On FGLM Algorithms with Tate Algebras

Xavier Caruso, Tristan Vaccon, Thibaut Verron

On FGLM Algorithms with Tate Algebras

Xavier Caruso
Université de Bordeaux, CNRS, INRIA
Bordeaux, France
xavier.caruso@normalesup.org

Tristan Vaccon
Université de Limoges; CNRS, XLIM UMR 7252
Limoges, France
tristan.vaccon@unilim.fr

Thibaut Verron
Johannes Kepler University, Institute for Algebra
Linz, Austria
thibaut.verron@jku.at

ABSTRACT

Tate introduced in [Ta71] the notion of Tate algebras to serve, in the context of analytic geometry over the p-adics, as a counterpart of polynomial algebras in classical algebraic geometry. In [CVV19, CVV20] the formalism of Gröbner bases over Tate algebras has been introduced and advanced signature-based algorithms have been proposed. In the present article, we extend the FGLM algorithm of [FGLM93] to Tate algebras. Beyond allowing for fast change of ordering, this strategy has two other important benefits. First, it provides an efficient algorithm for changing the radii of convergence which, in particular, makes effective the bridge between the polynomial setting and the Tate setting and may help in speeding up the computation of Gröbner basis over Tate algebras. Second, it gives the foundations for designing a fast algorithm for interreduction, which could serve as basic primitive in our previous algorithms and accelerate them significantly.

CCS CONCEPTS

• **Computing methodologies** → **Algebraic algorithms**.

KEYWORDS

Algorithms, Gröbner bases, Tate algebra, FGLM algorithm, p-adic precision

ACM Reference Format:
Xavier Caruso, Tristan Vaccon, and Thibaut Verron. 2021. On FGLM Algorithms with Tate Algebras. In . ACM, New York, NY, USA, 9 pages.

1 INTRODUCTION

Lying at the intersection of geometry and number theory, one finds p-adic geometry. A paramount part of this theory is the study of p-adic analytic varieties, first defined by Tate in [Ta71] (see also [FP04]). They have played a key role in many developments of number theory (*e.g.* p-adic cohomologies [LS07], p-adic modular forms [Go88]). The main algebraic objects upon which Tate's geometry is built are Tate algebras and their ideals, formed of convergent multivariate power series over a complete discrete valuation field K (*e.g.* $K = \mathbb{Q}_p$).

This work was supported by the ANR project CLap–CLap (ANR-18-CE40-0026-01). T. Verron was supported by the Austrian FWF grant P31571-N32.

Conference'21, July 2021, Washington, DC, USA

In earlier papers [CVV19, CVV20], the authors showed that it is possible to define and compute Gröbner bases of Tate ideals with coefficients in $\mathbb{Z}_p$ or $\mathbb{Q}_p$, and that the definitions are compatible with the usual theory on polynomials over the residue field $\mathbb{F}_p$ or over the coefficient ring. A major limitation of the algorithms is the increasing cost of reductions as the precision grows. Our previous paper [CVV20] addresses the case of expensive reductions to zero, through the use of signature algorithms, but computing the result of non-trivial reductions remains expensive. Another question left open was whether it is possible to exploit overconvergence properties, namely the knowledge that the series we are working with satisfy a stronger convergence condition.

In the present paper, we adapt the classical FGLM algorithm to the case of Tate series, and we show that it gives answers to both questions, in the case of zero-dimensional ideals. Precisely, we prove the following theorem.

Theorem 1.1. *Let $K\{\mathrm{X};\mathrm{r}\}$[1] and $K\{\mathrm{X};\mathrm{u}\}$ be two Tate algebras with $K\{\mathrm{X};\mathrm{r}\} \subset K\{\mathrm{X};\mathrm{u}\}$.*

There exists an algorithm that takes as input a reduced Gröbner basis G of a 0-dimensional ideal I of $K\{\mathrm{X};\mathrm{r}\}$ with respect to a given monomial ordering and output a Gröbner basis of the ideal $I{\cdot}K\{\mathrm{X};\mathrm{u}\}$ of $K\{\mathrm{X};\mathrm{u}\}$ for another given monomial ordering.

Moreover, if n denotes the number of variables, if δ is the dimension of the quotient $K\{\mathrm{X};\mathrm{r}\}/I$ and if prec *is the precision at which the result is output, the complexity of this algorithm is:*

- $O^{\sim}(n\delta^3\mathrm{prec})$ *operations in the base field K for a general K,*
- $O^{\sim}(n\delta^3\mathrm{prec}\cdot\log p)$ *bit operations when $K = \mathbb{Q}_p$.*

We underline that, although the classical FGLM algorithm only concerns change of ordering, our version also permits to change the radii of convergence of the underlying Tate algebra (namely the parameters r and u), and then provides efficient tools for dealing with the aforementioned overconvergence situation. In the extreme case where r is infinite, it makes effective the bridge between polynomials and Tate series, that is between classical algebraic geometry and rigid geometry. On a different note, being able to perform such a change of ordering opens up algorithmic strategies for overconvergent series, by giving freedom in the choice of the convergence radii.

An additional important outcome of our algorithm is that it can be slightly modified in order to accept certain nonreduced Gröbner bases as input. Hence, in many cases, calling it with the same radii of convergence and the same ordering as input and output, already performs a nontrivial operation: the interreduction of the input Gröbner basis. Moreover, it has a controlled complexity and

[1] Here K denotes the base field and r encodes the radii of convergence of our series; we refer to §2.1 for the precise definitions.

performs actually very well in practice (contrarily to the naive reduction algorithm). Since the intermediate interreduction of Gröbner bases is often the bottleneck in Buchberger and signature algorithms in the Tate setting, using our FGLM algorithm (or an adaptation of it) at this step could lead to a significant speed-up.

Strategy and ingredients. In the classical setting, the key step of the FGLM algorithm is to convert back and forth between Gröbner bases and the so-called multiplication matrices, which are defined as the multiplication maps by the variables in the quotient space. Performing the change of ordering on those multiplication matrices then reduces to basic linear algebra. Still in the classical case, thanks to the structure of normal forms, it can be shown that all steps can be done in sub-cubic time in the number of solutions.

In the Tate setting, Gröbner bases are defined using a term ordering, taking into account both a monomial ordering in the usual sense and a weight taking into account the degree of the monomials, the valuation of the coefficient and the convergence radius of the series in the algebra. It is the reason why we will eventually be able to change all these parameters at the same time. However, this feature also implies new difficulties.

Firstly, in the construction of the multiplication matrices, the structure of the normal forms does not allow us to read the values in one pass. Instead, we prove that an iterative process converges to the correct value of the matrices, and we show how this process can be done in different ways, including, for some particular base fields, the option of using relaxed arithmetic [vdH97, BvdHL11], which eventually leads to a significant improvement of the efficiency.

Secondly, if the change of ordering incurs a change of convergence radii, the size of the quotient algebra might change. We show that it is possible to recover multiplication matrices over the correct quotient by separating eigenspaces depending on the valuation of the eigenvalues. The reconstruction of the final Gröbner basis is finally achieved using the classical strategy in the residue field, and then lifting the basis.

Organization of the article. In Section 2, we introduce the notations and discuss some primitives of linear algebras over nonarchimedian fields which will be used repeatedly later on. The computation of multiplication matrices is addressed in Section 3. In Section 4, we consider the question of changing radii of convergence and design our final algorithm.

