Guessing Gröbner Bases of Structured Ideals of Relations of Sequences

Jérémy Berthomieu, Mohab Safey El Din

Guessing Gröbner Bases of Structured Ideals of Relations of Sequences

Jérémy Berthomieu, Mohab Safey El Din

Sorbonne Université, CNRS, LIP6, *F-75005, Paris, France*

Abstract

Assuming sufficiently many terms of an n-dimensional table defined over a field are given, we aim at guessing the linear recurrence relations with either constant or polynomial coefficients they satisfy. In many applications, the table terms come along with a structure: for instance, they may be zero outside of a cone, they may be built from a Gröbner basis of an ideal invariant under the action of a finite group. Thus, we show how to take advantage of this structure to reduce both the number of table queries and the number of operations in the base field to recover the ideal of relations of the table. In applications like in combinatorics, where all these zero terms make us guess many fake relations, this allows us to drastically reduce these wrong guesses. These algorithms have been implemented and, experimentally, they let us handle examples that we could not manage otherwise.

Furthermore, we show which kind of cone and lattice structures are preserved by skew-polynomial multiplication. This allows us to speed the guessing of linear recurrence relations with polynomial coefficients up by computing sparse Gröbner bases or Gröbner bases of an ideal invariant under the action of a finite group in a ring of skew-polynomials.

Keywords: Linear recurrence relations, Gröbner bases, Symmetries, Change of orderings

1. Introduction

Problem statement and motivations. Given a sequence $\boldsymbol{v} = (v_{i_1,\ldots,i_n})_{i_1,\ldots,i_n \geq 0}$, we consider the *table* made of a finite subset of its terms. Computing or *guessing* linear recurrence relations satisfied by such a table is a fundamental problem in coding theory for cyclic codes [10, 27] of dimension $n \geq 1$, combinatorics and computer algebra for solving sparse linear systems, performing sparse polynomial interpolation, polynomial least-square approximation and Gröbner bases changes of orderings in $n \geq 1$ variables [22, 23]. Furthermore, computing these relations with polynomial coefficients in the indices allows us to predict the growth of its terms, to classify the differential nature of their generating series or to evaluate said generating series [33].

Depending on the context, an upper bound on the number of table terms might be known in order to guess these relations. For instance, in coding theory, this is related to the length and the

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

Email addresses: jeremy.berthomieu@lip6.fr (Jérémy Berthomieu), mohab.safey@lip6.fr (Mohab Safey El Din)

minimum distance of the code. In the Gröbner bases change of orderings application, an upper bound is given by the degree of the ideal and the number of variables. Whenever no upper bound is known, one is still restricted to only consider a finite number of table terms to guess the linear recurrence relations the table satisfies. Thus, some of these relations may be proven incorrect when tested with many more table terms; further such relations will be called *fake relations*. This happens for instance in combinatorics where the nature itself of the table may be unknown.

In many applications, the table comes with a structure. For instance, in combinatorics, for nD-space walks in the nonnegative orthant, $v_{i_0,i_1,\ldots,i_n}$ counts the number of ways to reach $(i_1,\ldots,i_n) \in \mathbb{N}^n$ in i_0 steps of size 1 [11, 12]. Therefore, $v_{i_0,i_1,\ldots,i_n}$ is trivially 0 outside the cone $i_1,\ldots,i_n \leq i_0$. Thus, computationwise, not considering these terms would reduce the size of the table and thus might be beneficial for guessing the linear recurrence relations satisfied by the table. Hence, the goal is to exploit this structure to both reduce the number of table queries and the number of operations to guess the Gröbner basis of the ideal of relations.

Prior results. We distinguish two cases: the one-dimensional case, where tables are with one index, and the multidimensional one, where tables have $n > 1$ indices.

In the one-dimensional case, given the first D terms of a table, the Berlekamp–Massey algorithm [3, 32] guesses the linear recurrence relations with constant coefficients of smaller order. Using fast extended Euclidean algorithm, this algorithm can do so in $O(\mathsf{M}(D)\log D)$ operations in the base field [14], where $\mathsf{M}(D) = O(D\log D\log\log D)$ [15] is a cost function for multiplying two univariate polynomials of degree at most D. Through Hermite-Padé approximants, the Beckermann–Labahn algorithm [1] can be used to guess several relations with polynomial coefficients including the one of minimal order. Let us notice that finding relations with polynomial coefficients is a special case of Hermite-Padé approximants for which the Beckermann–Labahn algorithm is not quasi-optimal in the input size.

In the multidimensional case, several algorithms were designed for guessing linear recurrence relations with constant coefficients satisfied by the first terms of the tables using linear algebra routines. For instance, the Berlekamp–Massey–Sakata algorithm [36–38], the Scalar-FGLM algorithm [4, 5] or the Artinian Gorenstein border basis algorithm [34]. Given sufficiently many terms, the first two return a Gröbner basis of the ideal of relations while the third one returns a border basis of this ideal. Furthermore, in [9] the authors designed an algorithm extending both the Berlekamp–Massey–Sakata and the Scalar-FGLM algorithms using polynomial arithmetic and in [8], they extended the Scalar-FGLM algorithm for guessing relations with polynomial coefficients. However, none of these algorithms were designed to take the structure of the table terms into account. Another classical technique is the "ansatz + linear system solving" approach for finding relations. Usually the ansatz allows the user to find a set of relations, then a post-processing is needed in order to recover, for instance, the Gröbner basis of the ideal spanned by the computed relations. Though, if the ansatz is far from being tight, then the linear system to solve might be equivalent to the one of the Scalar-FGLM algorithm.

Gröbner bases are the output of several algorithms for guessing linear recurrence relations and are a fundamental tool in polynomial systems solving. In many applications, polynomials systems come with a structure, for instance they span an ideal globally invariant under the action of a finite group G or their supports are in a cone. From the table viewpoint, these are related to only considering table terms lying either on a lattice [28] or in a cone.

In [25], the authors show that for such an ideal, Gröbner bases computations through the F_4 [18], F_5 [19] and FGLM [21] algorithms can be sped up with a factor depending on $|G|$, whenever the characteristic of the field of coefficients does not divide $|G|$. To do so, they essentially

perform $|G|$ parallel smaller computations. In particular for the FGLM algorithm, this factor is $|G|^2$, see [25, Theorem 10]. Likewise, in [39], the author proposed algorithms for computing Gröbner bases of symmetric ideals over the rationals or a finite field.

In [2, 24], the authors show that if C is a semi-group of $\mathbb{Z}^n$ containing 0 and no pair of opposite elements and if $f_1, \dots, f_s$ are polynomials with support in the corresponding monomial set $\mathcal{T}(C) \coloneqq \left\{x_1^{i_1} \cdots x_n^{i_n} \middle| (i_1, \dots, i_n) \in C\right\}$, then one can consider the ideal spanned by $f_1, \dots, f_s$ in the subalgebra of polynomials with support in $\mathcal{T}(C)$. Modifying classical Gröbner bases algorithms, they obtain a sparse Gröbner basis, a set of generators with support in $\mathcal{T}(C)$ of this ideal that behaves like a Gröbner basis. This allows them to speed Gröbner basis computations up by taking into account the sparsity of the union of the supports of the original generators of the ideal.

Main results. We design variants of the Scalar-FGLM algorithm which guess linear recurrence relations for an n-dimensional table $\boldsymbol{v}$, given as polynomials in $x_1, \dots, x_n$. The original algorithm is recalled in page 9.

We first prove that restraining the Scalar-FGLM algorithm to terms of a table lying on a cone makes it compute a sparse Gröbner basis of the ideal of relations of the table. More precisely, we obtain Theorem 3.2, a simplified version of which is as follows.

Theorem 1.1. *Let C be a semi-subgroup of $\mathbb{N}^n$ containing 0. Let $\prec$ be a monomial ordering. Let $T \subset \mathcal{T}(C)$ be a finite set of monomials ordered for $\prec$, such that for all $\mu_1, \mu_2 \in \mathcal{T}(C)$, if $\mu_1\mu_2 \in T$, then μ_1 and μ_2 are in T.*

Let $\boldsymbol{v}$ be a n-dimensional table with nonzero elements $v_{i_1,\dots,i_n}$ only if $(i_1, \dots, i_n) \in C$.

Then, if T is large enough, the output of the Scalar-FGLM *called on $\boldsymbol{v}$, T and $\prec$ is the reduced sparse Gröbner basis of the ideal of relations of $\boldsymbol{v}$ with support in $\mathcal{T}(C)$.*

Let us remark that this allows us to remove trivial constraints on the relations induced by the zero terms outside of the cone, yielding in practice many fewer guessed relations that eventually fail. On the one hand, as a byproduct, this allows us to reduce the number of table queries to guess the relations. For instance, for a subtable of the Gessel walk, using 3 491 table terms, we can guess 142 relations amongst which 136 are fake and only 6 are correct. On the other hand taking only table terms in a cone allows us to consider table terms much further, which in turn allow us to guess more relations. Indeed, with 3 010 terms in a cone of the same table, we guess 21 relations and all of them are correct. We refer to Table 1 for more details. Let us also notice that these fake relations may hide correct ones as their leading monomials could divide the leading monomials of correct relations.

In the next theorem, we now consider table terms lying on a lattice Λ and affine translates thereof. This allows us to design a parallel variant of the Scalar-FGLM algorithm, called the Lattice Scalar-FGLM algorithm and given in page 13. Assuming the fundamental domain of Λ has L integer points, this variant essentially deals with L sets of table terms of sizes roughly divided by L. The following theorem is a simplified version of Theorem 3.4.

Theorem 1.2. *Let Λ be a sublattice of $\mathbb{Z}^n$. Let $\prec$ be a monomial ordering. Let $T \subset \mathcal{T}$ be a finite set of monomials ordered for $\prec$, such that for all $\mu_1, \mu_2 \in \mathcal{T}$, if $\mu_1\mu_2 \in T$, then μ_1 and μ_2 are in T.*

Let $f_1, \dots, f_s$ be polynomials spanning a zero-dimensional ideal I of degree D such that for all $1 \leq i \leq s$, there exists $\boldsymbol{a} \in \mathbb{Z}^n$ such that the support of f_i is included in $\mathcal{T}(\boldsymbol{a} + \Lambda) \coloneqq \left\{x_1^{i_1} \cdots x_n^{i_n} \middle| (i_1, \dots, i_n) \in \boldsymbol{a} + \Lambda\right\}$.

Let $\boldsymbol{v}$ be a n-dimensional generic table whose ideal of relations is I.

Then, if T is large enough, then the output of the Lattice Scalar-FGLM *called on $\boldsymbol{v}$, T and $\prec$ is the reduced Gröbner basis of the ideal of relations of $\boldsymbol{v}$. Furthermore, each polynomial in this Gröbner basis has its support in a set $\mathcal{T}(\boldsymbol{a} + \Lambda)$.*

Finally, we also make an adaptive variant of the Lattice Scalar-FGLM algorithm, following what has been done for the Scalar-FGLM. This adaptive variant, given in page 21, aim at reducing the number of table queries using the shape of the staircase associated to the Gröbner basis of the ideal of relations of the table.

Structure of the paper. We first recall in Section 2 the classical connection between linear recurrence relations with polynomial coefficients and skew-polynomials in $2n$ variables. Then, we recall how using linear algebra routines on a special kind of matrix, a *multi-Hankel* one, the Scalar-FGLM algorithm, and its adaptive variant the Adaptive Scalar-FGLM algorithm, guesses linear recurrence relations.

In Section 3, we design variants of the Scalar-FGLM algorithm that take the table structure into account for guessing linear recurrence relations, then we prove Theorems 3.2 and 3.4. As an application, we provide a modification of the Sparse-FGLM algorithm [22, 23] whenever the ideal is globally invariant under the action of a finite group.

The same kind of variants of the Adaptive Scalar-FGLM algorithm are then designed, in Section 4. Likewise, we prove Theorem 4.2 in this section. Then, we show how one can perform skew-polynomial operations in order to preserve the cone and lattice structures of the support of the polynomials.

Finally, in Section 5, we report on our speedup using our `C` implementation of the Sparse-FGLM algorithm when the ideal is invariant under the action of a finite group. We also guess linear recurrence relations satisfied by nD-space walks with and without exploiting the cone structure of the table and then test further the guessed relations. We then report on how the cone structure allows us to guess fewer fake linear recurrence relations.

2. Preliminaries

2.1. Tables and relations

In all this paper, we take the convention that $0 \in \mathbb{N}$. For $n \in \mathbb{N}$, $n \geq 1$, we let $\boldsymbol{i} = (i_1, \ldots, i_n) \in \mathbb{N}^n$, $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{x}^{\boldsymbol{i}} = x_1^{i_1} \cdots x_n^{i_n}$. For a subset $\mathcal{S}$ of $\mathbb{N}^n$, we let $\mathcal{T}(\mathcal{S}) = \{\boldsymbol{x}^{\boldsymbol{s}} | \boldsymbol{s} \in \mathcal{S}\}$ be the set of monomials with exponents in $\mathcal{S}$. To ease the presentation, we let $\mathcal{T} \coloneqq \mathcal{T}(\mathbb{N}^n)$. Finally, for a polynomial $f = \sum_{\boldsymbol{s} \in \mathcal{S}} f_{\boldsymbol{s}} \boldsymbol{x}^{\boldsymbol{s}}$, we let $\operatorname{supp} f = \{\boldsymbol{s} \in \mathcal{S} | f_{\boldsymbol{s}} \neq 0\}$ be its support.

Let $\mathbb{K}$ be a field and $\boldsymbol{v} \in \mathbb{K}^{\mathbb{N}^n}$ be a n-indexed sequence with values in $\mathbb{K}$, that is $\boldsymbol{v} = (v_{i_1,\ldots,i_n})_{(i_1,\ldots,i_n) \in \mathbb{N}^n}$. There is a natural correspondence between finite linear combinations of terms of $\boldsymbol{v}$ and polynomials in $\mathbb{K}[x_1, \ldots, x_n]$. For $g = \sum_{\boldsymbol{s} \in \mathcal{S}} \gamma_{\boldsymbol{s}} \boldsymbol{x}^{\boldsymbol{s}}$, with $\mathcal{S}$ a finite subset of $\mathbb{N}^n$, we can write $[g]_{\boldsymbol{v}} \coloneqq \sum_{\boldsymbol{s} \in \mathcal{S}} \gamma_{\boldsymbol{s}} v_{\boldsymbol{s}}$. Hence shifting a relation by an index $\boldsymbol{i}$ comes down to multiplying the corresponding polynomial by $\boldsymbol{x}^{\boldsymbol{i}}$ since

$$\left[g\boldsymbol{x}^{\boldsymbol{i}}\right]_{\boldsymbol{v}} = \sum_{\boldsymbol{s} \in \mathcal{S}} \gamma_{\boldsymbol{s}} v_{\boldsymbol{s}+\boldsymbol{i}}.$$

In particular, a polynomial g defines a *linear recurrence relation with constant coefficients*, or *C-relation* for short, on $\boldsymbol{v}$ if, and only if, for all $\boldsymbol{i} \in \mathbb{N}^n$, $\left[g\boldsymbol{x}^{\boldsymbol{i}}\right]_{\boldsymbol{v}} = 0$. The set of all such polynomials is an ideal of $\mathbb{K}[\boldsymbol{x}]$ called the *ideal of C-relations of $\boldsymbol{v}$*, see for instance [5, Definition 2 and Proposition 4].

Finally, a nonzero sequence $\boldsymbol{v}$ is said to be *C-finite* if together with a finite number of terms of $\boldsymbol{v}$ and a finite number of C-relations, one can recover all the terms of $\boldsymbol{v}$. This is equivalent to requiring that the ideal of C-relations of $\boldsymbol{v}$ is 0-dimensional, see also [4, Definition 2 and Proposition 3], where such sequences are called *linear recursive*.

Example 2.1. *On the one hand, the terms* $v_{i,j} = (5+4i+3j)2^{i+j} + (3+6i+j)5^{i+j}$ *of* $\boldsymbol{v} \in \mathbb{F}_7^{\mathbb{N}^2}$ *can all be computed thanks to* $v_{0,0} = v_{0,1} = v_{0,2} = 1$, $v_{1,0} = 0$ *and the C-relations, for all* $(i,j) \in \mathbb{N}^2$,

$$v_{i+1,j+1} + 3v_{i,j} = v_{i+2,j} + v_{i,j+2} + 6v_{i,j} = v_{i,j+3} + 4v_{i+1,j} + 6v_{i,j+1} = 0.$$

On the other hand, they can also be computed knowing $v_{0,0} = v_{0,1} = v_{0,2} = v_{0,3} = 1$ *and that for all* $(i,j) \in \mathbb{N}^2$,

$$v_{i,j+4} + 6v_{i,j+2} + 2v_{i,j} = v_{i+1,j} + 2v_{i,j+3} + 5v_{i,j+1} = 0.$$

Thus, the ideal of C-relations of $\boldsymbol{v}$ *is the* 0*-dimensional one* $\left\langle xy+3, x^2+y^2+6, y^3+4x+6y \right\rangle = \left\langle y^4+6y^2+2, x+2y^3+5y \right\rangle$ *and* $\boldsymbol{v}$ *is C-finite.*

On the other hand, the binomial sequence, $\mathbf{b} = \left(\mathsf{b}_{i,j}\right)_{(i,j)\in\mathbb{N}^2} = \left(\binom{i}{j}\right)_{(i,j)\in\mathbb{N}^2}$*, satisfies Pascal's rule: for all* $(i,j) \in \mathbb{N}^2$, $\mathsf{b}_{i+1,j+1} - \mathsf{b}_{i,j+1} - \mathsf{b}_{i,j} = 0$*. Moreover, one can show that this relation spans all the other C-relations, i.e. its ideal of C-relations is the* 1*-dimensional one* $\langle xy - y - 1 \rangle$*, thus* $\mathbf{b}$ *is not C-finite.*

Furthermore, some sequences satisfy *linear recurrence relations with coefficients that are polynomials* in the indices of the sequence, or *P-relations* for short. For instance, the binomial sequence satisfies the following two P-relations for all $(i,j) \in \mathbb{N}^2$:

$$(j+1)\,\mathsf{b}_{i,j+1} - (i-j)\,\mathsf{b}_{i,j} = 0$$
$$(i+1-j)\,\mathsf{b}_{i+1,j} - (i+1)\,\mathsf{b}_{i,j} = 0.$$

Combining them by shifting the former by index $(0,1)$ and then adding the latter yields

$$(i-j)\mathsf{b}_{i+1,j+1} - (i-j)\mathsf{b}_{i,j+1} - (i-j)\mathsf{b}_{i,j} = 0.$$

This proves that Pascal's rule holds whenever $i \neq j$.

