Signature-based algorithms for Gröbner bases over 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.vernon@jku.at

ABSTRACT

Introduced by Tate in [Ta71], Tate algebras play a major role in the context of analytic geometry over the p-adics, where they act as a counterpart to the use of polynomial algebras in classical algebraic geometry. In [CVV19] the formalism of Gröbner bases over Tate algebras has been introduced and effectively implemented. One of the bottleneck in the algorithms was the time spent on reductions, which are significantly costlier than over polynomials. In the present article, we introduce two signature-based Gröbner bases algorithms for Tate algebras, in order to avoid many reductions. They have been implemented in SAGE-MATH. We discuss their superiority based on numerical evidences.

CCS CONCEPTS

• Computing methodologies → Algebraic algorithms;

KEYWORDS

Algorithms, Power series, Tate algebra, Gröbner bases, F5 algorithm, p-adic precision

ACM Reference Format:


1 INTRODUCTION

For several decades, many computational questions arising from geometry and arithmetic have received much attention, leading to the development of more and more efficient algorithms and softwares. A typical example is the development of the theory of Gröbner basis, which provides nowadays quite efficient tools for manipulating ideals in polynomial algebras and, eventually, algebraic varieties and schemes. At the intersection of geometry and number theory, one finds p-adic geometry and, more precisely, the notion of p-adic analytic varieties first defined by Tate in 1971 [Ta71], which plays quite an important role in many modern theories and achievements (e.g. p-adic cohomologies, p-adic modular forms).

The main algebraic objects upon which Tate’s geometry is built are Tate algebras and their ideals. In an earlier paper [CVV19], the authors started to study computational aspects related to Tate algebras: they introduce Gröbner bases in this context and design two algorithms (adapted from Buchberger’s algorithm and the F4 algorithm, respectively) for computing them.

In the classical setting, the main complexity bottleneck in Gröbner basis computations is the time spent reducing elements modulo the basis. The most costly reductions are typically reductions to 0, because they require successively eliminating all terms from the polynomial; yet their output has little value for the rest of the algorithm. Fortunately, it turns out that many such reductions can be predicted in advance (for example those coming from the obvious equality \(fg - gf = 0\)) by keeping track of some information on the module representation of elements of an ideal, called their signature. This idea was first presented in Algorithm F5 [Fa02] and led to the development of many algorithms showing different ways to define signatures, to use them or to compute them. The interested reader can look at [EF17] for an extensive survey.

The Tate setting is not an exception to the wisdom that reductions are expensive. The situation is actually even worse since reductions to 0 are theoretically the result of an infinite sequence of reduction steps converging to 0. In practice, the process actually stops because we are working at finite precision; however, the higher the precision is, the more expensive the reductions to 0 are, for no benefit. This observation motivates investigating the possibility of adding signatures to Gröbner basis algorithms for Tate series.

Our contribution. In this paper, we present two signature-based algorithms for the computation of Gröbner bases over Tate algebras. They differ in that they use different orderings on the signatures.

Our first variant, called the PoTe (position over term) algorithm, is directly adapted from the G2V algorithm [GGV10]. It adopts an incremental point of view and uses the so-called cover criterion [GVW16] to detect reductions to 0. A key difficulty in the Tate setting is that the usual way to handle signatures assumes the constant term 1 is the smallest one. However, the assumption fails in the Tate setting. We solve this issue by importing ideas from [L+18] in which the case of local algebras is addressed.

In the classical setting, incremental algorithms have the disadvantage of sometimes computing larger Gröbner bases for intermediate ideals, only to discard them later on. In order to mitigate
this misfeature, the F5 algorithm uses a signature ordering taking into account the degree of the polynomials first, in order to process lower-degree elements first. In the Tate setting, the degree no longer makes sense and a better measure of progression of the algorithms is the valuation. Nonetheless, similarly to the classical setting, an incremental algorithm could perform intermediate computations to high valuation and just discard them later on. The second algorithm we will present, called the VaPoTe (valuation over position over term) algorithm, uses an analogous idea to that of F5 to mitigate this problem.

Organization of the article. In §2, we recall the basic definitions and properties of Tate algebras and Gröbner basis over them, together with the principles of the G2V algorithm. The two next sections are devoted to the PoTe and the VaPoTe algorithms respectively: they are presented and their correctness and termination are proved. Finally, implementation, benchmarks and possible future improvements, are discussed in §5.