2 SETTING AND PRELIMINARIES

Throughout this article, we consider a field K equipped with a discrete valuation val for which it is complete. We denote its ring of integers by K° and fix a uniformizer π of K. The quotient K°/π is called the residue field of K and will be denoted by $\bar{K}$ in what follows. Classical examples of such fields are $K = \mathbb{Q}_p$ (equipped with the p-adic valuation) and $k((T))$ (equipped with the T-adic valuation) for any base field k.

The complexity statements are given with the usual asymptotic notations $O(f)$ and $O^{\tilde{}}(f) = O(f\log(f)^n)$ for some n.

We will consider two different models of complexity: arithmetic complexity, counting operations in K or K°, and base complexity, taking into account the precision. In the case of equal characteristic (*i.e.* $\mathrm{char}\ K = \mathrm{char}\ \bar{K}$), such as $k((T))$, the base complexity counts operations in the residue field, and the correspondence between both models satistfies:

$$(\text{Arithmetic complexity}) = O^{\tilde{}}\big((\text{Base complexity})\cdot \text{prec}\big).$$

where prec stands for the working precision. On the contrary, in the case of mixed characteristic, such as $\mathbb{Q}_p$, the base complexity counts bit operations. When the residue field is finite, the correspondence between both models satisfies:

$$(\text{Arithmetic complexity}) = O^{\tilde{}}\big((\text{Base complexity})\cdot \text{prec}\cdot \log|\bar{K}|\big).$$

2.1 Tate algebras and ideals

In order to fix notations, we briefly recall the definition of Tate algebras and the theory of Gröbner bases over them. Let $\mathrm{r} = (r_1,\dots,r_n) \in \mathbb{Q}^n$. The *Tate algebra* $K\{\mathrm{X};\mathrm{r}\}$ is defined by:

$$K\{\mathrm{X};\mathrm{r}\} := \left\{ \sum_{\mathrm{i}\in\mathbb{N}^n} a_\mathrm{i}\mathrm{X}^\mathrm{i} \text{ s.t. } a_\mathrm{i}\in K \text{ and } \mathrm{val}(a_\mathrm{i}) - \mathrm{r}\cdot\mathrm{i} \xrightarrow[|\mathrm{i}|\to+\infty]{} +\infty \right\}$$

The tuple r is called the convergence log-radii of the Tate algebra. We define the Gauss valuation of a term $a_\mathrm{i}\mathrm{X}^\mathrm{i}$ as $\mathrm{val}(a_\mathrm{i}\mathrm{X}^\mathrm{i}) = \mathrm{val}(a_\mathrm{i}) - \mathrm{r}\cdot\mathrm{i}$, and the Gauss valuation of $\sum a_\mathrm{i}\mathrm{X}^\mathrm{i} \in K\{\mathrm{X};\mathrm{r}\}$ as the minimum of the Gauss valuations of its terms. The integral Tate algebra ring $K\{\mathrm{X};\mathrm{r}\}^\circ$ is the subring of $K\{\mathrm{X};\mathrm{r}\}$ consisting of elements with nonnegative valuation. In what follows, when $\mathrm{r} = (0,\dots,0)$, we will simply write $K\{\mathrm{X}\}$ instead of $K\{\mathrm{X};(0,\dots,0)\}$.

We fix a classical *monomial order* $\leq_m$ on the set of monomials X^i. Given two terms $a\mathrm{X}^\mathrm{i}$ and $b\mathrm{X}^\mathrm{j}$ (with $a,b\in K^\times$), we write $a\mathrm{X}^\mathrm{i} < b\mathrm{X}^\mathrm{j}$ if $\mathrm{val}(a\mathrm{X}^\mathrm{i}) > \mathrm{val}(b\mathrm{X}^\mathrm{j})$, or $\mathrm{val}(a\mathrm{X}^\mathrm{i}) = \mathrm{val}(b\mathrm{X}^\mathrm{j})$ and $\mathrm{X}^\mathrm{i} <_m \mathrm{X}^\mathrm{j}$. The leading term of a Tate series $\sum a_\mathrm{i}\mathrm{X}^\mathrm{i} \in K\{\mathrm{X};\mathrm{r}\}$ is, by definition, its maximal term.

A Gröbner basis of an ideal I of $K\{\mathrm{X};\mathrm{r}\}$ is, by definition, a family $(g_1,\dots,g_s)$ of elements of I with the property that for all $f\in I$, there exists an index $i\in\{1,\dots,s\}$ such that $\mathrm{LT}(g_i)$ divides $\mathrm{LT}(f)$. A Gröbner basis $(g_1,\dots,g_s)$ is *reduced* if given a term t of g_i which is not the leading term, t is not divisible by any $\mathrm{LT}(g_j)$. The following theorem is proved in [CVV19].

Theorem 2.1. *(1) Any ideal of $K\{\mathrm{X};\mathrm{r}\}$ admits a Gröbner basis.*

(2) If $\mathrm{r} = (0,\dots,0)$ and I is an ideal of $K\{\mathrm{X}\}$, a family $G = (g_1,\dots,g_s)$ consisting of elements of $K\{\mathrm{X}\}$ with Gauss valuation 0 is a Gröbner basis of I if and only if its reduction modulo π is a classical Gröbner basis of the quotient ideal $(I\cap K\{\mathrm{X};\mathrm{r}\}^\circ)/\pi(I\cap K\{\mathrm{X};\mathrm{r}\}^\circ)$ of $\bar{K}[\mathrm{X}]$ for $\leq_m$.

In the present article, we will be particularly interested in 0-dimensional ideals. By definition, I is such an ideal if the quotient $K\{\mathrm{X};\mathrm{r}\}/I$ is a finite dimensional K-vector space. If I is a 0-dimensional ideal, the set:

$$B = \left\{ \mathrm{X}^\mathrm{i} \text{ with } i\in\mathbb{N}^n \text{ and } \mathrm{X}^\mathrm{i}\notin \mathrm{LT}(I) \right\}$$

is finite and forms a K-basis of $K\{\mathrm{X};\mathrm{r}\}/I$. It is called the *staircase* of G. Moreover, if we are given a Gröbner basis $(g_1,\dots,g_s)$ of I, the staircase B consists of all monomials X^i which are not divisible by any $\mathrm{LT}(g_j)$ for j varying in $\{1,\dots,s\}$. This observation implies in particular that any *reduced* Gröbner basis of a 0-dimensional ideal I consists only of polynomials.

Algorithm 1: `Big`(f, s)

Input : $f \in K^{\delta\times\delta}$, $s \in \mathbb{R}$.
Output: A basis S of $\mathrm{Big}_s(f)$.

1.1 $\chi_f \leftarrow$ `charpoly`(f); // use [KV04] or [CRV17]
1.2 Write $\chi_f = \chi_{\mathrm{Small},s,f}\chi_{\mathrm{Big},s,f}$; // use [CRV16]
1.3 $g \leftarrow \chi_{\mathrm{Big},s,f}(f)$; // use [PS73]
1.4 $S \leftarrow \ker g$; // use pnumerical kernel from [KV20, §2.2.1]
1.5 **return** S

2.2 Linear algebra

It is an understatement to say that the FGLM strategy relies heavily on linear algebra. In the Tate setting, this assertion is even more true and new basic operations in linear algebra, which are specific to non-archimedean base fields, will be needed. The aim of this subsection is to review briefly these operations.