We thus aim at representing the former relations as polynomials g_1 and g_2 such that for all $(i,j) \in \mathbb{N}^2$, $\left[g_1 x^i y^j\right]_{\boldsymbol{v}} = \left[g_2 x^i y^j\right]_{\boldsymbol{v}} = 0$. For instance, we could say that the first one corresponds to $\left[(j+1)x^i y^{j+1} - (i-j)x^i y^j\right]_{\boldsymbol{v}} = \left[((j+1)y - (i-j))\, x^i y^j\right]_{\boldsymbol{v}} = 0$, but this would mean that g_1 has coefficients in i and j, which are meaningless on their own. To circumvent this, in [8], the authors introduced new variables $\boldsymbol{t} = (t_1, \ldots, t_n)$, such that t_p behaves like $x_p\partial_p$, where ∂_p is the differential operator with respect to x_p. That is, $\left[\boldsymbol{t}^{\boldsymbol{k}}\boldsymbol{x}^{\boldsymbol{i}}\right]_{\boldsymbol{v}} \coloneqq \left[t_1^{k_1}\cdots t_n^{k_n}\boldsymbol{x}^{\boldsymbol{i}}\right]_{\boldsymbol{v}} = \left[(x_1\partial)^{k_1}\cdots(x_n\partial_n)^{k_n}\boldsymbol{x}^{\boldsymbol{i}}\right]_{\boldsymbol{v}} = \left[i_1^{k_1}\cdots i_n^{k_n}\boldsymbol{x}^{\boldsymbol{i}}\right]_{\boldsymbol{v}} = i_1^{k_1}\cdots i_n^{k_n}v_{\boldsymbol{i}} = \boldsymbol{i}^{\boldsymbol{k}}v_{\boldsymbol{i}}$. Then, the $[.]_{\boldsymbol{v}}$ notation is naturally $\mathbb{K}$-linearly extended to polynomials in $\boldsymbol{t}$ and $\boldsymbol{x}$. Therefore, the $2n$ variables $t_1, \ldots, t_n, x_1, \ldots, x_n$ follow, for all $1 \leq p, q \leq n$ and $p \neq q$, the commutation rules $x_p x_q = x_q x_p$, $t_p t_q = t_q t_p$, $t_p x_q = x_q t_p$ and $t_p x_p = x_p(t_p + 1)$, making polynomials in $\boldsymbol{t}$ and $\boldsymbol{x}$ *quasi-commutative*. The ring of skew-polynomials in $\boldsymbol{t}$ and $\boldsymbol{x}$, satisfying the quasi-commutative rules defined above, will be denoted $\mathbb{K}[\boldsymbol{t}]\langle\boldsymbol{x}\rangle$ while the ring of skew-polynomials in $\boldsymbol{x}$ with coefficients in $\mathbb{K}(\boldsymbol{t})$ will simply be denoted $\mathbb{K}(\boldsymbol{t})\langle\boldsymbol{x}\rangle$. Now, a P-relation is given by a finite subset $\mathcal{S}$ of $\mathbb{N}^n$ and polynomials $\gamma_{\boldsymbol{s}} \in \mathbb{K}[\boldsymbol{t}]$ for $\boldsymbol{s} \in \mathcal{S}$, such that

$$\forall \boldsymbol{i} \in \mathbb{N}^n, \sum_{\boldsymbol{s}\in\mathcal{S}} \gamma_{\boldsymbol{s}}(\boldsymbol{s}+\boldsymbol{i})v_{\boldsymbol{s}+\boldsymbol{i}} = 0.$$

This relation corresponds to the polynomial $g = \sum_{s\in\mathcal{S}} \gamma_s(\boldsymbol{t})\boldsymbol{x}^s \in \mathbb{K}[\boldsymbol{t}]\langle\boldsymbol{x}\rangle$ such that for all $\boldsymbol{i} \in \mathbb{N}^n$, $\left[g\boldsymbol{x}^{\boldsymbol{i}}\right]_{\boldsymbol{v}} = 0$.

Remark 2.2. *While we can obviously find polynomials $\tilde{\gamma}_s \in \mathbb{K}[\boldsymbol{t}]$ such that $\sum_{s\in\mathcal{S}} \tilde{\gamma}_s(\boldsymbol{i})v_{s+\boldsymbol{i}} = \sum_{s\in\mathcal{S}} \gamma_s(\boldsymbol{s}+\boldsymbol{i})v_{s+\boldsymbol{i}} = \left[g\boldsymbol{x}^{\boldsymbol{i}}\right]_{\boldsymbol{v}} = 0$, the notation with the $\gamma_s(\boldsymbol{s}+\boldsymbol{i})$'s makes more explicit the corresponding polynomial g in $\mathbb{K}[\boldsymbol{t}]\langle\boldsymbol{x}\rangle$.*

Example 2.3. *Let $t = t_1$, $u = t_2$, $x = x_1$ and $y = x_2$. Then, the P-relations satisfied by the binomial sequence can be rewritten as*

$$\begin{aligned}
(j+1)\,\mathsf{b}_{i,j+1} - (i-j)\,\mathsf{b}_{i,j} &= \left[(j+1)\,x^i y^{j+1} - (i-j)\,x^i y^j\right]_{\boldsymbol{v}} \\
0 &= \left[u x^i y^{j+1} - (t-u)\,x^i y^j\right]_{\boldsymbol{v}} \\
0 &= \left[(uy - (t-u))\,x^i y^j\right]_{\boldsymbol{v}} \\
\textit{and}\qquad (i+1-j)\,\mathsf{b}_{i+1,j} - (i+1)\,\mathsf{b}_{i,j} &= \left[(i+1-j)\,x^{i+1}y^j - (i+1)\,x^i y^j\right]_{\boldsymbol{v}} \\
0 &= \left[(t-u)\,x^{i+1}y^j - (t+1)\,x^i y^j\right]_{\boldsymbol{v}} \\
0 &= \left[((t-u)\,x - (t+1))\,x^i y^j\right]_{\boldsymbol{v}}.
\end{aligned}$$

Thus, $g_1 = uy - (t-u)$ and $g_2 = (t-u)\,x - (t+1)$ in $\mathbb{K}[t,u]\langle x, y\rangle$.

The set of all such polynomials is a right ideal of $\mathbb{K}[\boldsymbol{t}]\langle\boldsymbol{x}\rangle$. Indeed, it is stable by multiplication on the right by any monomial $\boldsymbol{x}^{\boldsymbol{i}}$ as requested. Furthermore, since $\boldsymbol{t}^{\boldsymbol{\ell}}\boldsymbol{x}^{\boldsymbol{j}}\boldsymbol{t}^{\boldsymbol{k}}\boldsymbol{x}^{\boldsymbol{i}} = \boldsymbol{t}^{\boldsymbol{\ell}}\,(\boldsymbol{t}-\boldsymbol{j})^{\boldsymbol{k}}\,\boldsymbol{x}^{\boldsymbol{j}+\boldsymbol{i}}$, $\left[\boldsymbol{t}^{\boldsymbol{\ell}}\boldsymbol{x}^{\boldsymbol{j}}\boldsymbol{t}^{\boldsymbol{k}}\boldsymbol{x}^{\boldsymbol{i}}\right]_{\boldsymbol{v}} = \left[\boldsymbol{t}^{\boldsymbol{\ell}}\,(\boldsymbol{t}-\boldsymbol{j})^{\boldsymbol{k}}\,\boldsymbol{x}^{\boldsymbol{j}+\boldsymbol{i}}\right]_{\boldsymbol{v}} = (\boldsymbol{j}+\boldsymbol{i})^{\boldsymbol{\ell}}\,\boldsymbol{i}^{\boldsymbol{k}}v_{\boldsymbol{j}+\boldsymbol{i}} = \boldsymbol{i}^{\boldsymbol{k}}\left[\boldsymbol{t}^{\boldsymbol{\ell}}\boldsymbol{x}^{\boldsymbol{j}+\boldsymbol{i}}\right]_{\boldsymbol{v}}$. In other words, multiplying on the right by $\boldsymbol{t}^{\boldsymbol{k}}\boldsymbol{x}^{\boldsymbol{i}}$ corresponds to multiplying on the right by $\boldsymbol{x}^{\boldsymbol{i}}$ and to multiply the evaluation by a constant, namely $\boldsymbol{i}^{\boldsymbol{k}}$. Thus if $\left[g\boldsymbol{x}^{\boldsymbol{i}}\right]_{\boldsymbol{v}}$ vanishes, then so does $\left[g\boldsymbol{t}^{\boldsymbol{k}}\boldsymbol{x}^{\boldsymbol{i}}\right]_{\boldsymbol{v}}$.

Such relations allow one to compute new terms of the sequence, though integer roots of the leading coefficient may prevent some computations. For instance, one cannot compute $\mathsf{b}_{i+1,i+1}$ from $\mathsf{b}_{i,i+1}$ using $(i+1-j)\mathsf{b}_{i+1,j} - (i+1)\mathsf{b}_{i,j}$ and $j = i+1$ as the coefficient in front of $\mathsf{b}_{i+1,j}$ vanishes. Thankfully, for this sequence, one can use the other relation $(j+1)\mathsf{b}_{i,j+1} - (i-j)\mathsf{b}_{i,j}$ with $i = j+1$ to achieve this goal.

Sequences satisfying P-relations form a large set. Among them, there are the *P-finite* ones. In particular, analogously to the C-finite case, a nonzero sequence $\boldsymbol{v}$ such that a finite number of its terms and a finite number of P-relations allows one to recover all of its terms is P-finite.

Example 2.4 (Cont. of Example 2.3)**.** *The ideal of P-relations of b in $\mathbb{K}[t,u]\langle x, y\rangle$ is*

$$\langle uy - (t-u),\, (t-u)x - (t+1),\, xy - y - 1\rangle\,.$$

Furthermore, since

$$\begin{aligned}
(xy - y - 1)\,(t-u) &= (t-u)\,xy - (t+1-u)\,y - (t-u) \\
&= ((t-u)\,x - (t+1))\,y + (uy - (t-u))\,,
\end{aligned}$$

in $\mathbb{K}(t,u)\langle x, y\rangle$, its ideal of P-relations is only spanned by $uy - (t-u)$ and $(t-u)x - (t+1)$.

Note that P-finite sequences are actually those whose generating series are D-finite and there exist P-finite sequences that do not satisfy the above prerequisites.

2.2. Gröbner bases

This section briefly recalls some basic definitions on Gröbner bases. The interested reader will find more details in [16] in the commutative case and [31, Chapter 2] in the quasi-commutative one.

For $\mathcal{T}$ the set of monomials in $\mathbb{K}[\boldsymbol{t}]\langle \boldsymbol{x}\rangle$, a monomial ordering $\prec$ on $\mathcal{T}$ is a total order relation satisfying the following three properties

1. $\forall m \in \mathcal{T}$, $1 \preceq m$;
2. $\forall m, m', s \in \mathcal{T}$, $m \preceq m' \Rightarrow ms \preceq m's$ and $sm \preceq sm'$.

For a monomial ordering $\prec$ on $\mathbb{K}[\boldsymbol{t}]\langle \boldsymbol{x}\rangle$, the *leading monomial* of f, denoted $\mathrm{LM}_\prec(f)$, is the greatest monomial in the support of f for $\prec$. For an ideal I, we let $\mathrm{LM}_\prec(I) = \{\mathrm{LM}_\prec(f),\ f \in I\}$. We recall briefly the definition of a Gröbner basis and of its associated staircase.

Definition 2.5. *Let I be a nonzero ideal of $\mathbb{K}[\boldsymbol{t}]\langle \boldsymbol{x}\rangle$ and let $\prec$ be a monomial ordering. A set $\mathcal{G} \subseteq I$ is a* Gröbner basis *of I if for all $f \in I$, there exists $g \in \mathcal{G}$ such that $\mathrm{LM}_\prec(g) | \mathrm{LM}_\prec(f)$, it is* reduced *if for any $g, g' \in \mathcal{G}$, and $g \neq g'$, any monomial $m \in \operatorname{supp} g'$ satisfies $\mathrm{LM}_\prec(g) \nmid m$.*

The staircase *of $\mathcal{G}$ is defined as $S = \mathrm{Staircase}(\mathcal{G}) = \{s \in \mathcal{T},\ \forall g \in \mathcal{G}, \mathrm{LM}_\prec(g) \nmid s\}$.*

More generally, a set S will be said to be a staircase *if for two monomials μ_1 and μ_2 such that $\mu_1\mu_2 \in S$, we have $\mu_1 \in S$ and $\mu_2 \in S$.*

Let us recall that $\mathrm{Staircase}(\mathcal{G})$ is also the canonical basis of $\mathbb{K}[\boldsymbol{t}]\langle \boldsymbol{x}\rangle / I$ as a $\mathbb{K}$-vector space.

Gröbner basis theory allows us to choose any monomial ordering, among which we mainly use, on the $\boldsymbol{x}$ variables, the

$\textsc{lex}(x_n \prec \cdots \prec x_1)$ **ordering** which satisfies $\boldsymbol{x}^{\boldsymbol{i}} \prec \boldsymbol{x}^{\boldsymbol{j}}$ if, and only if, there exists $1 \leq p \leq n$ such that for all $q < p$, $i_q = j_q$ and $i_p < j_p$, see [16, Chapter 2, Definition 3];

$\textsc{drl}(x_n \prec \cdots \prec x_1)$ **ordering** which satisfies $\boldsymbol{x}^{\boldsymbol{i}} \prec \boldsymbol{x}^{\boldsymbol{j}}$ if, and only if, $i_1 + \cdots + i_n < j_1 + \cdots + j_n$ or $i_1 + \cdots + i_n = j_1 + \cdots + j_n$ and there exists $2 \leq p \leq n$ such that for all $q > p$, $i_q = j_q$ and $i_p > j_p$, see [16, Chapter 2, Definition 6].

We will also use monomial orderings on the $\boldsymbol{t}$ and $\boldsymbol{x}$ variables. Since we want to freely switch from $\mathbb{K}[\boldsymbol{t}]\langle \boldsymbol{x}\rangle$ to $\mathbb{K}(\boldsymbol{t})\langle \boldsymbol{x}\rangle$ and vice versa, it makes sense to choose an ordering such that $t_k \prec x_\ell$ for any k and ℓ, such as $\textsc{lex}(t_n \prec \cdots \prec t_1 \prec x_n \prec \cdots x_1)$ or $\textsc{drl}(t_n \prec \cdots \prec t_1 \prec \cdots \prec x_1)$. The latter is more suitable as it allows us to enumerate all the monomials in $\boldsymbol{t}$ and $\boldsymbol{x}$ in increasing order.

2.3. Structured Gröbner bases

The *cones* we are dealing with are those that are *submonoids* of $\mathbb{N}^n$. These are subsets C of $\mathbb{N}^n$ such that $0 \in C$ and for all $\boldsymbol{i}, \boldsymbol{j} \in C$, $(\boldsymbol{i} + \boldsymbol{j}) \in C$.

Given such a cone C and polynomials with support in its associated set of monomials $\mathcal{T}(C) = \left\{\boldsymbol{x}^{\boldsymbol{i}} \in \mathcal{T} \middle| \boldsymbol{i} \in C\right\}$, one may want to perform all the polynomial operations with monomials in $\mathcal{T}(C)$ in order to take advantage of the structure of the support when computing a Gröbner basis of the ideal they span. While, this is not always possible, one can achieve this goal by considering the ideal the polynomials span in the subalgebra defined by C.

This leads to the definition of sparse Gröbner basis with support in $\mathcal{T}(C)$ that uses its monoid structure.