Notations. Throughout this article, we fix a positive integer $n$ and use the short notation $X$ for $(X_1, \ldots, X_n)$. Given $i = (i_1, \ldots, i_n) \in \mathbb{N}^n$, we shall write $X^i$ for $X_1^{i_1} \cdots X_n^{i_n}$.

2 INGREDIENTS

In this section, we present the two main ingredients we are going to mix together later on. They are, first, the G2V [GGV10] and GVW [GVW16] signature-based algorithms, and, second, the Tate algebras and the theory of Gröbner bases over them as developed in [CVV19].

2.1 The G2V algorithm

In what follows, we present the G2V algorithm which was designed by Gao, Guan and Volny IV in [GGV10] as an incremental variant of the classical F5 algorithm. Our presentation includes the cover criterion which was formulated later on in [GVW16] by Gao, Volny IV and Wang. The incremental point of view is needed for the application we will discuss in §4. Moreover we believe that it has two extra advantages: first, it leads to simplified notations and, more importantly, it shows clearly where intermediate inter-reductions are possible.

Let $k$ be a field and $k[X]$ denote the ring of polynomials over $k$ with indeterminates $X$. We endow $k[X]$ with a fixed monomial order $\preceq_{\text{st}}$. Let $I_0$ be an ideal in $k[X]$. Let $G_0$ be a Gröbner basis of $I_0$ with respect to $\preceq_{\text{st}}$. Let $f \in k[X]$. We aim at computing a GB of the ideal $I = I_0 + \langle f \rangle$. Let $M \subset k[X] \times k[X]$ be the $k[X]$-sub-module defined by the $(u,v)$ such that $uf - v \in I_0$. The leading monomial of $u$ is the signature of $(u,v)$.

Definition 2.1 (Regular reduction). Let $p_1 = (u_1, v_1)$ and $p_2 = (u_2, v_2)$ be two elements in $M$. We say that $p_1$ is top-reducible by $p_2$ if

1. either $v_2 = 0$ and $LM(u_2)$ divides $LM(u_1)$,
2. or $v_2 \neq 0$, $LM(u_2)$ divides $LM(v_1)$ and:

$$\frac{LM(u_1)}{LM(u_2)} \cdot LM(v_2) \leq LM(u_1).$$

The corresponding top-reduction is

$$p = p_1 - t p_2 = (u_1 - tu_2, v_1 - tv_2)$$

where $t = \frac{LM(u_1)}{LM(u_2)}$ is the first case and $t = \frac{LM(v_1)}{LM(v_2)}$ in the second case. This top-reduction is called regular when $LM(u_1) > tLM(u_2)$, that is when the signature of the reduced pair $p$ agrees with that of $p_1$; it is called super otherwise.

Definition 2.2 (Strong Gröbner bases). A finite subset $G$ of $M$ is called a strong Gröbner basis (SGB, for short) of $M$ if any nonzero $(u,v) \in M$ is top-reducible by some element of $G$.

The G2V strategy derives the computation of a Gröbner basis through the computation of an SGB. They are related through the following proposition.

Proposition 2.3. Suppose that $G = \{(u_1, v_1), \ldots, (u_r, v_r)\}$ is an SGB of $M$. Then:

1. $\{u \text{ s.t. } (u, 0) \in G\}$ is a Gröbner basis of $I_0$.
2. $\{v \text{ s.t. } (u, v) \in G \text{ for some } u\}$ is a Gröbner basis of $I$.

To compute an SGB, we rely on J-pairs instead of S-polynomials.

Definition 2.4 (J-pair). Let $p_1 = (u_1, v_1)$ and $p_2 = (u_2, v_2)$ be two elements in $M$ such that $u_1 v_2 \neq 0$. Let $t = \operatorname{lc}(LM(v_2), LM(u_2))$ and set $t_i = \frac{t}{LM(u_i)}$ for $i \in \{1, 2\}$. Then:

1. if $LM(t_i u_1) < LM(t_i u_2)$, the J-pair of $(p_1, p_2)$ is $t_2 p_2$.
2. if $LM(t_i u_1) > LM(t_i u_2)$, the J-pair of $(p_1, p_2)$ is $t_1 p_1$.
3. if $LM(t_i u_1) = LM(t_i u_2)$, the J-pair of $(p_1, p_2)$ is not defined.