Slope decomposition. Let V be a finite K-dimensional vector space and let $f : V \to V$ be a K-linear mapping. Let χ_f be the characteristic polynomial of f. Given an auxiliary real number s, one can factor χ_f as a product $\chi_f = \chi_{\mathrm{Big},s,f} \times \chi_{\mathrm{Small},s,f}$ where $\chi_{\mathrm{Big},s,f}$ (resp. $\chi_{\mathrm{Small},s,f}$) is the factor corresponding to all roots (in an algebraic closure) of valuation $< s$ (resp. valuation $\geq s$). Moreover, both $\chi_{\mathrm{Big},s,M}$ and $\chi_{\mathrm{Small},s,M}$ have coefficients in K. Letting $\mathrm{Big}_s(f)$ denote the kernel of $\chi_{\mathrm{Big},s,f}(f)$ and $\mathrm{Small}_s(f)$ denote that of $\chi_{\mathrm{Small},s,f}(f)$, the above factorization corresponds to a decomposition of V as a direct sum $V = \mathrm{Big}_s(f) \oplus \mathrm{Small}_s(f)$. Computing efficiently this decomposition is a basic task in linear algebra over non-archimedean fields.

In this article, we assume that we are given a routine `Big` which takes as input (f, s) and outputs (a basis of) the subspace $\mathrm{Big}_s(f)$. A naive implementation of the procedure `Big` is reported in Algorithm 1. It has cubic complexity in the dimension of V (which will be enough for our applications) but has the advantage of being numerically stable.

K°-modules and saturation. As before, we let V be a finite dimensional K-vector space. We recall basic facts about sub-K°-modules of V and their algorithmic. If V is equipped with a distinguished basis, one can represent a finitely generated sub-K°-module of V by the matrix M whose columns are the generators of L. Performing column reduction, one can always assume that M is under Hermite normal form. With this additional assumption, it is uniquely determined by L.

Let L_1 and L_2 be sub-K°-modules of V, represented by the square matrices M_1 and M_2 respectively. The sum $L = L_1 + L_2$ is then generated by the columns of the block matrix:

$$M = \begin{pmatrix} M_1 \mid M_2 \end{pmatrix}$$

Computing the Hermite normal form of M, one obtains a canonical matrix representating L. The cost of this computation is cubic in the dimension of V with a naive algorithm.

We now assume that we are given a sub-K°-module $L \subset V$ together with a K-linear endomorphism $f : V \to V$. The *saturation* of L with respect to f is the sub-K°-module of V defined by:

$$\mathrm{Sat}_f(L) = L + f(L) + f^2(L) + \cdots + f^n(L) + \cdots$$

Algorithm 2: `Saturate`(f, L)

Input : a K-linear map $f : V \to V$ s.t. $V = \mathrm{Small}_0(f)$, a finitely generated K°-module $L \subset V$
Output: $\mathrm{Sat}_f(L)$

2.1 $S \leftarrow L$; $g \leftarrow f$;
2.2 $w \leftarrow \lceil \log_2 \dim V \rceil$;
2.3 **for** $k \in [\![1, w]\!]$ **do**
2.4 $\quad S \leftarrow S + g(S)$;
2.5 $\quad g \leftarrow g^2$;
2.6 **return** S

Lemma 2.2. *We assume that $V = \mathrm{Small}_0(f)$. Then:*

$$\mathrm{Sat}_f(L) = L + f(L) + f^2(L) + \cdots + f^{\dim V - 1}(L).$$

In particular, if L is finitely generated, then $\mathrm{Sat}_f(L)$ is also.

Proof. The assumption on f implies that the coefficients of χ_f are all in K°. From Cayley-Hamilton theorem, we deduce that f^δ is a linear combination with coefficients in K° of the f^i's with $i < \delta$. The lemma follows. □

The routine `Saturate` presented in Algorithm 2 computes $\mathrm{Sat}_f(L)$ under the assumption that L is finitely generated and $V = \mathrm{Small}_0(f)$. Indeed, one checks by induction that after the k-th iteration of the loop, one has $g = f^{2^k}$ and:

$$S = L + f(L) + f^2(L) + \cdots + f^{2^k - 1}(L).$$

Therefore, when $2^k \geq \delta$, we find $S = \mathrm{Sat}_f(L)$. The complexity of Algorithm 2 is equal to the cost of $O(\log \delta)$ Hermite reductions. If we use the naive algorithm for this task, we obtain an algorithm of arithmetic complexity $O^{\tilde{}}(\delta^3)$.

Remark 2.3. For a general K-linear mapping f, one always has:

$$\mathrm{Sat}_f(L) = \mathrm{Big}_0(f) + L + f(L) + f^2(L) + \cdots + f^{\dim V - 1}(L)$$

provided that L spans V as a K-vector space. Under this assumption, one can then combine Algorithms 1 and 2 to compute the saturation of L with respect to f even when $\mathrm{Small}_0(f) \subsetneq V$.

3 MULTIPLICATION MATRICES

Throughout this section, we fix a tuple $\mathrm{r} = (r_1, \ldots, r_n)$ and consider the Tate algebra $K\{\mathrm{X}; \mathrm{r}\}$. We consider in addition a 0-dimensional ideal I of $K\{\mathrm{X}; \mathrm{r}\}$ and assume that we are given a Gröbner basis $G = (g_1, \ldots, g_s)$ of I.

The first step in the FGLM algorithm is the computation of the matrices of multiplication by the variables on the quotient $K\{\mathrm{X}; \mathrm{r}\}/I$ (which has finite dimension by assumption). We recall that a K-basis of $K\{\mathrm{X}; \mathrm{r}\}/I$ is given by the staircase B, which consists of all monomials m with $m \notin \mathrm{LT}(I)$. We let T_i be the matrix of the multiplication by X_i with respect to this basis. Observe that the (μ, m)-entry of T_i has valuation at least:

$$v_{\mu,m} = \mathrm{val}(mX_i) - \mathrm{val}(\mu) = \mathrm{val}(m) - r_i - \mathrm{val}(\mu).$$

Our goal is to design an algorithm for computing the T_i's. In order to express our complexity estimates, we introduce two important parameters. This first one is the degree of the ideal $\delta = |B| =$

$\dim K\{\mathrm{X};\mathrm{r}\}/I$. The second one, denoted by ε, is the size of the boundary of the staircase defined as $\bar{B} \setminus B$ with

$$\bar{B} = \left\{ X_i m \,:\, i \in \{1,\dots,n\}, m \in B \right\}.$$

Obviously the cardinality of $\bar{B}$ is at most $n\delta$; thus $\varepsilon \leq n\delta$ as well.

The theorem we are going to prove is the following (we refer to the beginning of Section 2 for the definition of the arithmetic and base complexity).

Theorem 3.1. *There exists an algorithm that takes as input a reduced Gröbner basis of G and outputs the multiplication matrices T_i with (μ, m)-entry known at precision $O(\pi^{\mathrm{prec}+v_{\mu,m}})$ for a cost of $O(\varepsilon\delta^2\mathrm{prec})$ arithmetic operations.*

Besides, if the base field K is either a Laurent series field or $\mathbb{Q}_p$, the above complexity can be lowered to $\tilde{O}(\varepsilon\delta^2\mathrm{prec})$ base operations.

We will also present an algorithm accepting as input certain nonreduced Gröbner basis. This variant is interesting because, in some cases, it will eventually provide a fast algorithm for interreducing Gröbner basis.