Definition 2.6 ([24, Definition 3.1] and [2, Definition 3.3])**.** *Let $C \subseteq \mathbb{N}^n$ be a cone and $\mathcal{T}(C)$ be its associated set of monomials. Then, $\mathbb{K}[C]$, the set of polynomials with support in $\mathcal{T}(C)$, is an algebra.*

Let $f_1, \dots, f_s \subseteq \mathbb{K}[C]$ be polynomials. We let $I = \langle f_1, \dots, f_s \rangle_C = \left\{ \sum_{k=1}^{s} f_k q_k \middle| q_1, \dots, q_s \in \mathbb{K}[C] \right\}$ be the ideal spanned by $f_1, \dots, f_s$ in $\mathbb{K}[C]$*. Then, a* sparse Gröbner basis *of I for a monomial ordering $\prec$ is a generating set $\mathcal{G} = \{g_1, \dots, g_r\} \subseteq \mathbb{K}[C]$ such that for all $f \in I$, $\operatorname{LM}_\prec(f) = \operatorname{LM}_\prec(g)m$ for some $g \in \mathcal{G}$ and $m \in \mathcal{T}(C)$.*

The associated staircase $\operatorname{Staircase}(\mathcal{G})$ *of $\mathcal{G}$ is the set of monomials s in $\mathcal{T}(C)$ such that for any $g \in \mathcal{G}$, there is no monomial $m \in \mathcal{T}(C)$ such that $s = \operatorname{LM}_\prec(g)m$.*

Let us notice that for $C = \mathbb{N}^n$, $\mathbb{K}[C] = \mathbb{K}[\mathbb{N}^n] = \mathbb{K}[\boldsymbol{x}]$ and sparse Gröbner bases are classical Gröbner bases. Furthermore, like classical Gröbner bases, sparse Gröbner bases allow one to solve the ideal membership problem in $\mathbb{K}[C]$ in an effective way.

For a *lattice* $\Lambda \subseteq \mathbb{Z}^n$, we let $\Lambda_{\geq 0} = \Lambda \cap \mathbb{N}^n$ be its nonnegative cone, so that, naturally, $\mathbb{Z}^n_{\geq 0} = \mathbb{N}^n$. In particular, Λ and $\Lambda_{\geq 0}$ are cones and we may intersect them with another cone. For $\boldsymbol{a} \in \mathbb{Z}^n$, we also denote by $\boldsymbol{a} + \Lambda$ the affine lattice obtained by translating Λ by $\boldsymbol{a}$ and likewise we can consider its intersection with a cone. In particular, $(\boldsymbol{a} + \Lambda)_{\geq 0} = (\boldsymbol{a} + \Lambda) \cap \mathbb{N}^n$.

Given a lattice Λ, its affine translates $\boldsymbol{a}_0 + \Lambda = \Lambda, \dots, \boldsymbol{a}_L + \Lambda$ and polynomials $f_1, \dots, f_k$, each with supports in an associated set of monomials $\mathcal{T}((\boldsymbol{a}_\ell + \Lambda)_{\geq 0})$, then a reduced Gröbner basis of $\langle f_1, \dots, f_k \rangle$ satisfies also this support property. This allows one to speed the Gröbner bases computations up by essentially performing L computations in parallel with input of sizes divided by L.

2.4. Multi-Hankel matrices

Given a table $\boldsymbol{v}$ and a polynomial $g \in \mathbb{K}[\boldsymbol{t}]\langle \boldsymbol{x} \rangle$, in order to determine if a polynomial g is in the ideal of P-relations of $\boldsymbol{v}$, one must check that $\left[g\boldsymbol{x}^{\boldsymbol{i}} \right]_{\boldsymbol{v}} = 0$ for all $\boldsymbol{i}$. As only a finite number of terms of $\boldsymbol{v}$ are known, only a finite number of such tests can be done.

Definition 2.7. *Let T be a finite subset of $\mathcal{T}\left(\mathbb{N}^{2n}\right)$, the set of monomials in $t_1, \dots, t_n, x_1, \dots, x_n$, and X be a finite subset of $\mathcal{T}(\mathbb{N}^n)$, the set of monomials in $x_1, \dots, x_n$.*

The multi-Hankel matrix *$H_{X,T}$ is the matrix whose rows are indexed by X and columns by T and whose coefficient at row $\boldsymbol{x}^{\boldsymbol{i}}$ and column $\boldsymbol{t}^{\boldsymbol{k}}\boldsymbol{x}^{\boldsymbol{j}}$ is $\left[\boldsymbol{t}^{\boldsymbol{k}}\boldsymbol{x}^{\boldsymbol{j}+\boldsymbol{i}} \right]_{\boldsymbol{v}}$.*

A vector in the right kernel of this matrix corresponds to a polynomial g with support in T such that $[gm]_{\boldsymbol{v}} = 0$ for all $m \in X$.

Example 2.8. *Let $\boldsymbol{v} = (v_{i,j})_{(i,j)\in\mathbb{N}^2}$ be a table and $T = \{1, u, t, y, x, uy, ty, ux, tx\} \subset \mathcal{T}\left(\mathbb{N}^{2n}\right)$ and $X = \left\{1, y, x, y^2, xy, x^2\right\} \subset \mathcal{T}(\mathbb{N}^n)$ be two sets of monomials, then their multi-Hankel matrix is*

$$H_{X,T} = \begin{array}{c} \\ 1 \\ y \\ x \\ y^2 \\ xy \\ x^2 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 1 & u & t & y & x & uy & ty & ux & tx \end{array} \\ \begin{pmatrix} v_{0,0} & 0 & 0 & v_{0,1} & v_{1,0} & v_{0,1} & 0 & 0 & v_{0,1} \\ v_{0,1} & v_{0,1} & 0 & v_{0,2} & v_{1,1} & 2v_{0,2} & 0 & v_{1,1} & v_{1,1} \\ v_{1,0} & 0 & v_{0,1} & v_{1,1} & v_{1,1} & v_{1,1} & v_{1,1} & 0 & 2v_{2,0} \\ v_{0,2} & 2v_{0,2} & 0 & v_{0,3} & v_{1,2} & 3v_{0,3} & 0 & 2v_{1,2} & v_{1,2} \\ v_{1,1} & v_{1,1} & v_{1,1} & v_{1,2} & v_{2,1} & 2v_{1,2} & v_{1,2} & v_{2,1} & 2v_{2,1} \\ v_{2,0} & 0 & 2v_{2,0} & v_{2,1} & v_{3,0} & v_{2,1} & 2v_{2,1} & 0 & 3v_{3,0} \end{pmatrix} \end{array}.$$

We give some computation details. The coefficient on the third column (t) and first row (1) is $[t \times 1]_{\boldsymbol{v}} = \left[tx^0y^0\right]_{\boldsymbol{v}} = 0^1 v_{0,0} = 0$. *Likewise, the coefficient on sixth column (uy) and the second to last row (xy) is* $[uyxy]_{\boldsymbol{v}} = \left[uxy^2\right]_{\boldsymbol{v}} = 2^1 v_{1,2} = 2v_{1,2}$.

Note that rows are only indexed with monomials in $\boldsymbol{x}$ and not in $\boldsymbol{t}, \boldsymbol{x}$ since the row labeled with $\boldsymbol{t}^{\boldsymbol{k}}\boldsymbol{x}^{\boldsymbol{i}}$, $\boldsymbol{k} \neq 0$ would be a multiple of the row labeled with $\boldsymbol{x}^{\boldsymbol{i}}$.

2.5. *The* Scalar-FGLM *algorithm*

The Scalar-FGLM algorithm [4, 5], takes as input the table $\boldsymbol{v}$ and a set of monomials T, which is a staircase, and computes the right kernel of the multi-Hankel matrix $H_{T,T}$. Vectors in this kernel can be seen as polynomials in $\mathbb{K}[\boldsymbol{x}]$ and these polynomials with a leading term minimal for the partial order induced by the division are the ones returned by the algorithm. Furthermore, if T is ordered for a monomial ordering $\prec$ and contains the staircase and the leading monomials of the reduced Gröbner basis of the ideal of C-relations of $\boldsymbol{v}$ for $\prec$, then the Scalar-FGLM algorithm returns this Gröbner basis.

As our goal is to extend the Scalar-FGLM algorithm in order to deal with table terms lying on a cone or a lattice, we recall this algorithm.

Algorithm 1: Scalar-FGLM

Input: A table $\boldsymbol{v} = (v_i)_{i \in \mathbb{N}^n}$ with coefficients in $\mathbb{K}$, a monomial ordering $\prec$, a sufficiently large staircase T and ordered for $\prec$.
Output: A reduced Gröbner basis of the ideal of C-relations of $\boldsymbol{v}$.
Build the matrix $H_{T,T}$.
Compute the set $S \subseteq T$ of smallest monomials, for $\prec$, such that $\operatorname{rank} H_{S,S} = \operatorname{rank} H_{T,T}$.
For all $m \in T \setminus S$ **do** // stabilize S for the division
  If $\exists s \in S$ *such that* $m \mid s$ **then** $S := S \cup \{m\}$.
$L := T \setminus S$ sorted for $\prec$.
$G := \varnothing$.
While $L \neq \varnothing$ **do**
  $g := \min_{\prec} L$
  Solve the linear system $H_{S,S}\boldsymbol{\gamma} + H_{S,\{g\}} = 0$.
  $G := G \cup \{g + \sum_{s \in S} \gamma_s s\}$.
  Remove g and any of its multiples from L.
Return G.

The algorithm computes the column rank profile of the matrix $H_{T,T}$, that is the set of left-most linearly independent columns of the matrix. Since these columns are independent from the previous ones, their labels cannot be the leading monomial, for $\prec$, of any polynomial in the ideal of C-relations, thus they are in the associated staircase of the reduced Gröbner basis of this ideal for $\prec$. If T is not large enough, a monomial m could be detected as not lying in the staircase while one of its multiples does, hence there is a stabilization process to add m to the staircase if this happens, see [4, Example 3]. Then, each output polynomial is computed by solving a linear system involving its leading monomial and the monomials in the staircase.

Example 2.9 (Cont. of Example 2.1)**.** *Let us recall that a Gröbner basis of the ideal of C-relation of $\boldsymbol{v}$ is* $\left\{xy + 3, x^2 + y^2 + 6, y^3 + 4x + 6y\right\}$ *for* drl$(y \prec x)$*, hence this ideal has degree* 4*. Therefore, the staircase of the Gröbner basis of this ideal for* lex$(y \prec x)$*, or any monomial ordering, can only contain monomials x^iy^j with $(i+1) \times (j+1) \leq 4$ and it suffices to take*

$T = \left\{1, y, y^2, y^3, y^4, x, xy, x^2, x^3, x^4\right\}$ *to recover the staircase and the Gröbner basis. The column rank profile of* $H_{T,T}$ *is given by* $S = \left\{1, y, y^2, y^3\right\}$ *so that* $L = \left\{y^4, x, xy, x^2, x^3, x^4\right\}$. *Then, the linear systems* $H_{S,S}\boldsymbol{\gamma} + H_{S,\{y^4\}} = 0$ *and* $H_{S,S}\boldsymbol{\gamma} + H_{S,\{x\}} = 0$ *yield the Gröbner basis*

$$\left\{y^4 + 6y^2 + 2, x + 2y^3 + 5y\right\}.$$

In many applications, for instance the Gröbner bases change of orderings one through the Sparse-FGLM algorithm, the computation of a single table element is costly. Therefore, we may want to reduce the number of table queries performed by the Scalar-FGLM algorithm. Algorithm 1 called on a set T requires $\#2T$ table terms, where $2T$ is the Minkowski sum of T with itself. To reduce this number of queries, the goal is to let the multi-Hankel grow step by step. We start with the 1×1 matrix

$$\begin{matrix} & 1 \\ 1 & \begin{pmatrix} [1]_v \end{pmatrix} \end{matrix}.$$

If $[1]_v = v_{\mathbf{0}} \neq 0$, then 1 is in the associated staircase of the Gröbner basis of the ideal of C-relations of $\boldsymbol{v}$, otherwise it stops and returns the set of relations $\{1\}$. The algorithm extends a full-rank matrix $H_{S,S}$ into $H_{S\cup\{m\},S\cup\{m\}}$ with m greater, for $\prec$, than any monomial in S. Now, there are two possibilities, either the new matrix has full rank or it is not and the column labeled with m is linearly dependent from the other ones. In the former case, m is actually in this staircase and S is replaced by $S \cup \{m\}$. In the latter case, a polynomial with support in $S \cup \{m\}$ and leading monomial, for $\prec$, m is found and no multiples of m will ever be proposed to extend the multi-Hankel matrix. The algorithm stops either when no monomials can be added to the staircase or when the size of the staircase has reached a threshold given in input. There is, however, a possibility of finding wrong relations if the first terms of the table exceptionally satisfies a relation of smaller order, for instance if $v_{\mathbf{0}} = 0$. This problem can be circumvented by testing relations further, that is adding a small buffer of constraints, i.e. rows of the matrix. This can be noticed for instance when the relations are suspiciously small or in FGLM applications where the degree of the ideal is known in advance.

3. Guessing with structures

In this section, we show how to guess linear recurrence relations of a table by taking the structure of the table terms into account. We first start with the case where only table terms in a cone are considered. Then, we study how to guess these relations when table terms are in a lattice or some affine translates thereof.

3.1. Terms in a cone

In this subsection, we aim at describing how we can take advantage of the structure of a given cone C to recover the ideal of relations of a table $\boldsymbol{v}$ by only considering table terms inside the cone. That is, we aim at guessing polynomials $g \in \mathbb{K}[C]$ such that for all $\boldsymbol{x^i} \in \mathcal{T}(C)$, $\left[g\boldsymbol{x^i}\right]_{\boldsymbol{v}} = 0$. This latter condition is the guessing part as we will only be able to ensure that $\left[g\boldsymbol{x^i}\right]_{\boldsymbol{v}} = 0$ for all $\boldsymbol{x^i}$ in a finite subset T of $\mathcal{T}(C)$.

To do so, two strategies are at our disposal and they both rely on the generators of C as a submonoid of $\mathbb{N}^n$. Let us denote by $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_\nu$ a set of generators of C, i.e. for all $\boldsymbol{i} \in C$, there exists $\boldsymbol{j} \in \mathbb{N}^\nu$ such that $\boldsymbol{i} = j_1\boldsymbol{a}_1 + \cdots + j_\nu\boldsymbol{a}_\nu$. Then, note that, first and foremost, there is no reason

for ν to be less than or equal to n. Second, even if ν is minimal and $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_\nu$ is a generating set, there is no reason for $(j_1, \ldots, j_\nu)$ to be unique.

Example 3.1. *The cone* $C = \left\{\boldsymbol{i} \in \mathbb{N}^2 \middle| i_1 \leq 2i_2, i_2 \leq 2i_1\right\}$ *is spanned by* $\boldsymbol{a}_1 = (1,1)$, $\boldsymbol{a}_2 = (1,2)$ *and* $\boldsymbol{a}_3 = (2,1)$ *so that* $C = \left\{j_1\boldsymbol{a}_1 + j_2\boldsymbol{a}_2 + j_3\boldsymbol{a}_3 \middle| \boldsymbol{a}_1 = (1,1), \boldsymbol{a}_2 = (1,2), \boldsymbol{a}_3 = (2,1), (j_1, j_2, j_3) \in \mathbb{N}^3\right\}$. *Yet, we have the two decompositions* $(3,3) = 3\boldsymbol{a}_1 = \boldsymbol{a}_2 + \boldsymbol{a}_3$.

The first strategy is designed to only consider table terms lying in C. Assuming a generating set $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_\nu$ of C is known, the set of monomials $\mathcal{T}(C)$ can be defined as

$$\mathcal{T}(C) = \left\{\boldsymbol{x}^{j_1\boldsymbol{a}_1} \cdots \boldsymbol{x}^{j_\nu\boldsymbol{a}_\nu} \middle| (j_1, \ldots, j_\nu) \in \mathbb{N}^\nu\right\}.$$

The second strategy makes use of a new set of variables $\boldsymbol{y} = (y_1, \ldots, y_\nu)$, so that y_1 represents $\boldsymbol{x}^{\boldsymbol{a}_1}$, etc and an auxiliary table $\boldsymbol{w} = (w_{\boldsymbol{j}})_{\boldsymbol{j} \in \mathbb{N}^\nu}$ defined by $w_{\boldsymbol{j}} = v_{j_1\boldsymbol{a}_1 + \cdots + j_\nu\boldsymbol{a}_\nu}$. Then, two monomials $\boldsymbol{y}^{\boldsymbol{j}}$ and $\boldsymbol{y}^{\boldsymbol{k}}$ represent the same monomial $\boldsymbol{x}^{\boldsymbol{i}}$ if, and only if, $\boldsymbol{i} = j_1\boldsymbol{a}_1 + \cdots + j_\nu\boldsymbol{a}_\nu = k_1\boldsymbol{a}_1 + \cdots + k_\nu\boldsymbol{a}_\nu$. This implies that both $w_{\boldsymbol{j}}$ and $w_{\boldsymbol{k}}$ are equal to $v_{\boldsymbol{i}}$. Thus, $\boldsymbol{w}$ satisfies extra relations coming from these multiple equivalent writings. They are given by binomials, namely $\boldsymbol{y}^{\boldsymbol{j}} - \boldsymbol{y}^{\boldsymbol{k}}$. Hence, not all monomials in $\mathcal{T}(\mathbb{N}^\nu)$ are of interest and we clean them up by using the binomial ideal $I(C)$ they span, for instance by reducing $\boldsymbol{y}^{\boldsymbol{j}}$ to $\boldsymbol{y}^{\boldsymbol{k}}$.