Definition 2.5 (Cover). We say that $p = (u, v)$ is covered by $G \subset M$ if there is a pair $(u_i, v_i) \in G$ such that $LM(u_i)$ divides $LM(u)$ and:

$$\frac{LM(u_1)}{LM(u)} \cdot LM(v_i) < LM(v).$$

Theorem 2.6 (Cover Theorem). Let $G$ be a finite subset of $M$ such that:

1. $G$ contains $(1, f)$,
2. the set $\{g \in k[X] : (0, g) \in G\}$ forms a Gröbner basis of $I_0$.

Then $G$ is an SGB of $M$ if every J-pair of $G$ is covered by $G$.

This theorem leads naturally to the G2V algorithm (see [GGV10, Fig. 1]) which is rephased hereafter in Algorithm 1 (page 4). We underline that, in Algorithm 1, the SGB does not entirely appear. Indeed, we remark that one can always work with pairs $(LM(u), v)$ in place of $(u,v)$, reducing then drastically the memory occupation and the complexity. The algorithm maintains two lists $G$ and $S$ which are related to the SGB in construction as follows: $G \cup (S \times \{0\})$ is equal to the set of all $(LM(u), v)$ when $(u,v)$ runs over the SGB. The criterion coming from the cover theorem is implemented on lines 10 and 11: the first (resp. the second) statement checks if $(u,v)$ is covered by an element of $G$ (resp. an element of $S \times \{0\}$).

Syzygies. The G2V algorithm does not give direct access to the module of syzygies of the ideal. However, it does give access to a GB of $(I_0 : f)$ (see Proposition 2.3), from which one can recover partial information about the syzygies, as shown below.

Definition 2.7. Given $f_1, \ldots, f_m \in k[X]$, we define

$$\text{Syz}(f_1, \ldots, f_m) = \left\{ (a_1, \ldots, a_m) \in k[X]^m \text{ s.t. } \sum_{i=1}^m a_i f_i = 0 \right\}.$$
LEMMA 2.8. Let $f_1, \ldots, f_m$ generating $I_0$ and let $u_1, \ldots, u_s$ generating $(I_0 : f)$. For $i \in \{1, \ldots, s\}$, we write

$$-u_if = a_i1f_1 + \cdots + a_imf_m$$

and define $z_i = (a_{i1}, \ldots, a_{im}, u_i) \in \text{Syz}(f_1, \ldots, f_m)$. Then

$$\text{Syz}(f_1, \ldots, f_m, f) = (\text{Syz}(f_1, \ldots, f_m) \times \{0\}) + \langle z_1, \ldots, z_s \rangle.$$

Proof. Let $(a_1, \ldots, a_m, u) \in \text{Syz}(f_1, \ldots, f_m)$ and we can write $u = \sum_{i=1}^s b_iu_i$. Then the syzygy $(a_1, \ldots, a_m, u) - \sum_{i=1}^s b_iz_i$ has its last coordinate equal to 0 and thus belongs to $(\text{Syz}(f_1, \ldots, f_m) \times \{0\})$, which is enough to conclude. \qed

2.2 Tate algebras

Definitions. We fix a field $K$ equipped with a discrete valuation $\nu : K \to \mathbb{Z} \sqcup \{+\infty\}$, normalized by $\nu(K^*) = \mathbb{Z}$. We assume that $K$ is complete with respect to the distance defined by $\nu$. Let $K^\nu$ be the subring of $K$ consisting of elements of nonnegative valuation $\nu$ and $\pi$ be a uniformizer of $K$, that is an element of valuation 1. We set $K = K^\nu/\pi K^\nu$. The Tate algebra $K\{X\}$ is defined by:

$$K\{X\} := \left\{ \sum_{i \in \mathbb{N}^n} a_i X^i \mid a_i \in K \text{ and } \nu(a_i) \rightarrow +\infty \right\}$$

Series in $K\{X\}$ have a natural analytic interpretation: they are analytic functions on the closed unit disc in $K^\nu$. We recall that $K\{X\}$ is equipped with the so-called Gauss valuation defined by:

$$\nu \left( \sum_{i \in \mathbb{N}^n} a_i X^i \right) = \min_{i \in \mathbb{N}^n} \nu(a_i).$$

Series with nonnegative valuation form a subring $K\{X\}^\nu$ of $K\{X\}$. The reduction modulo $\pi$ defines a surjective homomorphism of rings $K\{X\}^\nu \rightarrow K\{X\}$. Theorem 2.8.