3.1 Iterative algorithm

Throughout this subsection, we assume that $G = (g_1, \dots, g_s)$ is *reduced* and $\mathrm{r} = (0, \dots, 0)$. We will explain later on how these assumptions can be relaxed. For simplicity, we assume in addition that the g_i's are all monic (*i.e.* the coefficients of their leading terms are 1). This hypothesis is of course harmless since renormalizing the g_i's and making them monic does not affect the fact that G is a Gröbner basis.

Computing the T_i's amounts to computing the normal forms of m modulo I for all m in $\bar{B}$. In a classical setting, this can be done iteratively with linear algebra, by considering the monomials following the monomial order. Indeed, for m in B and $i \in \{1,\dots,n\}$, if $X_i m \notin B$, either $X_i m$ is a leading monomial in G, or there exists $\mu \notin B$ such that $X_i m = X_j \mu$, and then $\mathrm{NF}(X_i m) = X_j \,\mathrm{NF}(\mu)$. In the classical setting, the normal form of a monomial μ only involves monomials in B strictly smaller than μ, so $X_j \,\mathrm{NF}(\mu)$ only involves monomials in $\bar{B}$ strictly smaller than $X_i m$. This allows to write $\mathrm{NF}(X_i m)$ as a linear combination of already computed normal forms.

In the case of Tate term orderings, similarly to what was observed for example for tropical orderings [IVY20], the normal form of a monomial μ can involve all monomials of B, and computing the wanted normal forms *a priori* requires solving a large nonlinear system of equations.

However, because Tate Gröbner basis are just classical Gröbner basis when they are reduced modulo π (Theorem 2.1), the above strategy allows to recover the value of the multiplication matrices modulo π. Following the same computations again lifts the multiplication matrices to coefficients in K°/π^2, and so on and so forth.

The algorithm formalizing this idea is described in Algorithm 3. For $P \in K[\mathrm{X}]$ with support contained in B, the notation $[P]$ represents the vector of coefficients of P in the basis B. With that notation, given a monomial $m \in B$ and a matrix M with rows and columns indexed by B, $M \cdot [m]$ is the column of M corresponding to m.

Algorithm 3: `MulMat_iter`(G, prec)

Input : a reduced Gröbner basis G of the ideal $I \subset K\{\mathrm{X}\}$,
an integer prec such that all elements of G are known at precision prec

Output : $T_1, \dots, T_n$, the multiplication matrices over $K\{\mathrm{X}\}/I$ (w.r.t the basis B) modulo π^{prec}

3.1 $B \leftarrow \{m$ monomials not divisible by any $\mathrm{LT}(g), g \in G\}$;
3.2 $T_i = (c_{i,m,m'}) \leftarrow$ zero matrices of size $\delta \times \delta$, with rows and columns indexed by B, for all $i \in [\![1, n]\!]$;
3.3 **for** k from 0 to prec **do**
3.4 **for** $i \in [\![1, n]\!]$, $m \in B$ in increasing order of $X_i m$ **do**
3.5 **if** $X_i m \in B$ **then**
3.6 $T_i \cdot [m] \leftarrow [X_i m]$;
3.7 **else if** $X_i m = \mathrm{LT}(g)$ for some $g \in G$ **then**
3.8 $T_i \cdot [m] \leftarrow [g - \mathrm{LT}(g)]$;
3.9 **else**
3.10 Write $m = X_j m'$ for some $m' \notin B$;
3.11 $T_i \cdot [m] \leftarrow T_j \cdot (T_i \cdot [m'])$;
3.12 **return** $T_1, \dots, T_n$

For $i \in \{1,\dots,n\}, \mu, m \in B$, we denote by $c_{i,\mu,m}$ the value at row μ and column m in the multiplication matrix T_i. The following theorem states the correctness and the complexity of the algorithm.

Theorem 3.2. *Algorithm 3 is correct. More precisely, at the end of the k-th run of the loop, the matrices (T_i) are correct modulo π^k. Furthermore, each run through the loop requires $O(\delta^2\varepsilon)$ operations in K°.*

The proof uses the following observation, which is the translation to the Tate setting of the structure of the normal forms of the staircase in the classical setting.

Lemma 3.3. *If $X_i m \in B$, then $\mathrm{val}(c_{i,\mu,m}) > 0$ if $\mu \neq X_i m$. Otherwise, if $X_i m \leq \mu$, then $\mathrm{val}(c_{i,\mu,m}) > 0$.*

Proof. By definition, the column indexed by m in the multiplication matrix M_i is the vector of the coordinates of the normal form N of $X_i m$ modulo G, in the basis B. If $X_i m \in B$, then $N = X_i m$ and the result is clear. Otherwise, if $c_{i,\mu,m}\mu$ is a term of N, then $c_{i,\mu,m}\mu < X_i m$, which, by definition of the Tate term ordering, means that either $\mu < X_i m$ or $\mathrm{val}(c_{i,\mu,m}) > 0$. □

Proof of the theorem. We prove the result by induction on $k \geq 0$, and, for each value of k, by induction on $X_i m$, $i \in \{1,\dots,n\}$, $m \in B$.

The initial case $k = 0$ is empty. Let $k > 0$, $i \in \{1,\dots,n\}$ and $m \in B$, and assume by induction that we know the coefficient $c_{j,\mu',m'}$ with precision k if $X_j m' < X_i m$, and with precision $k-1$ otherwise.

If $X_i m \in B$, then there is nothing to prove, because the coefficients are 0 or 1. If $X_i m = \mathrm{LT}(g)$ for some $g \in G$, there is also nothing to prove, since all the coefficients of $M_i \cdot [m]$ are known to precision prec $\geq k$.

In the remaining case, for each $\mu \in B$, the algorithm performs the substitution

$$c_{i,\mu,m} \leftarrow \sum_{\mu' \in B} c_{j,\mu,\mu'} c_{i,\mu',m'}.$$

Since $X_j m' = m$, $m' < m$ and $X_i m' < X_i m$ and so the induction hypothesis applies. Let $\mu' \in B$. Note that $X_j \mu' \neq X_i m$: otherwise, $X_j \mu' = X_i X_j m'$ so $\mu' = X_i m'$, which cannot lie in B. If $X_j \mu' < X_i m$, then both $c_{j,\mu,\mu'}$ and $c_{i,\mu',m'}$ are known up to precision k (induction on $X_i m$), so the product is known to precision k. And if $X_j \mu' > X_i m$, then $c_{j,\mu,\mu'}$ is known up to precision $k-1$ (induction on k) and $c_{i,\mu',m'}$ is known up to precision k (induction on $X_i m$) and divisible by π (by Lemma 3.3), so the product is known up to precision k.

For the number of operations, observe that there are less than ε pairs (i, m) such that the alsorithm needs to perform the computation at line 3.11; each computation involves δ coefficients of the matrix, and for each of them, δ products in K°. □

3.2 Nonreduced Gröbner bases

An interesting feature of the algorithm above is that contrary to the usual case, it has to handle monomials which are larger than the current monomial $X_i m$. This removes the main reason for the requirement that the input Gröbner basis is reduced, and with slight modifications, it can handle any Gröbner basis as long as it is reduced modulo π. Precisely, this is achieved by replacing line 3.8 with the following.

Algorithm 3a: Update the matrices using a nonreduced basis element

```
3.8a for aμ in the support of g − LT(g) do
3.8b     if μ < X_i m then
3.8c         T_i · [m] ← T_i · [m] + a[μ]
3.8d     else
3.8e         Pick m' ∈ B such that μ = X_1^{α_1} ··· X_n^{α_n} m';
3.8f         T_i · [m] ← T_i · [m] + a T_1^{α_1} ··· T_n^{α_n} · [m'];
```

Unless the staircase is trivial, *i.e.* as long as the ideal is proper, it is always possible to find a suitable m' at line 3.8e, by picking the monomial $1 \in B$. Nonetheless, to avoid computing large powers of matrices, it is more efficient to find m' as large as possible.