In practice, both strategies are equivalent. They only differ in how they enumerate table terms $v_{\boldsymbol{i}}$ with $\boldsymbol{i} \in C$. Note, though, that the second strategy requires computing a Gröbner basis of $I(C)$, for instance using [30] while the first one only requires checking that a monomial has already been generated. However, such a Gröbner basis computation should not be the bottleneck compared to the computations of the table terms or the linear algebra routines for the guessing step.

Since the first strategy comes down to directly calling the Scalar-FGLM algorithm with a set of monomials $T \subset \mathcal{T}(C)$, this yields Theorem 3.2.

Theorem 3.2. *Let C be a submonoid cone of $\mathbb{N}^n$ spanned by the minimal set of generators $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_\nu\}$. Let $\prec$ be a monomial ordering on $\mathcal{T}$, the set of monomials in n variables, and let $T \subset \mathcal{T}(C)$ be a staircase ordered for $\prec$.*

Then, the Scalar-FGLM *algorithm called on table $\boldsymbol{v}$, T and $\prec$ returns a set of polynomials G with support in $\mathcal{T}(C)$, such that for all $s \in T \setminus \langle \mathrm{LM}_\prec(G)\rangle$, s is in the associated staircase of a sparse Gröbner basis of the ideal of C-relations of $\boldsymbol{v}$ for $\prec$.*

Furthermore, if the ideal of C-relations of $\boldsymbol{v}$ is 0-dimensional and has a reduced sparse Gröbner basis with support in T for $\prec$, then the output of the Scalar-FGLM *algorithm called on $\boldsymbol{v}$ and T is this reduced sparse Gröbner basis.*

Proof. As the Scalar-FGLM algorithm computes kernel vectors of $H_{T,T}$, the corresponding polynomials can only have support in $\mathcal{T}(C)$.

Let S be the associated staircase of a sparse Gröbner basis of the ideal of C-relations of $\boldsymbol{v}$.

Let us show first that no monomial $m \notin S$ is found in the staircase by the algorithm. As $m \in \mathrm{LM}_\prec(I)$, there exist $\alpha_s \in \mathbb{K}$, for all $s \in S$ such that $m + \sum_{s\in S} \alpha_s s \in I$, thus $\left[t\left(m + \sum_{s\in S} \alpha_s s\right)\right]_{\boldsymbol{v}} = 0$ for all $t \in \mathcal{T}$. Since $T \subset \mathcal{T}$, this means that the column labeled with m is linearly dependent from the previous ones and neither m nor any multiple thereof is in the staircase associated to the output. Hence, the computed staircase is included in the correct staircase.

Let us now assume that the ideal of C-relations of $\boldsymbol{v}$ is 0-dimensional, that is S is finite. We shall show by contradiction that the matrix $H_{S,S}$ has full rank, so that the output of the Scalar-FGLM algorithm called on $T \supset S$ is a reduced Gröbner basis whose associated staircase contains

S. Let us assume that $H_{S,S}$ has not full rank and let $m \notin S$ be the smallest monomial for $\prec$ such that $\operatorname{rank} H_{S,S\cup\{m\}} > \operatorname{rank} H_{S,S}$, such a monomial exists for otherwise a monomial in S would be the leading monomial of a relation. Let R be any finite subset of $S \cup \{\mu | \mu \preceq m\}$, which is also a staircase containing $S \cup \{m\}$. By minimality of m, for $\prec$, $\operatorname{rank} H_{S,R} = \operatorname{rank} H_{S,S\cup\{m\}} > \operatorname{rank} H_{S,S}$ and in particular the column labeled with m must be independent from the previous ones. Thus, no polynomial with leading monomial m can be in the ideal of relations and m is in the staircase of this ideal. This is a contradiction with the assumption that m is not in S. Since $S \subseteq T$, the algorithm correctly computes a superset of the staircase S and thus the algorithm discovers the correct staircase.

Finally, the polynomials of the sparse Gröbner basis are found by linear algebra. □

Concerning the second strategy, since $I(C)$ is spanned by binomials, the reduced Gröbner basis $\mathcal{G}$ of $I(C)$ for $\prec$ is made of binomials, see for instance [16, Chapter 5, Section 3, Exercise 13]. Note that while the result is only asked to be proved for the lexicographic ordering, the given hint can be used to show that the statement holds for any monomial ordering. Thus, any monomial in $\mathcal{T}(\mathbb{N}^\nu)$ reduces to a single monomial modulo $\mathcal{G}$ and we denote by $\mathcal{T}(\mathbb{N}^\nu)/I(C)$ the set of monomials that cannot be reduced by $\mathcal{G}$. Furthermore, if $\boldsymbol{y}^{\boldsymbol{j}} \in \mathcal{T}(\mathbb{N}^\nu)$, then any monomial $\boldsymbol{y}^{\boldsymbol{k}} \in \mathcal{T}(\mathbb{N}^\nu)$ that divides $\boldsymbol{y}^{\boldsymbol{j}}$ is in $\mathcal{T}(\mathbb{N}^\nu)/I(C)$. Indeed, if $\boldsymbol{y}^{\boldsymbol{k}}$ were not, then it would be a leading monomial in $I(C)$ and so would $\boldsymbol{y}^{\boldsymbol{j}}$. Hence, one can always pick a finite staircase $T \subset \mathcal{T}(\mathbb{N}^\nu)/I(C)$ and call the Scalar-FGLM algorithm with T and $\prec$. Then, by construction, it remains to replace the output polynomials in $\mathbb{K}[\boldsymbol{y}]$ by the corresponding ones in $\mathbb{K}[\boldsymbol{x}]$. They will naturally have support in $\mathcal{T}(C)$.

Example 3.3 (Continuation of Example 3.1)**.** *It is clear that* $3\boldsymbol{a}_1 = \boldsymbol{a}_2 + \boldsymbol{a}_3$ *generates all the other different ways to decompose an element of* C*, hence* $I(C) = \left\langle y_1^3 - y_2 y_3 \right\rangle$*. Thus, when listing the monomials for* $\textsc{drl}(y_1 \prec y_2 \prec y_3)$ *in* $\mathcal{T}(\mathbb{N}^\nu)/I(C)$*, we will skip any multiple of* y_1^3.

3.2. Terms in a lattice

Let $\Lambda_{\geq 0}$ be the set of nonnegative terms of a sublattice of $\mathbb{Z}^n$, we aim at guessing the recurrence relations of a table $\boldsymbol{v}$ by following $\Lambda_{\geq 0}$. Since a lattice is a special case of a cone, by Theorem 3.2, restricting ourselves to only considering the subtable $(v_{\boldsymbol{i}})_{\boldsymbol{i}\in\Lambda_{\geq 0}}$ shall make us guess the reduced sparse Gröbner basis of the ideal of relations of $\boldsymbol{v}$ in $\mathbb{K}[\Lambda_{\geq 0}]$.

Yet, doing so would in some way make us forget the extra structure coming with a sublattice: namely its fundamental domain, i.e. the quotient group $\mathbb{Z}^n/\Lambda$. Indeed, if a set of polynomials $\{f_1, \ldots, f_r\}$ is such that for all k, there exists $\boldsymbol{a}_k \in \mathbb{Z}^n/\Lambda$ such that $\operatorname{supp} f_k \subset (\boldsymbol{a}_k + \Lambda)_{\geq 0}$, then a classical reduced Gröbner basis $\mathcal{G} = \{g_1, \ldots, g_s\}$ of the ideal it spans in $\mathbb{K}[\boldsymbol{x}]$ satisfies the same property. Therefore, if we expect, or even can ensure beforehand, that the reduced Gröbner basis of the ideal of relations of $\boldsymbol{v}$ also satisfies this property, we aim at guessing this Gröbner basis by working *in parallel* on several smaller multi-Hankel matrices whose sizes have been divided by $\#(\mathbb{Z}^n/\Lambda)$.

To do so, considering an input set of monomials $T \subset \mathcal{T}$, we shall split it up into $T = \bigsqcup_{\boldsymbol{a}\in\mathbb{Z}^n/\Lambda} T_{\boldsymbol{a}}$, with $T_{\boldsymbol{a}} = T \cap \mathcal{T}((\boldsymbol{a} + \Lambda)_{\geq 0})$, and then call the Scalar-FGLM algorithm on $\boldsymbol{v}$ and $T_{\boldsymbol{a}}$ for each $\boldsymbol{a}$. However, the table terms that appear in $H_{T_{\boldsymbol{a}},T_{\boldsymbol{a}}}$ are $v_{\boldsymbol{i}}$ with $\boldsymbol{i} \in (2\boldsymbol{a} + \Lambda)_{\geq 0}$. Thus, we might never consider certain table terms. To circumvent this, we always add the row and the column labeled with 1 in these matrices. This yields the Lattice Scalar-FGLM algorithm or Algorithm 2 and Theorem 3.4.

Algorithm 2: Lattice Scalar-FGLM

Input: A table $\boldsymbol{v} = (v_i)_{i\in\mathbb{N}^n}$ with coefficients in $\mathbb{K}$, a monomial ordering $\prec$, a staircase T ordered for $\prec$, a nonnegative lattice $\Lambda \subseteq \mathbb{Z}^n$, a set $\mathcal{A} \subseteq \mathbb{N}^n$ containing 0 such that $\Lambda + \mathcal{A} = \mathbb{Z}^n$.

Output: A truncated reduced Gröbner basis.

Partition T into $T = \bigsqcup_{\boldsymbol{a}\in\mathcal{A}} T_{\boldsymbol{a}}$ with $T_{\boldsymbol{a}} = (T \cap \mathcal{T}((\boldsymbol{a}+\Lambda)_{\geq 0}))$.

For all $\boldsymbol{a} \in \mathcal{A}$ **do**

  Build the matrix $H_{\{1\}\cup T_{\boldsymbol{a}},\{1\}\cup T_{\boldsymbol{a}}}$.

  Compute its column profile rank $S_{\boldsymbol{a}}$.

$S \coloneqq \bigcup_{\boldsymbol{a}\in\mathcal{A}} S_{\boldsymbol{a}}$.

For all $m \in T \setminus S$ **do** // make S a staircase

  If $\exists s \in S$ *such that* $m \mid s \in S$ **then** $S \coloneqq S \cup \{m\}$.

$L \coloneqq T \setminus S$ sorted for $\prec$.

$G \coloneqq \varnothing$.

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

  $g \coloneqq \min_{\prec} L$

  Find $\boldsymbol{a} \in \mathcal{A}$ such that $g \in \mathcal{T}((\boldsymbol{a}+\Lambda)_{\geq 0})$.

  Solve the linear system $H_{S_{\boldsymbol{a}},S_{\boldsymbol{a}}}\boldsymbol{\gamma} + H_{S_{\boldsymbol{a}},\{g\}} = 0$.

  $G \coloneqq G \cup \left\{g + \sum_{s\in S_{\boldsymbol{a}}} \gamma_s s\right\}$.

  Remove g and any of its multiples from L.

Return G.

Theorem 3.4. *Let Λ be a sublattice of $\mathbb{Z}^n$ with fundamental domain $\mathcal{A}$. Let $\prec$ be a monomial ordering on $\mathcal{T}$ and let $T \subset \mathcal{T}$ be a finite staircase ordered for $\prec$.*

Then, the Lattice Scalar-FGLM *algorithm called on table $\boldsymbol{v}$, T and $\prec$ returns a truncated Gröbner basis of an ideal whose polynomials are each with support in $\{1\} \cup \mathcal{T}((\boldsymbol{a}+\Lambda)_{\geq 0})$, with $\boldsymbol{a} \in \mathcal{A}$.*

Furthermore, let $\mathcal{G}$ be a reduced Gröbner basis for $\prec$ satisfying this support property. Let $\mathcal{S}$ be the associated staircase and $\boldsymbol{v}$ be a generic C-finite table whose ideal of relations is spanned by $\mathcal{G}$. Let T be a staircase containing $\mathcal{S}$ and the leading monomials of all the polynomials in $\mathcal{G}$. Then, there exists a non empty Zariski open set of values for the table terms $[s]_{\boldsymbol{v}}$ of $\boldsymbol{v}$, with $s \in \mathcal{S}$, such that the Lattice Scalar-FGLM *algorithm called on $\boldsymbol{v}$, $\prec$, T and $\mathcal{A}$ correctly guesses $\mathcal{G}$.*

Proof. This proof follows mostly the same steps as that of Theorem 3.2.

As the algorithm computes kernel vectors of matrices $H_{\{1\}\cup T_{\boldsymbol{a}},\{1\}\cup T_{\boldsymbol{a}}}$, the corresponding polynomials can only have support in $\{1\} \cup \mathcal{T}((\boldsymbol{a}+\Lambda)_{\geq 0})$.

Let S be the associated staircase of a reduced Gröbner basis of the ideal of C-relations of $\boldsymbol{v}$. For each $\boldsymbol{a} \in \mathcal{A}$, we let $S_{\boldsymbol{a}} = S \cap \mathcal{T}((\boldsymbol{a}+\Lambda)_{\geq 0})$.

Let us show first that no monomial $m \notin S$ is found in the staircase by the algorithm. As $m \in \mathrm{LM}_{\prec}(I)$, there exist $g = \mathrm{LM}_{\prec}(g) + \sum_{\alpha_s\in S_{\boldsymbol{a}}} \alpha_s s \in I$ such that $\mathrm{LM}_{\prec}(g) \in \mathcal{T}((\boldsymbol{a}+\Lambda)_{\geq 0})$ and $\mathrm{LM}_{\prec}(g) | m$. Thus, $\frac{m}{\mathrm{LM}_{\prec}(g)} g \in I$ and for all $t \in \mathcal{T}$, $\left[t\frac{m}{\mathrm{LM}_{\prec}(g)} g\right]_{\boldsymbol{v}} = 0$. In particular, this is true for all $t \in T_{\mathbf{b}}$, with $m \in T_{\mathbf{b}}$, so that the column labeled with m is linearly dependent from the previous ones in $H_{\{1\}\cup T_{\mathbf{b}},\{1\}\cup T_{\mathbf{b}}}$. Hence, neither m nor any of its multiples is in the staircase associated to the output. That is, the computed staircase is included in the correct staircase.

It remains to prove the last statement. Let $\boldsymbol{v}$ be a sequence whose ideal of relations I is spanned by $\mathcal{G}$. First, from the proof of the Scalar-FGLM algorithm, we know that the matrix

$H_{\mathcal{S},\mathcal{S}}$ has full rank. For any $\boldsymbol{a} \in \mathcal{A}$, we let $\mathcal{S}_a = \{1\} \cup (\mathcal{S} \cap \mathcal{T}((\boldsymbol{a} + \Lambda)_{\geq 0}))$. If $H_{\mathcal{S}_a,\mathcal{S}_a}$ has full rank, then for $m \in \text{LM}_{\prec}(\mathcal{G}) \cap \mathcal{T}((\boldsymbol{a} + \Lambda)_{\geq 0})$, this matrix allows us to determine the polynomial in $\mathcal{G}$ with leading monomial m. Thus, the algorithm correctly returns $\mathcal{G}$.