Terms and monomials. By definition, an integral Tate term is an expression of the form $aX^i$ with $a \in K^\nu$, $a \neq 0$ and $i \in \mathbb{N}^n$. Integral Tate terms form a monoid, denoted by $T\{X\}^\nu$, which is abstractly isomorphic to $(K^\nu(\{0\}) \times \mathbb{N}^n)^\nu$. We say that two Tate terms $aX^i$ and $bX^j$ are equivalent when $\nu(a) = \nu(b)$ and $i = j$. A Tate term modulo equivalence defines a quotient $T\{X\}^\nu / T\{X\}$, which is isomorphic to $\mathbb{N} \times \mathbb{N}^n$. The image in $T\{X\}^\nu$ of a term $t \in T\{X\}^\nu$ is called the monomial of $t$ and is denoted by $\text{mon}(t)$. We fix a monomial order $\leq_m$ on $\mathbb{N}^n$ and order $T\{X\}^\nu \cong \mathbb{N} \times \mathbb{N}^n$ lexicographically by block with respect to the reverse natural ordering on the first factor $\mathbb{N}$ and the order $\leq_m$ on $\mathbb{N}^n$. Pulling back this order along the morphism $\nu$, we obtain a preorder of $T\{X\}^\nu$ that we shall continue to denote by $\leq$. The leading term of a Tate series $f = \sum_{i \in \mathbb{N}^n} a_i X^i \in K\{X\}^\nu$ is defined by:

$$LT(f) = \max_{i \in \mathbb{N}^n} \nu(a_i) X^i \in T\{X\}^\nu.$$

We observe that the $a_iX^i$'s are pairwise nonequivalent in $T\{X\}^\nu$, showing that there is no ambiguity in the definition of $LT(f)$. The leading monomial of $f$ is by definition $LM(f) = \text{mon}(LT(f))$. Gröbner bases. The previous inputs allow us to define the notion of Grobner basis for an ideal of $K\{X\}^\nu$.

Definition 2.9. Let $I$ be an ideal of $K\{X\}^\nu$. A family $(g_1, \ldots, g_s) \in I^s$ is a Gröbner basis (in short, GB) of $I$ if, for all $f \in I$, there exists $i \in \{1, \ldots, s\}$ such that $LM(g_i)$ divides $LM(f)$.

A classical argument shows that any GB of an ideal $I$ generates $I$. The following theorem is proved in [CVV19, Theorem 2.19].

Theorem 2.10. Every ideal of $K\{X\}^\nu$ admits a GB.

The explicit computation of such a GB is of course a central question. It was addressed in [CVV19], in which the authors describe a Buchberger algorithm and an F4 algorithm for this task. The aim of the present article is to improve on these results by introducing signatures in this framework and eventually design F5-like algorithms for the computation of GB over Tate algebras.

Important remark. For the simplicity of exposition, we chose to restrict ourselves to the Tate algebra $K\{X\}$ and not consider the variants $K\{X; r\}$ allowing for more general radii of convergence. However, using the techniques developed in [CVV19] (paragraph General log-radii of §3.2), all the results we will obtain in this article can be more generally extended to $K\{X; r\}$.

3 POSITION OVER TERM

The goal of this section is to adapt the G2V algorithm to the setting of Tate algebras. Although all definitions, statements and algorithms are formally absolutely parallel to the classical setting, proofs in the framework of Tate algebras are more subtle, due to the fact the orderings on Tate terms are not well-founded but only topologically well-founded. In order to accommodate this weaker property, we import ideas from [L+18] where the case of local rings is considered.

3.1 The PoTe algorithm

We fix a monomial order $\leq_m$ on $\mathbb{N}^n$ and write $\leq$ for the term order on $T\{X\}^\nu$ it induces. We consider an ideal $I_0$ in $K\{X\}^\nu$ along with a GB $G_0$ of $I_0$. Let $f \in K\{X\}^\nu$. We are interested in computing a GB of $I = I_0 + \langle f \rangle$. Mimicking what we have recalled in §2.1, we introduce the $K\{X\}^\nu$-sub-module $M \subset K\{X\}^\nu \times K\{X\}^\nu$ consisting of pairs $(u, v)$ such that $uf - v \in I_0$. The definitions of regular reduction (Definition 2.1), strong Gröbner bases (Definition 2.2), J-pair (Definition 2.4) and cover (Definition 2.5) extend verbatim to the context of Tate algebras, with the precaution that the leading monomial is now computed with respect to the order $\leq$ as explained in §2.2.