It is still true that any monomial $\mu > X_i m$ appearing in the process necessarily carries a coefficient with valuation ≥ 1, and thus the loop invariant that the coefficients are known to precision k still holds.

The complexity of the computation is no longer bounded merely in terms of δ, ε and prec, but also depends on the degree of the nonreduced terms in the basis, and on the choices of the monomials m'.

3.3 Recursive algorithm

As described above, the computations can be done in increasing order of the monomials $X_i m$, ensuring that all the necessary coefficients are known with the necessary precision for the next step. Another way to proceed is by dynamic programming, computing the necessary coefficients recursively if they are not known yet.

The recursive definition, using the matrices T_i as a cache, is described in Algorithm 4, and is very similar to that described in Algorithm 3.

It can then be called, for all values of i, μ, m and $k = \text{prec}$, instead of lines 3.3–3.11 in Algorithm 3. The main difference is that the algorithm does not need to specify in which order the coefficients are computed: the recursive definition queries the missing coefficients as needed. The decision on which precision is needed depends on the valuation of the coefficient: the idea is that if a is known with precision k and has valuation v, and b is known with precision l and has valuation w, then $a \cdot b$ is known with precision $\min(k + w, l + v)$. Note that it also works if we only know a lower bound on the valuation, typically if all the digits we know are 0.

The proof that the recursive algorithm terminates is the existence of such an order, as demonstrated in Theorem 3.2. And the proof of complexity is also immediate: there are ε coefficients for which the calculation is non-trivial, and for each of them, after δ multiplications, we gain one digit of precision. The total complexity is then $O(\varepsilon\delta^2 \text{prec})$ operations in K° as in the iterative case.

The advantage of the recursive presentation is twofold. Firstly, it will allow in Section 3.4 to generalize the construction, the proof of termination, and the complexity bounds, to arbitrary log-radii.

Secondly, it offers a way to immediately improve the performance of the algorithms, on coefficient rings such as $\mathbb{Z}_p$ or $k((T))$ where fast arithmetic is available. This works by using a lazy representation of the number, that is, a representation where each number is the data of its first digits, as well as a function allowing to compute the next digit. Algorithm 4 gives us precisely such a function, and as such, the process can be viewed as a recursive definition of lazy numbers (the function definition) together with a delayed evaluation (the function call for all values).

For many coefficient rings, it is possible to do better by using the so-called relaxed, or on-line, arithmetic. Such arithmetics are available for formal power series rings [vdH97] and p-adic numbers [BvdHL11, BL12]. In that case, the cost of the computation of each new digit (of each variable) is polynomial in $\log(\text{prec})$ if we are counting base operations (in the sense of Section 2). Here, this allows us to compute the matrices with base complexity in $\tilde{O}(\varepsilon\delta^2 \text{prec})$.

Remark 3.4. For simplicity and for the complexity bounds, we only presented the procedure in the case where the Gröbner basis is reduced, but given that the recursive definition is equivalent to the loop presented in Algorithm 3, the case where the Gröbner basis is not reduced can be dealt with in exactly the same way.

3.4 General log-radii

We now consider the case of arbitrary log-radii $\mathrm{r} \in \mathbb{Q}^n$. We will prove that the algorithm presented above still works in that case, by using abstract changes of variables and base ring to justify the existence of a suitable execution order. Crucially, the algorithm works without performing those transformations, and the complexity is the same. We only need to be more careful about the handling of the precision and of the valuation.

Namely, given $\mathrm{r} \in \mathbb{Q}^n$, we will assume that the input basis G is normalized, in the sense that $0 \leq \mathrm{val}(\mathrm{LT}(g)) < 1$ for all $g \in G$. We will further require that for each $g \in G$, and for each t in

Algorithm 4: `MulMat_rec`(G, B, i, μ, m, k)

Input : G as in Algo. 3, B the staircase of G, $i \in [\![1, n]\!]$, $\mu \in B$, $m \in B$, $k \in \mathbb{Z}, k \leq \text{prec}$
Global : $(T_i) = (c_{i,\mu,m})_{\mu,m\in B})_{i\in[\![1,n]\!]}$
Output : T_i is such that $c_{i,\mu,m}$ is known to precision k

4.1 **if** $c_{i,\mu,m}$ is known to precision k in T_i **then**
4.2 **do nothing**
4.3 **else if** $k \leq 0$ **then**
4.4 $c_{i,\mu,m} \leftarrow O(1)$
4.5 **else if** $X_i m \in B$ **then**
4.6 $c_{i,\mu,m} \leftarrow 1$ **if** $\mu = X_i m$ **else** 0
4.7 **else if** $X_i m = \mathrm{LT}(g)$ for $g \in G$ **then**
4.8 $c_{i,\mu,m} \leftarrow$ the coordinate of μ in the support of $g - \mathrm{LT}(g)$
4.9 **else**
4.10 Write $m = X_j m'$ for some $m' \notin B$;
4.11 $c \leftarrow 0$;
4.12 **for** $\mu' \in B$ **do**
4.13 $v \leftarrow \mathrm{val}(c_{j,\mu,\mu'})$; $w \leftarrow \mathrm{val}(c_{i,\mu',m'})$;
4.14 `MulMat_rec`$(G, B, j, \mu, \mu', k-w)$;
4.15 `MulMat_rec`$(G, B, i, \mu', m', k-v)$;
4.16 $c \leftarrow c + c_{j,\mu,\mu'} c_{i,\mu',m'}$;
4.17 $c_{i,\mu,m} \leftarrow c + O(\pi^{k+1})$;

the support of G, t is known to precision $\text{prec} + \lfloor \mathrm{val}(t) \rfloor$, and we will ensure that we compute the matrices with similar precision by ensuring that $c_{i,\mu,m}$ is correct up to precision $k + \lfloor v_{\mu,m} \rfloor$.

Recall that $\mathrm{NF}(X_i m) = \sum_{\mu\in B} c_{i,\mu,m}\mu$ with, for all μ, $c_{i,\mu,m}\mu < X_i m$. So by definition of the Tate term ordering, $\mathrm{val}(c_{i,\mu,m}) \geq \mathrm{val}(X_i m) - \mathrm{val}(\mu) = v_{\mu,m}$, and the requirement on the precision is merely adjusting the number of digits we require beyond those we already know to be 0. The only difference is that each term is initialized with the zero digits and the precision which we already know:

Algorithm 4a: Base case with non-zero log-radii

4.3 **else if** $k \leq \lfloor \mathrm{val}(X_i m) - \mathrm{val}(\mu) \rfloor$ **then**
4.4 $c_{i,\mu,m} \leftarrow O(\pi^{\lfloor \mathrm{val}(X_i m) - \mathrm{val}(\mu) \rfloor})$;

Theorem 3.5. *Let $\mathrm{r} \in \mathbb{Q}^n$ be a system of log-radii. Algorithm 3, with input a reduced Gröbner basis of an ideal in $K\{\mathrm{X};\mathrm{r}\}^\circ$, and modified to compute the matrices using Algorithm 4, computes the multiplication matrices in $O(\varepsilon\delta^2\mathrm{prec})$ multiplications in K°.*

Proof. Let $\Gamma(G)$ be the dependency graph of the recurrence relation defined in Algorithm 4 with the modifications of Algorithm 4a: namely, $\Gamma(G)$ is a directed graph whose vertices are tuples (i, μ, m, k), and there is a directed edge $(i, \mu, m, k) \to (j, \mu', m', l)$ if and only if the computation of $c_{i,\mu,m}$ to precision k queries the coefficient $c_{j,\mu',m'}$ to precision l. Note that the vertices with no outgoing edge correspond to coefficients which are immediately known to precision prec. The recursive computation terminates if and only if the graph is cycle-free, namely, if every path through the graph eventually reaches a vertex with no outgoing edge.

