451–462. doi:10.1007/BF02141952.