Proposition 3.1. Suppose that $G = \{(u_1, v_1), \ldots, (u_s, v_s)\}$ is an SGB of $M$. Then:

(1) $\{u \circ t. u, 0 \in G\}$ is a Gröbner basis of $(I_0 : f)$.

(2) $\{v \circ t. (u, v) \in G \text{ for some } u\}$ is a Gröbner basis of $I$.

Proof. Let $G$ be an SGB of $M$. Let $h \in (I_0 : f)$. Then $hf \in I_0$ and $h, 0 \in M$. By definition, since $G$ is an SGB of $M$, there exists $(u, 0) \in G$ such that $LM(u)$ divides $LM(h)$. This implies the first statement of the proposition.

Let now $h \in I$. If $LM(h) \in I_0$, there exists a pair $(0, h') \in M$ with $LM(h) = LM(h')$. This pair is divisible by some $(0, v) \in G$, proving that $LM(v)$ divides $LM(h') = LM(h)$ in this case. We now suppose that $LM(h) \notin I_0$. This assumption implies that any $a \in K\{X\}^\nu$ with $(a, h) \in M$ (i.e. $ah - h \in I_0$) must satisfy $LM(a) \leq LM(h)/LM(f)$. We can then choose a series $a \in K\{X\}^\nu$ such that $(a, h) \in M$ and $LM(a)$ is minimal for this property. Moreover, since $G$ is an SGB, the pair $(a, h)$ has to be top-reducible by
some \((u, v) \in G\). If \(v \neq 0\), we deduce that \(LM(v)\) divides \(LM(h)\). Otherwise, letting \(t = LT(a)/LT(u)\), we obtain \((a - tu, h) \in M\) with \(LM(a - tu) < LM(a)\), contradicting the minimality of \(LM(a)\). As a conclusion, we have proved that \(LM(v)\) divides \(LM(h)\) in all cases, showing that the set \(\{v \in G \mid (u, v) \in G\} \) is a GB of \(I\). \(\square\)

**Theorem 3.2 (Cover Theorem).** Let \(G\) be a finite subset of \(M\) such that:

- \(G\) contains \((1, f)\);
- the set \(\{g \in K[X] : (0, g) \in G\}\) forms a Gröbner basis of \(I_0\).

Then \(G\) is an SGB of \(M\) if every J-pair of \(G\) is covered by \(G\).

The proof of Theorem 3.2 is presented in §3.2 below. Before this, let us observe that Theorem 3.2 readily shows that the G2V algorithm (see Algorithm 1, page 4) extends verbatim to Tate algebras. The resulting algorithm is called the PoTe\(^1\) algorithm. The correctness of the PoTe algorithm is clear thanks to Theorem 3.2. Its termination is not a priori guaranteed because the call to regular_reduce may enter an infinite loop (see [CVV19, §3.1]). However, if we assume that all regular reductions terminate (which is guaranteed in practice by working at finite precision), the PoTe algorithm terminates as well thanks to the Noetherianity of \(K[X]\).

### 3.2 Proof of the cover theorem

Throughout this subsection, we consider a finite set \(G\) satisfying the assumptions of Theorem 3.2.

\(^1\)PoTe means "Position over Term".

---

**Algorithm 1: G2V algorithm**

**Input:** \(f_1, \ldots, f_m\) in \(K[X]\) (resp. \(K[X]^0\))

**Output:** a GB of the ideal generated by the \(f_i\)’s

\[ Q \leftarrow (f_1, \ldots, f_m) \]

\[ GBasis \leftarrow \emptyset \]

\[ \text{for } f \in Q \text{ do} \]

\[ G \leftarrow \{(0, g) : g \in GBasis\} \cup \{(1, f)\} \]

\[ S \leftarrow \{LM(g) : g \in GBasis\} \]

\[ B \leftarrow \{J\text{-pair}((1, f), (0, g)) : g \in GBasis\} \]

\[ \text{while } B \neq \emptyset \text{ do} \]

\[ \text{pop } (u, v) \text{ from } B, \text{ with smallest } u \]

\[ \text{if } (u, v) \text{ is covered by } G \text{ then continue} \]