In the case of trivial log-radii $\mathrm{r} = (0, \dots, 0)$, the proof of that fact is Theorem 3.2. Assume that $\mathrm{r} \in \mathbb{Q}^n$. Let D be the common denominator of the log-radii, so that $\mathrm{r} = (r_1/D, \dots, r_n/D)$. Without loss of generality, we may assume that G is minimal: elements of G which can be removed will not take part in the computation. Consider the field extension $L = K[\eta]$ with $\eta^D = \pi$, and perform the change of variables $X_i \leftarrow \eta^{r_i} Y_i$. This change of variables transforms G into a Gröbner basis G' of an ideal in $L^\circ\{\mathrm{Y}\}$. In this case, the algorithm terminates, so the graph G' is cycle-free. If G is minimal, so is G', and by [CVV19, Prop. 3.10], the elements of this basis lie in $K^\circ\{\mathrm{Y}\} \subset L^\circ\{\mathrm{Y}\}$. In particular, all throughout the algorithm, the coefficients of the matrices are in K°.

The graph $\Gamma(G)$ is isomorphic to a subgraph of $\Gamma(G')$, the inclusion being given by $(i, \mu, m, k + \lfloor v_{m,\mu} \rfloor \to (i, \mu, m, k)$. Since $\Gamma(G')$ is cycle-free, so is $\Gamma(G)$ and the algorithm terminates.

The bound on the number of operations can be obtained with a similar argument as before, or read on the graph: the complexity is bounded by 2δ times the number of vertices of $\Gamma(G)$ since computing each new vertex has a cost of 2δ operations in K (the additions and multiplications on line 4.16). Since $\Gamma(G)$ has at most $\varepsilon\delta\cdot$prec vertices, the bound $O(\varepsilon\delta^2\mathrm{prec})$ follows. □

4 CHANGE OF LOG-RADII AND ORDERING

The next step in the FGLM algorithm consists in going in the opposite direction: starting from multiplication matrices and a term ordering, we aim at reconstructing the underlying Gröbner basis.

Moreover, in our setting where we want to be able to handle in addition changes of log-radii, a preliminary step is needed. Indeed, the multiplication matrices are usually affected by a modification of the log-radii. For example, the ideal generated by $2x^2 - y^2$ and $y^3 - x^2$ in $\mathbb{Q}_2[x, y]$ has staircase $\{1, y, x, y^2, xy, xy^2\}$ (for lex) while it spans an ideal over $\mathbb{Q}_2\{x, y\}$ with staircase $\{1, y\}$ (still using lex). We study this phenomenon in full generality in Section 4.1.

A toy implementation of the algorithms of this Section is available on https://gist.github.com/TristanVaccon.

4.1 New multiplication matrices

Theoretical results. Let r and u be two n-tuples such that $r_i \geq u_i$ for all i.[2] Under this assumption the Tate algebra $K\{\mathrm{X};\mathrm{r}\}$ is included in $K\{\mathrm{X};\mathrm{u}\}$ and, given an ideal I in $K\{\mathrm{X};\mathrm{r}\}$, it makes sense to consider the ideal $J = I \cdot K\{\mathrm{X};\mathrm{u}\}$ of $K\{\mathrm{X};\mathrm{u}\}$.

In what follows, we always assume that I is 0-dimensional. The quotient $K\{\mathrm{X};\mathrm{r}\}$ is then, by definition, a finite dimensional K-vector space; we will denote it by V. Similarly, we set $W = K\{\mathrm{X};\mathrm{u}\}/J$. The inclusion $K\{\mathrm{X};\mathrm{r}\} \hookrightarrow K\{\mathrm{X};\mathrm{u}\}$ induces a K-linear mapping $\Phi : V \to W$.

In order to study Φ, we use topological arguments. We let $\|\cdot\|_\mathrm{u}$ be the norm on $K\{\mathrm{X};\mathrm{u}\}$ associated to the Gauss valuation val_u and equip $K\{\mathrm{X};\mathrm{u}\}$ with the topology associated to this norm.

Lemma 4.1. *The ideal J is the closure of I in $K\{\mathrm{X};\mathrm{u}\}$.*

[2] The results of this section can be extended without difficulty to $r_i = +\infty$, *i.e.* to $K[\mathrm{X}]$.

Proof. The polynomial ring $K[\mathrm{X}]$ is dense in $K\{\mathrm{X};\mathrm{u}\}$ for the norm $\|\cdot\|_u$. Therefore, $K\{\mathrm{X};\mathrm{r}\}$ is dense as well, implying that I is dense in J. The fact that J is closed follows from [Bo14, Chap. 2, Cor. 8]. □

The norm $\|\cdot\|_{\mathrm{u}}$ induces by restriction a norm on $K\{\mathrm{X};\mathrm{r}\}$ (which is, of course, different from the standard norm $\|\cdot\|_{\mathrm{r}}$ on this space) and a mapping $\|\cdot\|_V : V \to \mathbb{R}^+$ defined by:

$$\|x\|_V = \inf_{\hat{x}} \|\hat{x}\|_{\mathrm{u}}$$

where the infinum runs over all $\hat{x} \in K\{\mathrm{X};\mathrm{r}\}$ lifting x. In general, $\|\cdot\|_V$ is not a norm but only a semi-norm, meaning that there might exist elements $x \in V$ for which $\|x\|_V = 0$. By definition, the *kernel* of $\|\cdot\|_V$ is the set of such elements; we denote it by N. It is easily seen that N is a sub-K-vector space of V.

Proposition 4.2. *The map Φ is surjective and its kernel is N.*

Proof. We notice that $K\{\mathrm{X};\mathrm{u}\}$ is the completion of $K\{\mathrm{X};\mathrm{r}\}$ for the norm $\|\cdot\|_u$. Combining this observation with Lemma 4.1, we deduce that W appears as the completion of V with respect to the semi-norm $\|\cdot\|_V$, which is also the completion of V/N. But, since V/N is finite dimensional, it is already complete. As a conclusion, $W \simeq V/N$ and the proposition is proved. □

We now assume that we are given the multiplication matrices $T_1, \dots, T_n$ over V. We want to relate them to W, or equivalently to N. This is the content of the following proposition.

Proposition 4.3. *With the above notations, we have:*

$$N = \sum_{i=1}^{n} \mathrm{Big}_{u_i}(T_i)$$

(where we recall that the notation Big_{u_i} was defined in §2.2).

Proof. First of all, we observe that, up to replacing K by $K[\pi^{1/D}]$ for a well-chosen integer D, we can assume without loss of generality that u is in $\mathbb{Z}^n$. Replacing T_i by $\pi^{-u_i}T_i$ and r by r−u, we may further suppose that $\mathrm{u} = (0, \dots, 0)$.