For each $\boldsymbol{a} \in \mathcal{A}$, we know that the matrix $H_{\mathcal{S}_a,\mathcal{S}}$ has full rank. Now, to recover a relation with support in $\mathcal{T}(\boldsymbol{a} + \Lambda)_{\geq 0}$, generically, it suffices to consider sufficiently many shifts of this relations. And in particular, we can take the shifts induced by monomials in $\mathcal{S}_{\boldsymbol{a}}$, meaning that the matrix $H_{\mathcal{S}_a,\mathcal{S}_a}$ generically has full rank. □

Remark 3.5. *Adding a row labeled with* 1 *in the matrices is necessary to prevent computations of incorrect relations when one of them is divisible by a non trivial monomial. Let us consider a unidimensional table* $\boldsymbol{v}$ *satisfying the relation* $x^4 - ax^2$ *with* $a \in \mathbb{K}$ *and let* $\Lambda = 2\mathbb{Z}$ *and* $T = \left\{1, x, x^2, x^3, x^4\right\}$*, so that* $T_0 = \left\{1, x^2, x^4\right\}$ *and* $T_1 = \left\{x, x^3\right\}$*. We thus build the matrices*

$$H_{T_0,T_0} = \begin{array}{c} \\ 1 \\ x^2 \\ x^4 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & x^2 & x^4 \end{array} \\ \begin{pmatrix} [1]_{\boldsymbol{v}} & \left[x^2\right]_{\boldsymbol{v}} & \left[x^4\right]_{\boldsymbol{v}} \\ \left[x^2\right]_{\boldsymbol{v}} & \left[x^4\right]_{\boldsymbol{v}} & \left[x^6\right]_{\boldsymbol{v}} \\ \left[x^4\right]_{\boldsymbol{v}} & \left[x^6\right]_{\boldsymbol{v}} & \left[x^8\right]_{\boldsymbol{v}} \end{pmatrix} \end{array} = \begin{array}{c} \\ 1 \\ x^2 \\ x^4 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & x^2 & x^4 \end{array} \\ \begin{pmatrix} v_0 & v_2 & av_2 \\ v_2 & av_2 & a^2v_2 \\ av_2 & a^2v_2 & a^3v_2 \end{pmatrix} \end{array},$$

$$H_{T_1,T_1} = \begin{array}{c} \\ x \\ x^3 \end{array} \begin{array}{c} \begin{array}{cc} x & x^3 \end{array} \\ \begin{pmatrix} \left[x^2\right]_{\boldsymbol{v}} & \left[x^4\right]_{\boldsymbol{v}} \\ \left[x^4\right]_{\boldsymbol{v}} & \left[x^6\right]_{\boldsymbol{v}} \end{pmatrix} \end{array} = \begin{array}{c} \\ x \\ x^3 \end{array} \begin{array}{c} \begin{array}{cc} x & x^3 \end{array} \\ \begin{pmatrix} v_2 & av_2 \\ av_2 & a^2v_2 \end{pmatrix} \end{array}.$$

By hypothesis, clearly the column labeled with x^4 *is linearly dependent from the ones with label* 1 *and* x^2*. However, since* $\left[x^4 - ax^2\right]_{\boldsymbol{v}} = \left[x^6 - ax^4\right]_{\boldsymbol{v}} = 0$*, the column labeled with* x^3 *is linearly dependent from the column labeled with* x *in the second matrix. Therefore, these matrices do not allow us to recover that* x^3 *is in the staircase of the ideal of relations of the input table.*

Yet, the matrix

$$H_{\{1\}\cup T_1,\{1\}\cup T_1} = \begin{array}{c} \\ 1 \\ x \\ x^3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & x & x^3 \end{array} \\ \begin{pmatrix} [1]_{\boldsymbol{v}} & [x]_{\boldsymbol{v}} & \left[x^3\right]_{\boldsymbol{v}} \\ [x]_{\boldsymbol{v}} & \left[x^2\right]_{\boldsymbol{v}} & \left[x^4\right]_{\boldsymbol{v}} \\ \left[x^3\right]_{\boldsymbol{v}} & \left[x^4\right]_{\boldsymbol{v}} & \left[x^6\right]_{\boldsymbol{v}} \end{pmatrix} \end{array} = \begin{array}{c} \\ 1 \\ x \\ x^3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & x & x^3 \end{array} \\ \begin{pmatrix} v_0 & v_1 & v_3 \\ v_1 & v_2 & 0 \\ v_3 & 0 & 0 \end{pmatrix} \end{array}$$

has its column labeled with x^3 *independent from the previous two if, and only if,* $v_3 \neq 0$*, allowing us to detect that* x^3 *is in the staircase.*

Example 3.6. *Consider the table* $\boldsymbol{v} = \left(2^i\,(j + 1 \bmod 3)\right)_{(i,j)\in\mathbb{N}^2}$ *defined over* $\mathbb{Q}$*. Using, for instance, the Berlekamp–Massey–Sakata or the* Scalar-FGLM *algorithms, we can easily show that its ideal of relations is* $\left\langle y^3 - 1, x - 2\right\rangle$*. Let us consider the lattice* $\Lambda = (0,3)\mathbb{Z} + (1,0)\mathbb{Z}$*, so that* $\mathcal{A} = \{(0,0),(0,1),(0,2)\}$ *and* $T = \left\{1, y, y^2, y^3, y^4, y^5, x\right\}$*.*

Then, Algorithm 2 builds the matrices

$$H_{T_0,T_0} = \begin{array}{c} \\ 1 \\ y^3 \\ x \end{array} \begin{array}{c} \begin{array}{ccc} 1 & y^3 & x \end{array} \\ \begin{pmatrix} 1 & 1 & 2 \\ 1 & 1 & 2 \\ 2 & 2 & 4 \end{pmatrix} \end{array}, \; H_{T_1,T_1} = \begin{array}{c} \\ 1 \\ y \\ y^4 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & y & y^4 \end{array} \\ \begin{pmatrix} 1 & 2 & 2 \\ 2 & 0 & 0 \\ 2 & 0 & 0 \end{pmatrix} \end{array}, \; H_{T_2,T_2} = \begin{array}{c} \\ 1 \\ y^2 \\ y^5 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & y^2 & y^5 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 2 \\ 0 & 2 & 2 \end{pmatrix} \end{array}.$$

So that $S_0 = \{1\}$, $S_1 = \{1, y\}$ *and* $S_2 = \left\{1, y^2\right\}$. *Hence* $S = \left\{1, y, y^2\right\}$, $L = \left\{y^3, y^4, y^5, x\right\}$. *This yields the linear systems* $H_{S_0,S_0}\boldsymbol{\gamma} + H_{S_0,\{y^3\}} = 0$ *and* $H_{S_0,S_0}\boldsymbol{\gamma} + H_{S_0,\{x\}} = 0$ *allowing us to recover* $y^3 - 1$ *and* $x - 2$.

Notice that $\boldsymbol{w} = \left(2^i \,(j \bmod 3)\right)_{(i,j)\in\mathbb{N}^2}$ *has the same ideal of relations. Yet, the algorithm will build the matrices*

$$H_{T_0,T_0} = \begin{array}{c} \\ 1 \\ y^3 \\ x \end{array}\!\!\begin{array}{c} \begin{array}{ccc} 1 & y^3 & x \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array}, \quad H_{T_1,T_1} = \begin{array}{c} \\ 1 \\ y \\ y^4 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} 1 & y & y^4 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 2 \end{pmatrix} \end{array}, \quad H_{T_2,T_2} = \begin{array}{c} \\ 1 \\ y^2 \\ y^5 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} 1 & y^2 & y^5 \end{array} \\ \begin{pmatrix} 0 & 2 & 2 \\ 2 & 1 & 1 \\ 2 & 1 & 1 \end{pmatrix} \end{array},$$

so that $S_0 = \varnothing$, $S_1 = \{1, y\}$, $S_2 = \left\{1, y^2\right\}$ *and* $S = \left\{1, y, y^2\right\}$. *Since the linear systems* $H_{S_0,S_0}\boldsymbol{\gamma} + H_{S_0,\{x^3\}} = 0$ *and* $H_{S_0,S_0}\boldsymbol{\gamma} + H_{S_0,\{y\}} = 0$ *are empty, they do not allow us to recover* $x^3 - 1$ *and* $y - 2$. *Indeed,* $\varnothing = S_0 \neq S \cap \mathcal{T}(\Lambda_{\geq 0}) = \{0\}$.

Yet, for the table $\boldsymbol{w}' = \left(\left[(1 + \lambda y)x^i y^j\right]_{\boldsymbol{v}}\right)$, *the algorithm builds the matrices*

$$H_{T_0,T_0} = \begin{array}{c} \\ 1 \\ y^3 \\ x \end{array}\!\!\begin{array}{c} \begin{array}{ccc} 1 & y^3 & x \end{array} \\ \begin{pmatrix} \lambda & \lambda & 2\lambda \\ \lambda & \lambda & 2\lambda \\ 2\lambda & 2\lambda & 4\lambda \end{pmatrix} \end{array}, \quad H_{T_1,T_1} = \begin{array}{c} \\ 1 \\ y \\ y^4 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} 1 & y & y^4 \end{array} \\ \begin{pmatrix} \lambda & 1 + 2\lambda & 1 + 2\lambda \\ 1 + 2\lambda & 2 & 2 \\ 1 + 2\lambda & 2 & 2 \end{pmatrix} \end{array},$$

$$H_{T_2,T_2} = \begin{array}{c} \\ 1 \\ y^2 \\ y^5 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} 1 & y^2 & y^5 \end{array} \\ \begin{pmatrix} \lambda & 2 & 2 \\ 2 & 1 + 2\lambda & 1 + 2\lambda \\ 2 & 1 + 2\lambda & 1 + 2\lambda \end{pmatrix} \end{array}.$$

It is clear that $S_0 = \{1\}$, *provided* $\lambda \neq 0$, $S_1 = \{1, y\}$, *provided* $4\lambda^2 + 2\lambda + 1 \neq 0$, *and* $S_2 = \left\{1, y^2\right\}$, *provided* $2\lambda^2 + \lambda - 4 \neq 0$. *All in all, the algorithm succeeds for* $\boldsymbol{w}'$ *as long as* λ *does not satisfy* $\lambda(4\lambda^2 + 2\lambda + 1)(2\lambda^2 + \lambda - 4) = 0$.

Remark 3.7. *While we assume that* Λ *is a sublattice of* $\mathbb{Z}^n$, *hence of rank* n, *it can actually be any* $\mathbb{Z}$*-submodule of smaller rank* ν. *However, this means we can only guess an ideal of relations in* ν *variables so that it may not be the whole ideal of relations. Nevertheless, this kind of restriction can be of interest in the P-finite application where the kernel equation makes us study the P-finite nature of a subsequence where some indices are set.*

3.3. Application to Gröbner basis change of orderings with the action of a matrix group

In [23], the authors propose a variant of the FGLM algorithm [21], the so-called Sparse-FGLM algorithm, relying on guessing C-relations. More precisely, from the input Gröbner basis $\mathcal{G}$, they build a random table $\boldsymbol{v}$ whose ideal of relations is $\langle\mathcal{G}\rangle$. To do so, first, for each monomial s in the staircase associated to $\mathcal{G}$, they pick at random the table term $[s]_{\boldsymbol{v}}$, then they compute the other table terms using the C-relations induced by $\mathcal{G}$. Finally, applying an algorithm for guessing C-relations on this table and the second input ordering, they obtain the Gröbner basis of the ideal of relations of this table for this second ordering. If the first Gröbner basis spans a Gorenstein ideal [13, 17], then with high probability, the output Gröbner basis is a Gröbner basis of the same ideal and thus the target one.

In particular, assuming generic properties, detailed below, on the polynomials that span the ideal we want to compute a Gröbner basis of, this algorithm comes down to computing products of a sparse matrix and some vectors and solving Hankel systems.

The goal of this section is to extend this approach to abelian group actions on the ideal. In particular, we will restrict ourselves to finite abelian matrix group actions, that is finite abelian subgroups of $\mathrm{GL}(n)$ where $A \in \mathrm{GL}(n)$ acts on $f(\boldsymbol{x}) \in \mathbb{K}[\boldsymbol{x}]$ by sending it to $f(A\boldsymbol{x})$.

3.3.1. Finite matrix group actions

We start by recalling some results on finite matrix group actions on ideals of $\mathbb{K}[\boldsymbol{x}]$.

Let G be a finite abelian matrix group. By the invariant factors theorem, there exist $q_1 \mid \cdots \mid q_\ell$ such that $G \simeq \mathbb{Z}/q_1\mathbb{Z} \times \cdots \times \mathbb{Z}/q_\ell\mathbb{Z}$ and in particular, for any $g \in G$, $g^{q_\ell} = 1$ and q_ℓ is minimal for this property.

Furthermore if $|G| = q_1 \cdots q_\ell$ is not divisible by the characteristic of the coefficient field $\mathbb{K}$, then there exists a primitive q_ℓth root of unity ζ such that the matrices in G are simultaneously diagonalizable with powers of ζ on the diagonals, see [25, Theorem 2]. After this diagonalization process, which comes down to a linear change of variables, for each matrix in G, there exist natural numbers $0 \leq \varepsilon_1, \ldots, \varepsilon_n \leq q_\ell - 1$ such that x_i is sent onto $\zeta^{\varepsilon_i} x_i$ by this matrix.

Definition 3.8 ([25, Definition 3])**.** *Let $G \simeq \mathbb{Z}/q_1\mathbb{Z} \times \cdots \times \mathbb{Z}/q_\ell\mathbb{Z}$, with $q_1 \mid \cdots \mid q_\ell$, be a diagonal subgroup of* $\mathrm{GL}(n)$ *and ζ be a q_ℓth root of unity, then there exist matrices $D_1, \ldots, D_n$ spanning G such that each D_i has order q_i.*

*For each monomial $m \in \mathcal{T}$, there exist $(\mu_1, \ldots, \mu_\ell) \in \mathbb{Z}/q_1\mathbb{Z} \times \cdots \times \mathbb{Z}/q_\ell\mathbb{Z}$ such that for all i, m is sent onto $\zeta^{\mu_i q_\ell / q_i} m$ by D_i. Then, m is said to have G-*degree *$(\mu_1, \ldots, \mu_\ell)$.*

*Furthermore, a polynomial is G-*homogeneous *if all its monomials have same G-degree.*

From this, one can prove that the G-degree of the product of two monomials is the sum of their G-degrees. Since the G-degree of the monomial 1 is $(0, \ldots, 0)$, the subset of monomials of G-degree $(0, \ldots, 0)$ is a sublattice $\mathcal{T}(\Lambda_{\geq 0})$ of $\mathcal{T}$. A consequence of this is that if $f_1, \ldots, f_s$ are G-homogeneous polynomials, then a reduced Gröbner basis of $\langle f_1, \ldots, f_s \rangle$ is made of G-homogeneous polynomials as well and $\langle f_1, \ldots, f_s \rangle$ is stable by the action of G, see [25, Theorem 4].

3.3.2. Gröbner bases change of orderings

From the reduced DRL Gröbner basis $\mathcal{G}$ of such an ideal, it then makes sense to apply the SPARSE-FGLM algorithm in order to obtain the reduced LEX Gröbner basis. Since we already know that the support of each polynomial in the target Gröbner basis lies on a lattice, or an affine translate thereof, we can use Algorithm 2 to guess the relations on the table that is built by the algorithm. Furthermore, since the table is built with its first table terms picked at random, no fake relations, like in Example 3.6, should be guessed.

Proposition 3.9. *Let G be an abelian group as in Definition 3.8.*

Let $f_1, \ldots, f_s$ be generic polynomials of degree $d_1, \ldots, d_s$ such that $I = \langle f_1, \ldots, f_s \rangle$ is zero-dimensional stable by the action of G.

Let $\prec$ and $<$ be a monomial orders and let $\mathcal{G}$ and $\mathcal{H}$ be the reduced Gröbner basis of I for $\prec$ and $<$respectively.

Then, the guessing step of $\mathcal{H}$ when calling the SPARSE-FGLM *algorithm on $\mathcal{G}$, $\prec$ and $<$ can be sped up using the* LATTICE SCALAR-FGLM *algorithm by a factor $O(|G|^{\omega-1})$ instead of the* SCALAR-FGLM *algorithm.*

Proof. From [25] and the genericity assumption on I, the polynomials in $\mathcal{H}$ are evenly split between all the G-degrees. Furthermore, so are the monomials in the Staircase($\mathcal{H}$).

Now, to recover $\mathcal{H}$, one needs to call the Scalar-FGLM and Lattice Scalar-FGLM algorithms on a staircase T that contains S and $\mathrm{LM}_{\prec}(\mathcal{H})$. Then, the Scalar-FGLM algorithm computes the right-kernel of $H_{T,T}$ in at most $O(\#T^\omega)$ operations. Now, the Lattice Scalar-FGLM will build $|G|$ submatrices of $H_{T,T}$ of size roughly $\#T/|G|$ and compute the right-kernel of each. Thus it can be done in $O(\#T^\omega/|G|^{\omega-1})$. □

We shall say that a zero-dimensional ideal $I \subset \mathbb{K}[\boldsymbol{x}]$ has

Property S, if its reduced Gröbner basis for $\mathrm{LEX}(x_n \prec \cdots \prec x_1)$ is in *shape position*. That is, there exist $g_1, \ldots, g_n \in \mathbb{K}[x_n]$ of degree at most $D-1$ such that this reduced Gröbner basis is $\left\{x_n^D + g_n(x_n), x_{n-1} + g_{n-1}(x_n), \ldots, x_1 + g_1(x_n)\right\}$.

Property M, if its reduced Gröbner basis for $\mathrm{DRL}(x_n \prec \cdots \prec x_1)$ satisfies the following condition. For every monomial m in the staircase associated to this Gröbner basis, either mx_n is in the staircase or it is the leading monomial of some polynomial in this Gröbner basis.

Let us recall that if I is spanned by generic polynomials $f_1, \ldots, f_n \in \mathbb{K}[\boldsymbol{x}]$ of degree $d_1, \ldots, d_n$ and $\mathbb{K}$ is sufficiently large or infinite, then both Properties S and M are satisfied. See for instance [23, Proposition 5.3], where x_1 is chosen as the smallest variable, for the latter. For the former, this is a direct consequence of I being radical with solutions not sharing the same last coordinate. Thus, the Shape lemma applies without requiring any change of variables, see [26, Lemma 1.4].

Under these assumptions, several algorithms can be used to compute the reduced LEX Gröbner basis of an ideal of degree D from the reduced DRL one. The seminal one, FGLM [21] with a complexity $O(nD^3)$, the Sparse-FGLM one [22, 23] with a complexity $\tilde{O}(kD^2 + nD)$, where k is the number of polynomials in the reduced DRL Gröbner basis whose leading monomial is divisible by x_n, a faster variant [20] of the FGLM algorithm using Keller-Gehrig algorithm [29] or SyzygyModuleBasis [35, Algorithm 3] both with a complexity $\tilde{O}(nD^\omega)$.

Whenever an ideal is stabilized by the action of such a finite abelian matrix group, the goal is to take advantage of this to speed the change of orderings algorithm up. In [25, Theorem 10], the authors show the complexity of the FGLM algorithm drops to $O(D^3/|G|^2)$, mainly because they deal with $|G|$ matrices of sizes roughly $D/|G|$ instead of one larger matrix of size D. These matrices correspond to those of monomials of each G-degree. It would be interesting to study if, using the same trick, one could make the complexities of the faster variant of the FGLM algorithm or of the SyzygyModuleBasis algorithm drop to $\tilde{O}(nD^\omega/|G|^2)$ or even $\tilde{O}(nD^\omega/|G|^{\omega-1})$.