\[ \text{if } u \text{ is divisible by some } s \in S \text{ then continue} \]

\[ v_0 \leftarrow \text{regular_reduce } (u, v, G) \]

\[ \text{if } v_0 = 0 \text{ then} \]

\[ \text{add } u \text{ to } S \]

\[ \text{else} \]

\[ \text{for } (s, g) \in G \text{ do} \]

\[ \text{if } J\text{-pair}((u, v_0), (s, g)) \text{ is defined then} \]

\[ \text{add } J\text{-pair}((u, v_0), (s, g)) \text{ to } B \]

\[ \text{add } (u, v_0) \text{ to } G \]

\[ GBasis \leftarrow \{v : (u, v) \in G\} \]

\[ \text{return } GBasis \]

---

**Algorithm 2: VaPoTe algorithm**

**Input:** \(f_1, \ldots, f_m\) in \(K[X]^0\)

**Output:** a GB of the ideal generated by the \(f_i\)’s

\[ Q \leftarrow (f_1, \ldots, f_m) \]

\[ GBasis \leftarrow \emptyset \]

\[ \text{while } Q \neq \emptyset \text{ do} \]

\[ \text{pop } f \text{ from } Q, \text{ with smallest valuation} \]

\[ G \leftarrow \{(0, g) : g \in GBasis\} \cup \{(1, f)\} \]

\[ S \leftarrow \{LM(g) : g \in GBasis\} \]

\[ B \leftarrow \{J\text{-pair}((1, f), (0, g)) : g \in GBasis\} \]

\[ \text{while } B \neq \emptyset \text{ do} \]

\[ \text{pop } (u, v) \text{ from } B, \text{ with smallest } u \]

\[ \text{if } (u, v) \text{ is covered by } G \text{ then continue} \]

\[ \text{if } u \text{ is divisible by some } s \in S \text{ then continue} \]

\[ v_0 \leftarrow \text{regular_reduce } (u, v, G) \]

\[ \text{if } \text{val}(v_0) > \text{val}(f) \text{ then} \]

\[ \text{add } u \text{ to } S; \text{ add } v_0 \text{ to } Q \]

\[ \text{else} \]

\[ \text{for } (s, g) \in G \text{ do} \]

\[ \text{if } J\text{-pair}((u, v_0), (s, g)) \text{ is defined then} \]

\[ \text{add } J\text{-pair}((u, v_0), (s, g)) \text{ to } B \]

\[ \text{add } (u, v_0) \text{ to } G \]

\[ GBasis \leftarrow \{v : (u, v) \in G\} \]

\[ \text{return } GBasis \]

---

We first assume that \(G\) is an SGB of \(M\). Let \(p_1, p_2 \in G\) and write \(p_i = (u_i, v_i)\) for \(i \in \{1, 2\}\). We set \(t_1 = \text{lcm}(LM(v_1), LM(v_2)) \in T[X]^0\) and \(t_1 = t/LM(v_1)\). If \(LM(t_1u_1) = LM(t_2u_2)\), the J-pair of \((p_1, p_2)\) is not defined and there is nothing to prove. Otherwise, if \(i\) (resp. \(j\)) is the index for which \(LM(t_iu_i)\) is maximal (resp. \(LM(t_ju_j)\) is minimal), the J-pair of \((p_1, p_2)\) is \(t_i p_i\), which is regularly top-reducible by \(p_j\). Continuing to apply regular top-reductions by elements of \(G\) as long as possible, we reach a pair \((u_0, v_0) \in M\) which is no longer regularly top-reducible by any element of \(G\) and for which \(LM(v_0) = LM(t_iu_i)\) and \(LM(v_0) < LM(t_ju_j)\). Since \(G\) is an SGB of \(M\), \((u_0, v_0)\) must be super top-reducible by some pair \((u, v) \in G\). By definition of super top-reducibility, \(LM(u)\) divides \(LM(u_0) = LM(t_iu_i)\) and \(LM(v) \cdot LM(v_0) = LM(v_0) \cdot LM(u)\). This shows that \(LM(v) \cdot LM(u_0) < LM(v_0) \cdot LM(u)\) and then that \((u, v)\) covers \(t_i p_i\).

We now focus on the converse and assume that each J-pair of \(G\) is covered by \(G\). We define:

\[ W = \{ (u, v) \in M : \text{top-reducible by no pair of } G \} \]

and assume by contradiction that \(W\) is not empty.