Let $i \in \{1, \dots, n\}$ and let $x \in \mathrm{Big}_0(T_i)$. By definition, x is killed by $\chi_{\mathrm{Big},0,T_i}(T_i)$. In other words, if $f \in K\{\mathrm{X};\mathrm{r}\}$ is a lifting of x, the product $\chi_{\mathrm{Big},0,T_i}(X_i) \cdot f$ lies in I. Now, we claim that $\chi_{\mathrm{Big},0,T_i}(X_i)$ is invertible in $K\{\mathrm{X}\}$ because it is a product of factors of the form $a^{-1}(1 - aX_i)$ with $\mathrm{val}(a) > 0$. Consequently, f must be an element of J. By Proposition 4.2, we derive $x \in N$, which proves the inclusion $\mathrm{Big}_0(T_i) \subset N$. Since this holds for any i, the $\supset$ part of the Proposition is proved.

Set $N' = \sum_{i=1}^{n} \mathrm{Big}_0(T_i)$ and $W' = V/N'$. From what we have done so far, we deduce that the semi-norm $\|\cdot\|_V$ on V induces a semi-norm on W'. The proposition will follow if we can prove that $\|\cdot\|_V$ is indeed a norm (*i.e.* with trivial kernel) on W'. In order to do so, we consider the unit ball of W', namely:

$$D' = \left\{ x \in W' \text{ s.t. } \|x\|_V \leq 1 \right\}.$$

We want to prove that D' does not contain any K-line. For this, we remember that the unit ball of $K\{\mathrm{X}\}$ is exactly the K°-module generated by the monomials X^{i} for i varying in $\mathbb{N}^n$. Therefore, D' is the smallest K°-module stable under the T_i's and containing the image of $1 \in K\{\mathrm{X}\}$ in W'. Keeping in mind in addition that the T_i's commute pairwise, we get $D' = \mathrm{Sat}_{T_1}\mathrm{Sat}_{T_2}\cdots\mathrm{Sat}_{T_n}(L_0)$ where L_0 is the sub-K°-module of W' generated by the image of 1.

Besides, on W', all the eigenvalues of all the T_i's have nonnegative valuation since we have quotiented out all the $\mathrm{Big}_0(T_i)$'s. Consequently Lemma 2.2 applies and shows that D' is finitely generated. In particular, it contains no K-line, as wanted. □

Algorithm 5: `NewMulMat`$(T_1, \dots, T_n, v)$

Input : $T_1, \dots, T_n$ the multiplication matrices over V,
v the image of $1 \in K\{\mathrm{X};\mathrm{r}\}$ in V, $\mathrm{u} \in \mathbb{Z}^n$

Output : $U_1, \dots, U_n$ the multiplication matrices over the unit ball of W,
w the image of $1 \in K\{\mathrm{X};\mathrm{r}\}$ in W

5.1 $N \leftarrow \{0\}$;
5.2 **for** $i \in [\![1, n]\!]$ **do**
5.3 $\quad N \leftarrow N + \mathtt{Big}_0(T_i)$;
5.4 $W \leftarrow V/N$;
5.5 $L \leftarrow vK^\circ$;
5.6 **for** $i \in [\![1, n]\!]$ **do**
5.7 $\quad L \leftarrow \mathtt{Saturate}(T_i, L)$;
5.8 **return** $T_{1|L}, \dots, T_{n|L}, v \bmod N$;

Explicit computations. It is straightforward to turn the previous theoretical analysis into an actual algorithm that computes the space $W \simeq V/N$ and the multiplication matrices acting on it. In fact, for later use, it will not be enough to express these matrices in any K-basis of W, but we shall really need a K°-basis of the unit ball of W (for the norm $\|\cdot\|_V$ introduced before).

When $\mathrm{u} = (0, \dots, 0)$, Algorithm 5 does the job. In the description of this algorithm, we have implicitely assumed that all K-vector spaces and K°-modules are equipped with distinguished bases, and consequently used the same notation for a matrix and the endomophism it represents. All operations on K°-modules can be handled using Hermite normal forms as recalled in §2.2; similarly, operations on K-vectors spaces can be done using Smith normal forms, which permits to keep better numerical stability.

If δ denotes the dimension of $V = K\{\mathrm{X};\mathrm{r}\}/I$ (which is also the size of the matrices T_i's), Algorithm 5 requires at most $O^{\tilde{}}(n\delta^3)$ operations in the base field K.

4.2 Reconstruction of the Gröbner basis

The final step in the FGLM algorithm is the computation of a Gröbner basis from the datum of the multiplication matrices.

Trivial log-radii. We first address the case where $\mathrm{u} = (0, \dots, 0)$, which is covered by Algorithm 6 (page 8). This algorithm uses a routine `FGLMField` which takes as input a set of n multiplication matrices over a field (together with the vector representing the monomial 1) and a term ordering and returns the corresponding Gröbner basis. A description of such an algorithm performing this task can be found in many places in the litterature, for example in the original article by Faugère *et al.* [FGLM93].

Proposition 4.4. *Algorithm 6 is correct and runs in $O(n\delta^3)$ arithmetic operations where δ denotes the dimension of W.*

Algorithm 6: $\mathtt{GB}(U_1, \dots, U_n, w, \leq)$

Input : $U_1, \dots, U_n$ the multiplication matrices over the unit ball of W,
w the image of $1 \in K\{\mathrm{X}; \mathrm{r}\}$ in W,
$\leq$ a monomial ordering

Output : A Gröbner basis G of the ideal $J \subset K\{\mathrm{X}; \mathrm{u}\}$

$\bar{G} \leftarrow \mathtt{FGLMField}(U_1 \bmod \pi, \dots, U_n \bmod \pi, w \bmod \pi, \leq)$;
$B \leftarrow \{m \text{ monomials not divisible by any } \mathrm{LM}(g), g \in \bar{G}\}$
$M = (M_{\star,\mu})_{\star \in \mathcal{B}, \mu \in B} \leftarrow$ zero matrix;
// $\mathcal{B}$ denotes the distinguished basis of W we are working with
$M_{\star,1} \leftarrow w$;
for $\mu \in B \backslash \{1\}$ by increasing order for $\leq$ **do**
  write $\mu = X_i \mu'$ with $i \in \{1, \dots, n\}$, $\mu' \in B$;
  $M_{\star,\mu} \leftarrow U_i \cdot M_{\star,\mu'}$; // product matrix-vector
$N = (N_{\star,m})_{\star \in \mathcal{B}, m \in \mathrm{LM}(\bar{G})} \leftarrow$ zero matrix;
for $m \in \mathrm{LM}(\bar{G})$ **do**
  write $m = X_i \mu$ with $i \in \{1, \dots, n\}$, $\mu \in B$;
  $N_{\star,m} \leftarrow U_i \cdot M_{\star,\mu}$; // product matrix-vector
$Q \leftarrow M^{-1} N$;
$G \leftarrow \left(m - \sum_{\mu \in B} Q_{\mu,m} \mu\right)_{m \in \mathrm{LM}(\bar{G})}$;
return G

Proof. We recall that we assume $\mathrm{u} = (0, \dots, 0)$. Let $J = I \cdot K\{\mathrm{X}\}$ be as in Section 4.1. We recall that $W = K\{\mathrm{X}\}/J$ by definition. Let D denote the unit ball of W. From the facts that the unit ball of $K\{\mathrm{X}\}$ is $K\{\mathrm{X}\}^\circ$ and the norm on $W \simeq K\{\mathrm{X}\}/J$ is the quotient norm, we deduce that $D \simeq K\{\mathrm{X}\}^\circ / J^\circ$ with $J^\circ = J \cap K\{\mathrm{X}\}^\circ$. The reductions modulo π of the U_i's are then the multiplication matrices on the quotient $\bar{J} = J^\circ / \pi J^\circ$. The call to `FGLMField` then returns a Gröbner basis of the ideal $\bar{J}$. From Theorem 2.1.(2), we derive that the leading terms of a Gröbner basis of J are formed by the monomials in $\mathrm{LM}(\bar{G})$. It follows from this that B is the staircase of the ideal J. In particular, its cardinality is the dimension of W, showing that the matrix M is a square matrix. After the loops, the columns of M (resp. of N) contain the coordinates of the μ's (resp. the m's) in the distinguished basis $\mathcal{B}$ for μ varying in B (resp. for m varying in $\mathrm{LT}(G)$). The matrix $Q = M^{-1}N$ then contains the expression of the m's in terms of linear combination of the μ's. This shows the correctness of the algorithm.