Let us notice that in this situation, the Sparse-FGLM algorithm only relies on 1-dimensional algorithms like the Berlekamp–Massey one as the best strategy. We now focus on the complexity improvements one can reach in this setting when the ideal spanned by $\mathcal{G}$ and $\mathcal{H}$ is stable under the action of a G as in Definition 3.8.

We now focus on the Sparse-FGLM algorithm and we assume that a reduced G-homogeneous Gröbner basis for $\mathrm{DRL}(x_n \prec \cdots \prec x_1)$, spanning an ideal satisfying Property M, is given and the goal is to recover the reduced Gröbner basis for $\mathrm{LEX}(x_n \prec \cdots \prec x_1)$ satisfying Property S. By G-homogeneity, the support of each polynomial in the target Gröbner basis, namely $\{x_n^D + g_n(x_n), x_{n-1} + g_{n-1}(x_n), \ldots, x_1 + g_1(x_n)\}$, is already known. It is given by the G-degree of its leading monomial, namely $x_n^D, x_{n-1}, \ldots, x_1$. Since G is finite, there exists $d > 0$ minimal such that x_n^d has G-degree $(0, \ldots, 0)$ and there exists $\delta_n, \ldots, \delta_1 \geq 0$, all minimal, such that $x_n^{\delta_n}$ has same

G-degree as x_n^D and $x_n^{\delta_i}$ has same G-degree as x_i for $1 \leq i \leq n-1$. Therefore, for $1 \leq i \leq n$, $\operatorname{supp} g_i = \left\{ x_n^{\delta_i}, x_n^{\delta_i+d}, \ldots, x_n^{\delta_i+\left\lfloor \frac{D-1-\delta_i}{d} \right\rfloor d} \right\}$.

Thus, the polynomial g_n can be computed by solving the following Hankel system

$$\begin{array}{c} \\ x_n^{\delta_n} \\ x_n^{\delta_n+d} \\ \vdots \\ x_n^{\delta_n+\left\lfloor \frac{D-1-\delta_n}{d} \right\rfloor d} = x_n^{D-d} \end{array} \begin{array}{c} \begin{array}{cccc} x_n^{\delta_n} & x_n^{\delta_n+d} & \cdots & x_n^{D-d} \end{array} \\ \begin{pmatrix} \left[x_n^{2\delta_n}\right]_{\boldsymbol{v}} & \left[x_n^{2\delta_n+d}\right]_{\boldsymbol{v}} & \cdots & \left[x_n^{D-d+\delta_n}\right]_{\boldsymbol{v}} \\ \left[x_n^{2\delta_n+d}\right]_{\boldsymbol{v}} & \left[x_n^{2\delta_n+2d}\right]_{\boldsymbol{v}} & \cdots & \left[x_n^{D-d+\delta_n}\right]_{\boldsymbol{v}} \\ \vdots & \vdots & & \vdots \\ \left[x_n^{D-d}\right]_{\boldsymbol{v}} & \left[x_n^{D-d+\delta_n}\right]_{\boldsymbol{v}} & \cdots & \left[x_n^{2D-2d}\right]_{\boldsymbol{v}} \end{pmatrix} \end{array} \boldsymbol{\gamma} + \begin{array}{c} \\ x_n^{\delta_n} \\ x_n^{\delta_n+d} \\ \vdots \\ x_n^{D-d} \end{array} \begin{array}{c} \begin{array}{c} x_n^D \end{array} \\ \begin{pmatrix} \left[x_n^{D+\delta_n}\right]_{\boldsymbol{v}} \\ \left[x_n^{D+\delta_n+d}\right]_{\boldsymbol{v}} \\ \vdots \\ \left[x_n^{2D-d}\right]_{\boldsymbol{v}} \end{pmatrix} \end{array} = 0.$$

Denoting M_n the matrix of the multiplication by x_n in $\mathbb{K}[\boldsymbol{x}]/I$, $\mathbf{1} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ and $\mathbf{r}$ a vector picked at random, the table terms $\left[x_n^i\right]_{\boldsymbol{v}}$ are defined as $\mathbf{r}^{\mathrm{T}} M_n^i \mathbf{1}$. This is done by computing $v_0 = \mathbf{r}^{\mathrm{T}}, v_1 = v_0 M_n, v_2 = v_1 M_n, \ldots$ and then extracting the first coordinate of each vector to simulate the multiplication by $\mathbf{1}$.

Since we do not need all the terms but only $v_{2\delta_n}, v_{2\delta_n+d}, v_{2\delta_n+2d}, \ldots$, we first compute $v_{2\delta_n}$ and M_n^d in order to perform big steps. Let us notice that, following [22, 23], by Property M, the columns of matrix M_n are of two types. If a monomial m in the staircase is such that mx_n is still in the staircase, then the column corresponding to m is *trivial*, it is a vector of the canonical basis. Otherwise, m is the leading monomial of a polynomial g in the reduced Gröbner basis and the column corresponding to m is the coefficient vector of its normal form, namely $m - g$. Usually, these latter vectors are denser than the former. Then, M_n^d has the same shape as M_n, it has trivial and non-trivial columns. Furthermore, if M_n has k non-trivial columns, then M_n^d has at most $\max(D, kd)$ non-trivial columns. From [25] and the genericity assumption on I, we know we can split M_n in $|G|^2$ matrices of size at most $\lceil D/|G| \rceil$. Furthermore, its non-trivial columns are evenly split in the small matrices, i.e. the number of non-trivial columns of each small matrix is at most $\lceil k/|G| \rceil$. Then, we can multiply all these small matrices accordingly to obtain the splitting of M_n^d.

Now, polynomials $g_1, \ldots, g_{n-1}$ can be computed by solving a similar Hankel system:

$$\begin{array}{c} \\ x_n^{\delta_n} \\ x_n^{\delta_n+d} \\ \vdots \\ x_n^{D-d} \end{array} \begin{array}{c} \begin{array}{cccc} x_n^{\delta_i} & x_n^{\delta_i+d} & \cdots & x_n^{\delta_i+\left\lfloor \frac{D-1-\delta_i}{d} \right\rfloor d} \end{array} \\ \begin{pmatrix} \left[x_n^{\delta_i+\delta_n}\right]_{\boldsymbol{v}} & \left[x_n^{\delta_i+\delta_n+d}\right]_{\boldsymbol{v}} & \cdots & \left[x_n^{\delta_i+\delta_n+\left\lfloor \frac{D-1-\delta_i}{d} \right\rfloor d}\right]_{\boldsymbol{v}} \\ \left[x_n^{\delta_i+\delta_n+d}\right]_{\boldsymbol{v}} & \left[x_n^{\delta_i+\delta_n+2d}\right]_{\boldsymbol{v}} & \cdots & \left[x_n^{\delta_i+\delta_n+\left\lfloor \frac{D-1-\delta_i}{d}+1 \right\rfloor d}\right]_{\boldsymbol{v}} \\ \vdots & \vdots & & \vdots \\ \left[x_n^{\delta_i+D-d}\right]_{\boldsymbol{v}} & \left[x_n^{\delta_i+\delta_n+D-d}\right]_{\boldsymbol{v}} & \cdots & \left[x_n^{\delta_i+\delta_n+\left\lfloor \frac{D-1-\delta_i}{d}-1 \right\rfloor d+D}\right]_{\boldsymbol{v}} \end{pmatrix} \end{array} \boldsymbol{\gamma} + \begin{array}{c} \\ x_n^{\delta_n} \\ x_n^{\delta_n+d} \\ \vdots \\ x_n^{D-d} \end{array} \begin{array}{c} \begin{array}{c} x_i \end{array} \\ \begin{pmatrix} \left[x_i x_n^{\delta_n}\right]_{\boldsymbol{v}} \\ \left[x_i x_n^{\delta_n+d}\right]_{\boldsymbol{v}} \\ \vdots \\ \left[x_i x_n^{D-d}\right]_{\boldsymbol{v}} \end{pmatrix} \end{array} = 0.$$

However, the matrices might all be different. In order to speed the computation up, we change the linear systems into ones with the same matrix as the first one. This is done by multiplying all

the column labels by $x_n^{\delta_n-\delta_i}$. The constant vectors of the systems thus become

$$\begin{array}{c} \\ x_n^{\delta_n} \\ x_n^{\delta_n+d} \\ \vdots \\ x_n^{D-d} \end{array} \begin{array}{c} \begin{array}{c} x_n^{\delta_n-\delta_i}x_i \end{array} \\ \begin{pmatrix} \left[x_i x_n^{2\delta_n-\delta_i}\right]_{\boldsymbol{v}} \\ \left[x_i x_n^{2\delta_n+d-\delta_i}\right]_{\boldsymbol{v}} \\ \vdots \\ \left[x_i x_n^{D-d+\delta_n-\delta_i}\right]_{\boldsymbol{v}} \end{pmatrix} \end{array}.$$

Proposition 3.10. *Let $I \subset \mathbb{K}[\boldsymbol{x}]$ be a zero-dimensional ideal of degree D, invariant under the action of a finite diagonal matrix group G. Let us assume that I satisfies both properties S and M and that the matrix M_n has k non-trivial columns. Let furthermore S be the staircase associated to the* $\textsc{lex}(x_n \prec \cdots \prec x_1)$ *Gröbner basis of I, $\mathcal{T}(\Lambda_{\geq 0})$ be the set of monomials of G-degree* 0 *and for $A, B \subseteq \mathcal{T}$, $A + B = \{ab | a \in A, b \in B\}$ be the Minkowski sum of A and B.*

Then, we can recover the $\textsc{lex}(x_n \prec \cdots \prec x_1)$ *Gröbner basis, $\mathcal{G}$, of I from its* $\textsc{drl}(x_n \prec \cdots \prec x_1)$ *Gröbner basis using $\#\left((S \cap \mathcal{T}(\Lambda_{\geq 0})) + ((S \cap \mathcal{T}(\Lambda_{\geq 0})) \cup \textsc{lm}_{\prec}(\mathcal{G}))\right)$ table terms and $O\left(\frac{nkD^2}{|G|}\right)$ operations.*

Proof. Since the ideal I satisfies Property S, the staircase S associated to its $\textsc{lex}(x_n \prec \cdots \prec x_1)$ Gröbner basis is $\left\{1, x_n, \ldots, x_n^{D-1}\right\}$. Therefore, by definition of d, $S \cap \mathcal{T}(\Lambda_{\geq 0}) = \left\{1, x_n^d, \ldots, x_n^{\left\lfloor \frac{D-1}{d} \right\rfloor d}\right\}$. Thus, the matrix rows labels are in bijection with a subset of $S \cap \mathcal{T}(\Lambda_{\geq 0})$ while the matrix column labels and the column-vector label are in bijection with a subset of $(S \cap \mathcal{T}(\Lambda_{\geq 0})) \cup \textsc{lm}_{\prec}(\mathcal{G})$. This show that only $\#\left((S \cap \mathcal{T}(\Lambda_{\geq 0})) + ((S \cap \mathcal{T}(\Lambda_{\geq 0})) \cup \textsc{lm}_{\prec}(\mathcal{G}))\right)$ table terms are required.

Since I also satisfies Property M, M_n has k non-trivial columns and $D - k$ columns that are vectors of the canonical basis. Furthermore, these non-trivial columns correspond to G-homogeneous polynomials, so each of them has at most $O(D/|G|)$ nonzero coefficients. Thus, M_n has $O(kD/|G|)$ nonzero coefficients. Now, computing $v_{2\delta_n}$ requires $2\delta_n$ multiplications between M_n and a vector. Hence $v_{2\delta_n}$ can be computed in $O\left(\delta_n kD/|G|\right)$ operations.

It remains to compute $v_{2\delta_n+jd} = \mathbf{r}^{\mathrm{T}} M_n^{2\delta+jd}$ for all j up to $(2D - d - 2\delta_n)/d$ by successive multiplications by M_n^d. While M_n^d has $\max(D, kd)$ non-trivial columns, these non-trivial columns still represent G-homogeneous polynomials, thus M_n^d has $O(kdD/|G|)$ nonzero coefficients. Hence, all these vectors can be computed in $O(kD^2/|G|)$ operations.

For the constant vectors of the Hankel systems, we need to extract the coefficients corresponding to x_i of vectors $v_{2\delta_n-\delta_i+jd}$ for each i and each j. First, let us notice that each vector $v_{2\delta_n-\delta_i}$ has been computed in order to obtain $v_{2\delta_n}$. Then, the others are computed by successive multiplications by M_n^d, as for the vectors $v_{2\delta_n+jd}$. Thus, they can be obtained in $O(nkD^2/|G|)$ operations.

Finally, these linear systems are Hankel of size $O(D/d)$ sharing the same matrix and thus can be solved in $O\left(\mathsf{M}\left(\frac{D}{d}\right)\left(n + \log \frac{D}{d}\right)\right)$ operations, see [14]. This step is therefore not the bottleneck of the algorithm. □

4. Adaptive approach

4.1. The Adaptive Scalar-FGLM *algorithm*

A drawback of the Scalar-FGLM algorithm is that, in order to return the correct Gröbner basis, it needs to be called with a staircase T that contains both the support of the Gröbner basis

and its associated staircase. Without the help of an oracle, which we have in a multi-modular setting for instance, it is not an easy task to find such a T. Thus, an adaptive variant was designed by the authors in [4, 5] in order to discover the associated staircase and the Gröbner basis step by step. As a byproduct, it also minimizes the number of table queries.

Given a table $\boldsymbol{v}$ and a monomial ordering $\prec$, the Adaptive Scalar-FGLM starts with the empty set $S = \emptyset$. At each step, S is a staircase and a subset of the correct one. Then, for a monomial $\boldsymbol{x}^{\boldsymbol{i}}$ such that $S \cup \left\{\boldsymbol{x}^{\boldsymbol{i}}\right\}$ is also a staircase, if $H_{S\cup\{\boldsymbol{x}^{\boldsymbol{i}}\},S\cup\{\boldsymbol{x}^{\boldsymbol{i}}\}}$ has a greater rank than $H_{S,S}$, then S is replaced by $S \cup \left\{\boldsymbol{x}^{\boldsymbol{i}}\right\}$. Otherwise we have found a relation with leading monomial $\boldsymbol{x}^{\boldsymbol{i}}$ and we shall never try any multiple of $\boldsymbol{x}^{\boldsymbol{i}}$ as a new term in S.

In the cone setting, as in Section 3.1, the two strategies can be used. If we build an auxiliary table $\boldsymbol{w} \in \mathbb{K}^{\mathbb{N}^r}$, then the Adaptive Scalar-FGLM algorithm can directly be called on $\boldsymbol{w}$ provided we only try to add monomials $\boldsymbol{y}^{\boldsymbol{j}}$ that are in $\mathcal{T}(\mathbb{N}^r)/I(C)$. If we rather call it on the original table $\boldsymbol{v} \in \mathbb{K}^{\mathbb{N}^n}$, then we modify the algorithm so that only monomials in $\mathcal{T}(C)$ are used. Furthermore, once a relation with leading monomial $\boldsymbol{x}^{\boldsymbol{i}}$ is found, we shall never try any multiple $\boldsymbol{x}^{\boldsymbol{i}+\boldsymbol{j}}$ in the cone, i.e. with $\boldsymbol{x}^{\boldsymbol{j}} \in \mathcal{T}(C)$.

Example 4.1. *Consider the linear King walk $\boldsymbol{v} = (v_{i_0,i_1})_{(i_0,i_1)\in\mathbb{N}^2}$ counting the number of ways to reach i_1 in i_0 steps of size 1 starting from 0 in the nonnegative ray. It is clear that $v_{i_0,i_1} = 0$ whenever either $i_1 > i_0$ or $i_0 + i_1 = 1 \bmod 2$, so that we shall only consider the cone*

$$\begin{aligned} C &= \left\{(i_0, i_1) \in \mathbb{N}^2 \middle| i_0 + i_1 = 0 \bmod 2, i_1 \le i_0\right\} \\ &= (1,1)\mathbb{N} + (0,2)\mathbb{N}. \end{aligned}$$

Assume we consider the $\mathrm{LEX}(x_1 \prec x_0)$ *ordering, so that* $\mathcal{T}(C) = \left\{1, x_0x_1, x_0^2, x_0^2x_1^2, x_0^4, \ldots\right\}$.

1. *We build the matrix* $\begin{array}{c} \\ 1 \end{array}\begin{pmatrix} 1 \end{pmatrix}$ *which has full rank.*
2. *We increase the matrix by adding monomials in $\mathcal{T}(C)$ so we build* $\begin{array}{c} \\ 1 \\ x_0x_1 \end{array}\begin{pmatrix} 1 & x_0x_1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}$ *which does not have full rank, so we have found the fake relation $x_0x_1 - 1$.*
3. *We increase the matrix to* $\begin{array}{c} \\ 1 \\ x_0^2 \end{array}\begin{pmatrix} 1 & x_0^2 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}$ *which has full rank.*
4. *We increase the matrix to* $\begin{array}{c} \\ 1 \\ x_0^2 \\ x_0^4 \end{array}\begin{pmatrix} 1 & x_0^2 & x_0^4 \\ 1 & 1 & 2 \\ 1 & 2 & 5 \\ 2 & 5 & 14 \end{pmatrix}$ *which has full rank.*
5. *And so on.*

In the lattice setting however, we need to be more careful. We shall make one matrix per element in $\mathbb{Z}^n/\Lambda$ and each time we must add an extra column and an extra row, they will be added to the matrix corresponding to the monomial labeling the extra column. If there is no rank increase, then as usual a relation is found and no multiple of this monomial will ever label any new column in *any* matrix. This yields Algorithm 3 and Theorem 4.2.