The fact that the complexity is in $O(n\delta^3)$ arithmetic operations is easily checked. □

General log-radii. We now consider the general case where $\mathrm{u} = (u_1, \dots, u_n) \in \mathbb{Q}^n$. We take $D \in \mathbb{Z}_{>0}$ to be a common denominator of the coordinates of u (consequently $D \cdot \mathrm{u} \in \mathbb{Z}^n$). We define the field extension $L = K[\eta]$ such that $\eta^D = \pi$ and perform the change of variables $\tilde{X}_i = \eta^{Du_i} X_i$. The Tate algebra $L \otimes_K K\{\mathrm{X}; \mathrm{u}\}$ becomes isomorphic to $L\{\tilde{\mathrm{X}}\}$ and we can then apply all what precedes with $L \otimes_K K\{\mathrm{X}; \mathrm{u}\}$.

Inside $L\{\tilde{\mathrm{X}}\} \simeq L \otimes_K K\{\mathrm{X}; \mathrm{u}\}$ sits the subset $\eta^{\mathbb{Z}} K\{\mathrm{X}; \mathrm{u}\}$ consisting of series of the form $\eta^v f$ with $v \in \mathbb{Z}$ and $f \in K\{\mathrm{X}; \mathrm{u}\}$. Let I_L, J_L, V_L and W_L denote the spaces deduces by I, J, V and W respectively by extending scalars from K to L. Inside them, we can similarly define $\eta^{\mathbb{Z}} I$, $\eta^{\mathbb{Z}} J$, $\eta^{\mathbb{Z}} V$ and $\eta^{\mathbb{Z}} W$. We claim that then Algorithms 5 and 6 can be adapted so that they only have to manipulate vectors lying in these subsets. Indeed:

- the $\mathrm{Big}_0(\tilde{T}_i)$'s can be computed without passing to L because they are equal to the $\mathrm{Big}_{u_i}(T_i)$'s which are defined over K;
- similarly the quotient V/N is defined over K and then does not create any trouble;
- one checks that the Hermite normal form of a matrix whose column vectors are in $\eta^{\mathbb{Z}} V$, remains of this form;
- the column vectors of the matrices M and N of Algorithm (6) all come from monomials and so have the required shape.

Proceeding this way, we avoid the time penalty due to scalar extension from K to L and keep a complexity of $O(n\delta^3)$ arithmetic operations. At the end of the day, the output of Algorithm 6 is then a Gröbner basis G of J_L consisting of series in $\eta^{\mathbb{Z}} K\{\mathrm{X}; \mathrm{u}\}$. Still staying in the same subset, we can normalize these series so that they all have Gauss valuation 0. In this case, G is not only a Gröbner basis of J_L but also a Gröbner basis on the ideal $J_L^\circ = L \otimes_K J^\circ = J_L \cap K\{\mathrm{X}; \mathrm{u}\}^\circ$. From [CVV19, Proposition 3.10], we deduce that $G \cap K\{\mathrm{X}; \mathrm{u}\}$ remains a Gröbner basis of J° and hence of J.

Conclusion. Combining Algorithms 3 (or 4), 5 and 6 and the above discussion for covering the case of arbitrary log-radii, we finally end up with a complete FGLM algorithm as announced in Theorem 1.1 in the introduction. Plugging Algorithm 3a into the machine, we notice that our algorithm can also accept Gröbner bases which are nonreduced as soon as they are reduced modulo the maximal ideal. However, in this case, the complexity may grow up rapidly, depending on the shape of the input Gröbner basis.

REFERENCES

[BvdHL11] Berthomieu, J., van der Hoeven, J., Lecerf, G., Relaxed algorithms for p-adic numbers, Journal de Théorie des Nombres de Bordeaux, 23(3):541–577, 2011.

[BL12] Berthomieu, J., Lebreton, R., Relaxed p-adic Hensel lifting for algebraic systems, In Proceedings of ISSAC'12, pages 59-66. ACM Press, 2012.

[Bo14] Bosch, S., Lectures on Formal and Rigid Geometry, Lecture Notes in Mathematics **2105**, Springer, 2014.

[CRV16] Caruso, X., Roe, D., Vaccon T., Division and Slope Factorization of p-Adic Polynomials , in Proceedings: ISSAC 2016, Waterloo, Canada.

[CRV17] Caruso, X., Roe, D., Vaccon T., Characteristic polynomials of p-adic matrices, in Proceedings: ISSAC 2017, Kaiserslautern, Germany.

[CVV19] Caruso, X., Vaccon T., Verron T., Gröbner bases over Tate algebras, in Proceedings: ISSAC 2019, Beijing, China.

[CVV20] Caruso, X., Vaccon T., Verron T., Signature-based algorithms for Gröbner bases over Tate algebras , in Proceedings: ISSAC 2020, Kalamata, Greece.

[FGLM93] Faugère, J.-C., Gianni, P., Lazard, D., Mora, T., Efficient computation of zero-dimensional Gröbner bases by change of ordering, J. of Symbolic Computation 16 (4), 329–344, 1993

[FP04] Fresnel, J., van der Put, M., Rigid analytic geometry and its applications, Birkhäuser, 2004

[Go88] Gouvea, F., Arithmetic of p-adic Modular Forms, Lecture Notes in Mathematics **1304**, Springer-Verlag, 1988

[vdH97] van der Hoeven, J., Lazy multiplication of formal power series, in W. W. Küchlin, editor, Proc. ISSAC '97, pages 17–20, Maui, Hawaii, July 1997.

[IVY20] Ishihara, Y., Vaccon T., Yokoyama K., On FGLM Algorithms with Tropical Gröbner bases, in Proceedings: ISSAC 2020, Kalamata, Greece.

[KV04] Kaltofen, E., Villard, G., On the complexity of computing determinants, Comput. Complexity vol. 13, 2014

[KV20] Kulkarni, A., Vaccon, T., Super-linear convergence in the p-adic QR-algorithm, arxiv:2009.00129

[LS07] Le Stum Bernard, Rigid Cohomology, Cambridge tracts in mathematics **172**, Cambridge University Press, 2007

[PS73] Paterson, M.S. and Stockmeyer, L.J., On the number of nonscalar multiplications necessary to evaluate polynomials, SIAM J. Comput. 2, 60–66, 1973

[Sage] SageMath, the Sage Mathematics Software System (Version 9.2), The Sage Development Team, 2020, http://www.sagemath.org

[Ta71] Tate J., Rigid analytic spaces, Inventiones Mathematicae **12**, 1971, 257–289