Algorithm 3: Lattice Adaptive Scalar-FGLM

Input: A table $\boldsymbol{v} = (v_i)_{i\in\mathbb{N}^n}$ with coefficients in $\mathbb{K}$, a monomial ordering $\prec$, a nonnegative lattice $\Lambda \subseteq \mathbb{N}^n$, a set $\mathcal{A} \subseteq \mathbb{N}^n$ containing 0 such that $\Lambda + \mathcal{A} = \mathbb{Z}^n$.
Output: A set G of relations.
If $v_{(0,\dots,0)} = 0$ **then Return** [1].
$L \coloneqq \{x_1, \dots, x_n\}$.
Sort L by increasing order wrt. $\prec$.
$G \coloneqq \emptyset$ // the future set of relations
For all $\boldsymbol{a} \in \mathcal{A}$ **do** $S_{\boldsymbol{a}} \coloneqq \{1\}$. // the future staircase
While $L \neq \emptyset$ **do**
  $m \coloneqq$ first element of L and remove it from L.
  Pick $\boldsymbol{a} \in \mathcal{A}$ such that $m \in \mathcal{T}((\boldsymbol{a} + \Lambda)_{\geq 0})$.
  $S' \coloneqq S_{\boldsymbol{a}} \cup \{m\}$.
  If $H_{S',S'}$ *has full rank* **then** // No relation
    $S_{\boldsymbol{a}} \coloneqq S'$.
    $L \coloneqq L \cup \{x_1 m, \dots, x_n m\}$ Sort L by increasing order wrt. $\prec$ and remove duplicates and multiples of $\mathrm{LM}_{\prec}(G)$.
  Else // Relation!
    Solve $H_{S_{\boldsymbol{a}},S_{\boldsymbol{a}}}\boldsymbol{\gamma} + H_{S_{\boldsymbol{a}},\{m\}} = 0$.
    $G \coloneqq G \cup \left\{m + \sum_{s\in S_{\boldsymbol{a}}} \gamma_s s\right\}$ and remove multiples of m in L.
return G.

Theorem 4.2. *Let Λ be a sublattice of $\mathbb{Z}^n$ with fundamental domain $\mathcal{A}$. Let $\prec$ be a monomial ordering on $\mathcal{T}$. Let us assume that the* Lattice Adaptive Scalar-FGLM *algorithm called on table $\boldsymbol{v}$, $\prec$, Λ and $\mathcal{A} \subseteq \mathbb{N}^n$ returns a non-empty set of polynomials G.*

Let us denote by S the associated staircase to G and $S_{\boldsymbol{a}} = S \cap \mathcal{T}((\boldsymbol{a} + \Lambda)_{\geq 0})$ for each $\boldsymbol{a} \in \mathcal{A}$.

Then, for any polynomial $g \in G$ with $\mathrm{LM}_{\prec}(g) \in \mathcal{T}((\boldsymbol{a} + \Lambda)_{\geq 0})$ and any $s \in S_{\boldsymbol{a}}$ with $s \prec \mathrm{LM}_{\prec}(g)$, we have $[gs]_{\boldsymbol{v}} = 0$.

Furthermore, let $\mathcal{G}$ be a Gröbner basis for $\prec$ spanning a 0-dimensional ideal such that for all $g \in \mathcal{G}$, there exists $\boldsymbol{a} \in \mathcal{A}$ such that $\operatorname{supp} g \subset \mathcal{T}((\boldsymbol{a} + \Lambda)_{\geq 0})$. Let $\mathcal{S}$ be the associated staircase and $\boldsymbol{v}$ be a generic C-finite table whose ideal of relations is spanned by $\mathcal{G}$. Then, there exists a non empty Zariski open set of values for the table terms $[s]_{\boldsymbol{v}}$ of $\boldsymbol{v}$, with $s \in \mathcal{S}$, such that the Lattice Adaptive Scalar-FGLM *algorithm called on $\boldsymbol{v}$, $\prec$ and $\mathcal{A}$ correctly guesses $\mathcal{G}$.*

Proof. In the while loop, either monomial m is added to the staircase $S_{\boldsymbol{a}}$ or it is the leading monomial of a polynomial g that is added to G.

In the latter case, only monomials less than m can have been added to $S_{\boldsymbol{a}}$. Thus, the current set $S_{\boldsymbol{a}}$ is actually the final set $S_{\boldsymbol{a}}$ with only elements less than m, i.e. $S_{\boldsymbol{a}} \cap \{t \prec m\}$. Now, $H_{S_{\boldsymbol{a}}\cap\{t\prec m\},S_{\boldsymbol{a}}\cap\{t\prec m\}}\gamma + H_{S_{\boldsymbol{a}}\cap\{t\prec m\},\{m\}} = 0$ is equivalent to $[gs]_{\boldsymbol{v}} = 0$ for any s a row index, that is $s \in S_{\boldsymbol{a}}$ with $s \prec m$.

Let us prove the second assertion. For any $\boldsymbol{a} \in \mathcal{A}$, let $\mathcal{S}_{\boldsymbol{a}} = \mathcal{S} \cap \mathcal{T}((\boldsymbol{a} + \Lambda)_{\geq 0})$. A necessary and sufficient condition for the Lattice Adaptive Scalar-FGLM algorithm to correctly guess $\mathcal{G}$ is that for each $\boldsymbol{a}$, $\mathcal{S}_{\boldsymbol{a}} \subseteq S_{\boldsymbol{a}}$, which means that $S_{\boldsymbol{a}} = \{1\} \cup \mathcal{S}_{\boldsymbol{a}}$. This can only happen if, for each $\boldsymbol{a}$ and each monomial $m \in \mathcal{S}_{\boldsymbol{a}}$, the rank condition in the if statement is fulfilled. Following the proof of Theorem 3.4, we can build a sequence $\boldsymbol{w}$ from $\boldsymbol{v}$ whose ideal of relations is also spanned by $\mathcal{G}$ but whose such that the rank conditions in the if statement is satisfied for all monomial $m \in \mathcal{S}$. □

Remark 4.3. *If an incorrect staircase is guessed, then not much can be said on the output set of polynomials compared to the correct Gröbner basis. However, we know that the guessed staircase is included in the correct one.*

Example 4.4. *Let us consider the same first table as in Example 3.6,* $\boldsymbol{v} = \left(2^i\,(j+1 \bmod 3)\right)_{(i,j)\in\mathbb{N}^2}$ *and its associated lattice* $\Lambda = (0,3)\mathbb{Z}+(1,0)\mathbb{Z}$*, so that* $\mathcal{A} = \{(0,0),(0,1),(0,2)\}$*. We also consider the* $\text{LEX}(y \prec x)$ *ordering.*

1. *We build three matrices* $\begin{matrix}1\end{matrix}\begin{pmatrix}1\\1\end{pmatrix}$ *which have full rank.*
2. *We increase the second matrix to* $\begin{matrix}1\\y\end{matrix}\begin{pmatrix}1 & y\\1 & 2\\2 & 0\end{pmatrix}$ *which has full rank.*
3. *We increase the third matrix to* $\begin{matrix}1\\y^2\end{matrix}\begin{pmatrix}1 & y^2\\1 & 0\\0 & 2\end{pmatrix}$ *which has full rank.*
4. *We increase the first matrix to* $\begin{matrix}1\\y^3\end{matrix}\begin{pmatrix}1 & y^3\\1 & 1\\1 & 1\end{pmatrix}$ *which does not have full rank so that we have found that* y^3-1 *is in the ideal of relations.*
5. *We increase the first matrix to* $\begin{matrix}1\\x\end{matrix}\begin{pmatrix}1 & x\\1 & 2\\2 & 4\end{pmatrix}$ *which does not have full rank so that we have found that* $x-2$ *is in the ideal of relations.*
6. *We return* $\left\{y^3-1, x-2\right\}$.

4.2. Mixed approach for guessing P-relations

In [8], the authors proposed a mixed approach for guessing P-relations based on a Gröbner basis computations for reducing the number of table queries. The idea is that if two polynomials $g_1, g_2 \in \mathbb{K}[\boldsymbol{t}]\langle\boldsymbol{x}\rangle$ are P-relations satisfied by the table, then any polynomial in $\langle g_1, g_2\rangle$ is also a P-relation. Therefore, as soon as two P-relations g_1 and g_2 are guessed, the goal is to compute a Gröbner basis $\{g_1, g_2, \ldots, g_r\}$ of $\langle g_1, g_2\rangle$. This will yield polynomials, namely $g_3, \ldots, g_r$, whose leading monomials are not in $\langle \text{LM}_\prec(g_1), \text{LM}_\prec(g_2)\rangle$. The advantage of this method is twofold. First, since $\text{LM}_\prec(g_3), \ldots, \text{LM}_\prec(g_r) \succ \text{LM}_\prec(g_1), \text{LM}_\prec(g_2)$, they require more queries to the table to be correctly guessed. Yet, such a Gröbner basis computation does not require any more queries. Then, these P-relations may help us determine that the ideal of P-relations is 0-dimensional in $\mathbb{K}(\boldsymbol{t})\langle\boldsymbol{x}\rangle$. This is a necessary condition for the table to be P-finite.

The aim of this section is to extend this approach for guessing P-relations of a table when only considering terms in a cone or when the ideal of relations is stable by the action of a subgroup of $\text{GL}(n)$.

Lemma 4.5. *Let* $\mathcal{T}(C)$ *be a cone of monomials in* $x_1, \ldots, x_n$*, as before. Let us assume that* $f_1, f_2 \in \mathbb{K}[\boldsymbol{t}]\langle\boldsymbol{x}\rangle$ *are both polynomials with monomials in* $\mathcal{T}(\mathbb{N}^n)\times\mathcal{T}(C) = \left\{\boldsymbol{t}^{\boldsymbol{k}}\boldsymbol{x}^{\boldsymbol{i}} \middle| \boldsymbol{x}^{\boldsymbol{i}} \in \mathcal{T}(C)\right\}$*. Then, any polynomial* $f_1a_1 + f_2a_2$ *in the right ideal* $\langle f_1, f_2\rangle$*, such that* $\operatorname{supp} a_1, \operatorname{supp} a_2 \in \mathcal{T}(\mathbb{N}^n)\times\mathcal{T}(C)$*, has its support in* $\mathcal{T}(\mathbb{N}^n)\times\mathcal{T}(C)$ *as well.*

In particular, we can compute a sparse Gröbner basis of $\langle f_1, f_2\rangle$ *with monomials all in* $\mathcal{T}(\mathbb{N}^n)\times\mathcal{T}(C)$ *using Buchberger's algorithm or Faugère's* F_4 *algorithm, restricted to only multiplying the polynomials by monomials in* $\mathcal{T}(\mathbb{N}^n)\times\mathcal{T}(C)$*.*

Proof. We need to prove that if supp f and supp a are in $\mathcal{T}(\mathbb{N}^n) \times \mathcal{T}(C)$, then so is supp fa. By linearity, this comes down to proving that if two monomials $\boldsymbol{t}^{\boldsymbol{\ell}}\boldsymbol{x}^{\boldsymbol{j}}$ and $\boldsymbol{t}^{\boldsymbol{k}}\boldsymbol{x}^{\boldsymbol{i}}$ are in $\mathcal{T}(\mathbb{N}^n) \times \mathcal{T}(C)$, then so is the support of their product. Since

$$\begin{aligned}\boldsymbol{t}^{\boldsymbol{\ell}}\boldsymbol{x}^{\boldsymbol{j}}\boldsymbol{t}^{\boldsymbol{k}}\boldsymbol{x}^{\boldsymbol{i}} &= \boldsymbol{t}^{\boldsymbol{\ell}}\,(\boldsymbol{t}-\boldsymbol{j})^{\boldsymbol{k}}\,\boldsymbol{x}^{\boldsymbol{j}+\boldsymbol{i}}\\ &= \sum_{q_1,\ldots,q_n=0}^{\ell_1,\ldots,\ell_n} \binom{k_1}{q_1}\cdots\binom{k_n}{q_n}(-j_1)^{k_1-q_1}\cdots(-j_n)^{k_n-q_n}t_1^{\ell_1+q_1}\cdots t_n^{\ell_n+q_n}\boldsymbol{x}^{\boldsymbol{j}+\boldsymbol{i}}\end{aligned}$$

and $\boldsymbol{x}^{\boldsymbol{j}+\boldsymbol{i}} \in \mathcal{T}(C)$, $\boldsymbol{t}^{\boldsymbol{\ell}}\boldsymbol{x}^{\boldsymbol{i}}\boldsymbol{t}^{\boldsymbol{k}}\boldsymbol{x}^{\boldsymbol{i}} \in \mathcal{T}(\mathbb{N}^n) \times \mathcal{T}(C)$.

Now, in $\mathcal{T}(\mathbb{N}^n) \times \mathcal{T}(C)$, we can define the *division* of monomials with $m_2|m_1$ if there exists $m_3 \in \mathcal{T}(\mathbb{N}^n) \times \mathcal{T}(C)$ such that $m_1 = m_2m_3$. Then, we can make a new S-polynomial of two polynomials with supports in $\mathcal{T}(\mathbb{N}^n) \times \mathcal{T}(C)$ by considering the LCM in $\mathcal{T}(\mathbb{N}^n) \times \mathcal{T}(C)$ of their leading monomials. □

This lemma shows that the definition of sparse Gröbner bases and the algorithmic techniques to compute them in [24] can be extended to skew-polynomial rings $\mathbb{K}[\boldsymbol{t}]\langle\boldsymbol{x}\rangle$ with commutation rules $t_px_p = x_p(t_p + 1)$.

Using the definitions and notation of Section 3.3.1, we have the following lemma.

Lemma 4.6. *Let G be a finite group of diagonal matrices acting on $t_1, \ldots, t_n, x_1, \ldots, x_n$, then G leaves $t_1, \ldots, t_n$, each, invariant.*

Assume that $f_1, f_2 \in \mathbb{K}[\boldsymbol{t}]\langle\boldsymbol{x}\rangle$ are both G-homogeneous polynomials, then their S-polynomial is also G-homogeneous. Thus, so are all the elements of a reduced Gröbner basis of $\langle f_1, f_2\rangle$.

Proof. There exists a root of unity ζ such that for each matrix in G, there exist integers $\tau_1, \ldots, \tau_n$, $\varepsilon_1, \ldots, \varepsilon_n$ such that for all $1 \le p \le n$, x_p is sent onto $\zeta^{\varepsilon_p}x_p$ and t_p onto $\zeta^{\tau_p}t_p$.

Therefore, $t_px_p - x_pt_p = x_p$ is sent on both $\zeta^{\tau_p}t_p\zeta^{\varepsilon_p}x_p - \zeta^{\varepsilon_p}x_p\zeta^{\tau_p}t_p = \zeta^{\tau_p+\varepsilon_p}\left(t_px_p - x_pt_p\right) = \zeta^{\tau_p+\varepsilon_p}x_p$ and $\zeta^{\varepsilon_p}x_p$. Thus, $\zeta^{\tau_p} = 1$ and G lets t_p invariant. By Definition 3.8, this means that the G-degree of t_p is 0 so that the $\boldsymbol{t}^{\boldsymbol{k}}\boldsymbol{x}^{\boldsymbol{i}}$ and $\boldsymbol{x}^{\boldsymbol{i}}$ have same G-degree.

The S-polynomial of f_1 and f_2 is $f_1\boldsymbol{t}^{\boldsymbol{k}}\boldsymbol{x}^{\boldsymbol{i}} - f_2\frac{\mathrm{LC}_\prec(f_1)}{\mathrm{LC}_\prec(f_2)}\boldsymbol{t}^{\boldsymbol{\ell}}\boldsymbol{x}^{\boldsymbol{j}}$ with $\boldsymbol{t}^{\boldsymbol{k}}\,\mathrm{LM}_\prec(f_1)\boldsymbol{x}^{\boldsymbol{i}} = \boldsymbol{t}^{\boldsymbol{\ell}}\,\mathrm{LM}_\prec(f_2)\boldsymbol{x}^{\boldsymbol{j}} = \mathrm{GCD}\left(\mathrm{LM}_\prec(f_1), \mathrm{LM}_\prec(f_2)\right)$, where $\mathrm{LC}_\prec(f)$ stands for leading coefficient of f, i.e. the coefficient of $\mathrm{LM}_\prec(f)$. Since both terms of the sum have the same leading monomial, it remains to show that multiplying a polynomial by a monomial preserves the G-homogeneity. Since $\boldsymbol{t}^{\boldsymbol{\ell}}\boldsymbol{x}^{\boldsymbol{j}}\boldsymbol{t}^{\boldsymbol{k}}\boldsymbol{x}^{\boldsymbol{i}} = \boldsymbol{t}^{\boldsymbol{\ell}}\,(\boldsymbol{t}-\boldsymbol{j})^{\boldsymbol{k}}\,\boldsymbol{x}^{\boldsymbol{j}+\boldsymbol{i}}$, it is a G-homogeneous polynomial of same G-degree as $\boldsymbol{x}^{\boldsymbol{j}+\boldsymbol{i}}$. Now, the G-degree of $\boldsymbol{x}^{\boldsymbol{j}+\boldsymbol{i}}$ is the sum of the G-degrees of $\boldsymbol{x}^{\boldsymbol{j}}$ and $\boldsymbol{x}^{\boldsymbol{i}}$ and thus of $\boldsymbol{t}^{\boldsymbol{\ell}}\boldsymbol{x}^{\boldsymbol{j}}$ and $\boldsymbol{t}^{\boldsymbol{k}}\boldsymbol{x}^{\boldsymbol{i}}$. □

From Lemmas 4.5 and 4.6, we can compute a Gröbner basis or a sparse Gröbner basis of the ideal spanned by skew-polynomials associated to P-relations in $\mathbb{K}[\boldsymbol{t}]\langle\boldsymbol{x}\rangle$, with commutation rules $t_px_p = x_p(t_p + 1)$, to guess new P-relations in the cone and lattice settings.

Corollary 4.7. *Let G be a finite diagonal matrix group acting on variables $\boldsymbol{t}$ and $\boldsymbol{x}$. Let $I = \langle f_1, \ldots, f_s\rangle \subset \mathbb{K}[\boldsymbol{t}]\langle\boldsymbol{x}\rangle$ be an ideal spanned by G-homogeneous polynomials. Then, one can compute a Gröbner basis of I by using a quasi-commutative variant of the* F_4 *algorithm [18] building $|G|$ Macaulay matrices for each G-degree.*

5. Experiments

In this section, we report on our implementations of the different algorithms of this paper.

We start with the Scalar-FGLM algorithm on a cone, as in Subsection 3.1, in particular for guessing P-relations of tables in Maple 2019. This is an extension of Algorithm 1, see [8]. We investigate Gessel planar walk $\boldsymbol{g}$ in the nonnegative quadrant $\mathbb{N}^2$ with steps in $\{(1,0),(1,1),(-1,0),(-1,-1)\}$ and the 3D-space Walk-43 $\boldsymbol{w}$ of [11] in the nonnegative octant $\mathbb{N}^3$ with steps in $\{(-1,-1,-1),(-1,-1,1),(-1,1,0),(1,0,0)\}$. In particular, we restrict ourselves to a subsequence of each where one index is 0. These walks come naturally with a cone structure: for instance whenever $n \neq 2n' + 2j$, then $g_{n,0,j} = 0$. Likewise, whenever $n \neq 8n' + 2j + 4k$, then $w_{n,0,j,k} = 0$. Thus, it makes sense to look for the relations given by the table terms $g_{2n'+2j,0,j}$ and $w_{8n'+2j+4k,0,j,k}$.

In Table 1, we report on the number of computed relations and the number of relations that do not fail after further testing.

1. The column Full Orthant means that we consider all the table terms $g_{n,0,j}$ and $w_{n,0,j,k}$.
2. The column Half Orthant means that we consider all the table terms $g_{2n',0,j}$ and $w_{2n',0,j,k}$.
3. The column Cone means that we consider all the table terms $g_{2n'+2j,0,j}$ and terms $w_{8n'+2j+4k}$, corresponding to the potential nonzero terms.

We tested two kinds of matrices: matrices that are almost square, with just a little bit more rows than columns, and matrices that have many more rows than columns.

We can notice, as expected in both cases, that by considering only terms on the nonzero cone we guess many fewer false positive P-relations. This happens despite our matrices having fewer rows in the cone setting than in the full orthant setting, i.e. a priori the relations have fewer constraints. This means that amongst these constraints more are linearly independent and that in general the number of linearly dependent rows is responsible for the matrix rank decrease. As a byproduct, this reduces the number of operations.

Type	Cone				Half Orthant				Full Orthant			
	Matrix size	Queries	Relations		Matrix size	Queries	Relations		Matrix size	Queries	Relations	
			Fake	Correct			Fake	Correct			Fake	Correct
$g_{n,0,j}$	444×441	866	11	0	444×443	857	68	0	496×495	946	48	0
$g_{n,0,j}$	631×564	1 174	0	0	961×581	1 506	115	0	$1\,326 \times 661$	1 942	84	0
$g_{n,0,j}$	721×711	1 408	15	8	724×713	1 401	87	0	726×715	1 386	67	0
$g_{n,0,j}$	$1\,951 \times 1\,089$	3 010	0	21	$2\,209 \times 1\,036$	3 196	154	0	$2\,556 \times 1\,001$	3 491	136	6
$w_{n,0,i,j}$	223×211	430	7	1	222×211	411	25	0	220×210	395	24	0
$w_{n,0,i,j}$	444×253	552	2	1	520×260	758	40	0	680×267	912	37	0
$w_{n,0,i,j}$	406×400	799	11	6	406×400	772	40	0	406×400	771	27	0
$w_{n,0,i,j}$	806×522	1 320	2	6	$1\,200 \times 550$	1 716	78	0	$1\,540 \times 589$	2 073	68	0

Table 1: Guessing fake and correct P-relations with Algorithm 1 for P-relations on a cone.

In Table 2, we consider the FGLM application, presented in Subsection 3.3, running on an Intel Xeon E-2286M with 32 GB of RAM. We compute first a drl Gröbner basis of an ideal invariant by the action of a finite diagonal group $\mathbb{Z}/n\mathbb{Z}$ and then the eliminating polynomial of the last variable. The number n in the names of the systems denotes the number of variables and the computations were done modulo $2^{30} < p < 2^{31}$ such that a primitive nth root of unity exists in

$\mathbb{Z}/p\mathbb{Z}$. The SPARSE-FGLM algorithm [22, 23] has been implemented in C, as part of the `msolve` library [6, 7], it generates a scalar table first and then guesses its C-relation with the Berlekamp–Massey algorithm. Notice that the table generation is the bottleneck of the method, but it is also the part that benefit the most from the occurred speedup. In the column SPARSE-FGLM, we use the whole multiplication matrix, while in the column LATTICE SPARSE-FGLM, we use the n nonzero blocks of the multiplication matrix to perform the computations and taking advantage of the action of $\mathbb{Z}/n\mathbb{Z}$. We also compare with MAPLE 2019 where we use `Groebner:-FGLM` to compute a Gröbner basis for an ordering eliminating all the variables but the last one. As expected by Proposition 3.10, using the splitting of the multiplication matrix allows us to divide the computation time by n.

Type	Degree	SPARSE-FGLM		LATTICE SPARSE-FGLM		LATTICE speedup		MAPLE
		Seq. gen.	Guess.	Seq. gen.	Guess.	Seq. gen.	Guess.	
Cyclic-6	156	1 470	10	200	2.3	7.35	4.35	120 000
Cyclic-7	924	64 000	56	5 200	8.3	12.3	6.75	13 *s*
Random-3	294	3 100	18	1 100	6.8	2.82	2.65	510 000
Random-3 bis	3090	470 000	170	83 000	63	5.66	2.70	–
Random-4	896	69 000	53	8 600	14	8.02	3.79	2 000 *s*
Random-5	2000	386 000	110	35 000	24	11.0	4.58	49 *s*
Random-6	1656	330 000	91	26 000	17	12.7	5.35	1 200 *s*
Random-10	4160	13 *s*	250	37 000	26	351	9.62	–

Table 2: FGLM application with the action of $\mathbb{Z}/n\mathbb{Z}$ (in μs).

Acknowledgments

We thank the anonymous referees for their careful reading and their helpful comments to improve this paper. The authors are supported by the joint ANR-FWF ANR-19-CE48-0015 ECARP project, the ANR grants ANR-18-CE33-0011 SESAME and ANR-19-CE40-0018 DE RERUM NATURA projects, the PGMO grant CAMISADO, grant FA8665-20-1-7029 of the EOARD-AFOSR and and the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement N. 813211 (POEMA).

References

[1] 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. URL: `http://dx.doi.org/10.1137/S0895479892230031`, doi:`10.1137/S0895479892230031`.

[2] Bender, M.R., Faugère, J.Ch., Tsigaridas, E., 2018. Towards Mixed Gröbner Basis Algorithms: The Multihomogeneous and Sparse Case, in: Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation, ACM, New York, NY, USA. pp. 71–78. URL: `http://doi.acm.org/10.1145/3208976.3209018`, doi:`10.1145/3208976.3209018`.

[3] Berlekamp, E., 1968. Nonbinary BCH decoding. IEEE Trans. Inform. Theory 14, 242–242. doi:`10.1109/TIT.1968.1054109`.

[4] Berthomieu, J., Boyer, B., Faugère, J.Ch., 2015. Linear Algebra for Computing Gröbner Bases of Linear Recursive Multidimensional Sequences, in: Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation, ACM, New York, NY, USA. pp. 61–68. URL: `http://doi.acm.org/10.1145/2755996.2756673`, doi:`10.1145/2755996.2756673`.

[5] Berthomieu, J., Boyer, B., Faugère, J.Ch., 2017. Linear Algebra for Computing Gröbner Bases of Linear Recursive Multidimensional Sequences. Journal of Symbolic Computation 83, 36–67. URL: https://hal.inria.fr/hal-01253934, doi:10.1016/j.jsc.2016.11.005. special issue on the conference ISSAC 2015: Symbolic computation and computer algebra.

[6] Berthomieu, J., Eder, Ch., Safey El Din, M., 2021a. msolve: A library for solving polynomial systems, in: Proceedings of the 2021 on International Symposium on Symbolic and Algebraic Computation, Association for Computing Machinery, New York, NY, USA. pp. 51—-58. doi:10/gk8549.

[7] Berthomieu, J., Eder, Ch., Safey El Din, M., 2021b. msolve: A library for solving polynomial systems. https://msolve.lip6.fr/.

[8] Berthomieu, J., Faugère, J.Ch., 2016. Guessing Linear Recurrence Relations of Sequence Tuples and P-recursive Sequences with Linear Algebra, in: Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, ACM, New York, NY, USA. pp. 95–102. URL: http://doi.acm.org/10.1145/2930889.2930926, doi:10.1145/2930889.2930926.

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

[10] Bose, R., Ray-Chaudhuri, D., 1960. On a class of error correcting binary group codes. Information and Control 3, 68–79. URL: http://www.sciencedirect.com/science/article/pii/S0019995860902874, doi:http://dx.doi.org/10.1016/S0019-9958(60)90287-4.

[11] Bostan, A., Bousquet-Mélou, M., Kauers, M., Melczer, S., 2016. On 3-dimensional lattice walks confined to the positive octant. Annals of Combinatorics 20, 661–704. URL: https://doi.org/10.1007/s00026-016-0328-7, doi:10.1007/s00026-016-0328-7.

[12] Bousquet-Mélou, M., Petkovšek, M., 2003. Walks confined in a quadrant are not always D-finite. Theoret. Comput. Sci. 307, 257–276. URL: http://www.sciencedirect.com/science/article/pii/S0304397503002196, doi:http://dx.doi.org/10.1016/S0304-3975(03)00219-6. Random Generation of Combinatorial Objects and Bijective Combinatorics.

[13] Brachat, J., Comon, P., Mourrain, B., Tsigaridas, E.P.P., 2010. Symmetric tensor decomposition. Linear Algebra Appl. 433, 1851–1872.

[14] Brent, R.P., Gustavson, F.G., Yun, D.Y., 1980. Fast solution of Toeplitz systems of equations and computation of Padé approximants. Journal of Algorithms 1, 259–295. URL: http://www.sciencedirect.com/science/article/pii/0196677480900139, doi:https://doi.org/10.1016/0196-6774(80)90013-9.

[15] Cantor, D.G., Kaltofen, E., 1991. On fast multiplication of polynomials over arbitrary algebras. Acta Informatica 28, 693–701.

[16] Cox, D., Little, J., O'Shea, D., 2015. Ideals, Varieties, and Algorithms. Undergraduate Texts in Mathematics. fourth ed., Springer, New York. An Introduction to Computational Algebraic Geometry and Commutative Algebra.

[17] Elkadi, M., Mourrain, B., 2007. Introduction à la résolution des systèmes polynomiaux. volume 59 of *Mathématiques et Applications*. Springer.

[18] Faugère, J.Ch., 1999. A new efficient algorithm for computing Gröbner bases (F_4). Journal of Pure and Applied Algebra 139, 61–88. URL: http://www.sciencedirect.com/science/article/pii/S0022404999000055, doi:https://doi.org/10.1016/S0022-4049(99)00005-5.

[19] Faugère, J.Ch., 2002. A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5), in: Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, ACM, New York, NY, USA. pp. 75–83. URL: http://doi.acm.org/10.1145/780506.780516, doi:10.1145/780506.780516.

[20] Faugère, J.Ch., Gaudry, P., Huot, L., Renault, G., 2014a. Sub-cubic change of ordering for Gröbner basis: A probabilistic approach, in: Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation, Association for Computing Machinery, New York, NY, USA. p. 170–177. URL: https://doi.org/10.1145/2608628.2608669, doi:10.1145/2608628.2608669.

[21] Faugère, J.Ch., Gianni, P., Lazard, D., Mora, T., 1993. Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering. J. Symbolic Comput. 16, 329–344. doi:http://dx.doi.org/10.1006/jsco.1993.1051.

[22] Faugère, J.Ch., Mou, C., 2011. Fast algorithm for change of ordering of zero-dimensional Gröbner bases with sparse multiplication matrices, in: Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation, ACM, New York, NY, USA. pp. 115–122. URL: http://doi.acm.org/10.1145/1993886.1993908, doi:10.1145/1993886.1993908.

[23] Faugère, J.Ch., Mou, C., 2017. Sparse FGLM algorithms. Journal of Symbolic Computation 80, 538–569. doi:10.1016/j.jsc.2016.07.025.

[24] Faugère, J.Ch.., Spaenlehauer, P.J., Svartz, J., 2014b. Sparse Gröbner bases: The unmixed case, in: Proceedings

of the 39th International Symposium on Symbolic and Algebraic Computation, ACM, New York, NY, USA. pp. 178–185. URL: http://doi.acm.org/10.1145/2608628.2608663, doi:10.1145/2608628.2608663.

[25] Faugère, J.Ch., Svartz, J., 2013. Gröbner bases of ideals invariant under a commutative group: the non-modular case, in: Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation, ACM, New York, NY, USA. pp. 347–354. URL: http://doi.acm.org/10.1145/2465506.2465944, doi:10.1145/2465506.2465944.

[26] Gianni, P., Mora, T., 1989. Algebraic solution of systems of polynomial equations using Groebner bases, in: Huguet, L., Poli, A. (Eds.), Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 247–257.

[27] Hocquenghem, A., 1959. Codes correcteurs d'erreurs. Chiffres 2, 147–156.

[28] Kauers, M., Verron, T., 2019. Why you should remove zeros from data before guessing. ACM Commun. Comput. Algebra 53, 126–129. URL: https://doi.org/10.1145/3377006.3377017, doi:10.1145/3377006.3377017.

[29] Keller-Gehrig, W., 1985. Fast algorithms for the characteristics polynomial. Theoretical Computer Science 36, 309–317. URL: https://www.sciencedirect.com/science/article/pii/0304397585900490, doi:https://doi.org/10.1016/0304-3975(85)90049-0.

[30] Koppenhagen, U., Mayr, E.W., 1999. An Optimal Algorithm for Constructing the Reduced Gröbner Basis of Binomial Ideals. Journal of Symbolic Computation 28, 317–338. URL: http://www.sciencedirect.com/science/article/pii/S0747717199902857, doi:https://doi.org/10.1006/jsco.1999.0285.

[31] Levandovskyy, V., 2005. Non-commutative Computer Algebra for polynomial algebras: Gröbner bases, applications and implementation. doctoralthesis. Technische Universität Kaiserslautern. URL: http://nbn-resolving.de/urn:nbn:de:hbz:386-kluedo-18830.

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

[33] Mezzarobba, M., 2019. Truncation Bounds for Differentially Finite Series. Annales Henri Lebesgue 2, 99–148. URL: https://ahl.centre-mersenne.org/item/AHL_2019__2__99_0, doi:10.5802/ahl.17.

[34] Mourrain, B., 2017. Fast Algorithm for Border Bases of Artinian Gorenstein Algebras, in: Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation, ACM, New York, NY, USA. pp. 333–340. URL: http://doi.acm.org/10.1145/3087604.3087632, doi:10.1145/3087604.3087632.

[35] Neiger, V., Schost, É., 2020. Computing syzygies in finite dimension using fast linear algebra. Journal of Complexity 60, 101502. URL: https://www.sciencedirect.com/science/article/pii/S0885064X20300455, doi:https://doi.org/10.1016/j.jco.2020.101502.

[36] Sakata, S., 1988. Finding a minimal set of linear recurring relations capable of generating a given finite two-dimensional array. J. Symbolic Comput. 5, 321–337. URL: http://www.sciencedirect.com/science/article/pii/S0747717188800336, doi:http://dx.doi.org/10.1016/S0747-7171(88)80033-6.

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

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

[39] Steidel, S., 2013. Gröbner bases of symmetric ideals. Journal of Symbolic Computation 54, 72–86. URL: http://www.sciencedirect.com/science/article/pii/S074771711300014X, doi:https://doi.org/10.1016/j.jsc.2013.01.005.