**Lemma 3.3.** The set \(W\) does not contain any pair of the form \((u, v)\) with \(u = 0\) or \(LM(v) \in LM(I_0)\).

**Proof.** By our assumptions, if \(LM(v) \in LM(I_0)\), \(v\) is reducible by some \(g\) with \((0, g) \in G\). In particular, \((u, v)\) is top-reducible by \((0, g)\) and cannot be in \(W\). If \(u = 0\), then \(v \in I_0\) and we are reduced to the previous case. \(\square\)
LEMMA 3.4. Let \( p_0 = (u_0, v_0) \in W \). Then there exists a pair \( p_1 = (u_1, v_1) \in G \) such that \( LT(u_1) \) divides \( LT(u_0) \), say \( LT(u_0) = t_1 LT(u_1) \) and \( t_1 LT(v_1) \) is minimal for this property.

Furthermore, if \( t_1 p_1 \) is not regularly top-reducible by \( G \).

Proof. We have already noticed that \( u_0 \neq 0 \). Since \( 1, f \in G \), there exists a pair in \( G \) satisfying the first condition. Since \( G \) is finite, there exists one that further satisfies the minimality condition.

We assume by contradiction that \( t_1 p_1 \) is regularly top-reducible by \( G \). Consider \( p_2 = (u_2, v_2) \in G \) be a regular reducer of \( t_1 p_1 \), in particular there exists a term \( t_2 \) such that \( t_2 LT(v_2) = t_1 LT(v_1) \), and \( t_2 LT(u_2) < t_1 LT(u_1) \). The J-pair of \( p_1 \) and \( p_2 \) is then defined and equals to \( \tau \cdot (u_1, v_1) \) with \( \tau \) dividing \( t_1 \). Write \( t_1 = t_1' \) for some term \( t_1' \). By hypothesis, this J-pair is covered, so there exists \( P = (U, V) \in G \) and a term \( \theta \) such that \( \theta \cdot LT(U) = \tau \cdot LT(u_1) \) and \( \theta \cdot LT(V) < \tau \cdot LT(v_1) \). As a consequence:

\[
\begin{aligned}
t_1' \theta \cdot LT(U) &= t_1 \cdot LT(u_1) = LT(u_0) \\
t_1' \theta \cdot LT(V) &< \tau \cdot LT(v_1).
\end{aligned}
\]

So \( t_1' P \) contradicts the minimality of \( p_1 \). \( \Box \)

Let \( v \) be the minimal valuation of a series \( v \) for which \( (u, v) \in W \). We make the following additional assumption: \( v < +\infty \). In other words, we assume that \( W \) contains at least one element of the form \( (u, v) \) with \( v \neq 0 \). We set:

\[
W_1 = \{ (u, v) \in W \text{ s.t. } \text{val}(LM(u)) = v \}.
\]

LEMMA 3.5. The set \( L = \{ LM(u) : (u, v) \in W_1 \} \) admits a minimal element.

Proof. We assume by contradiction that \( L \) does not have a minimal element. Thus, we can construct a sequence \((u_k, v_k)_{k \geq 1}\) with values in \( W_1 \) such that \( LM(u_k) \) is strictly decreasing. As a consequence, in the Tate topology, \( u_k \) converges to 0. Hence, for \( k \) large enough, \( \text{val}(u_k f) > v = \text{val}(v_k) \). From \( W_1 \subseteq M \), we get \( v_k = u_k f \in I_0 \) and \( LM(v_k) = LM(u_k) \cdot LM(f) \in LM(I_0) \). By Lemma 3.3, this is a contradiction. \( \Box \)

Let \( W_2 \) be the subset of \( W_1 \) consisting of pairs \( (u, v) \) for which \( LM(u) \) is minimal. Note that by Lemma 3.3, this minimal value is nonzero.

LEMMA 3.6. For any \((u_1, v_1), (u_2, v_2) \in W_2, LM(u_1) = LM(u_2)\).

Proof. Let \((u_1, v_1), (u_2, v_2) \in W_2\), and assume that the leading terms are not equivalent, that is \( LM(u_1) \neq LM(u_2) \). Without loss of generality, we can assume that \( LM(u_1) > LM(u_2) \). By construction of \( W_2 \), \( \text{LT}(u_1) = \text{LT}(u_2) \), that is \( LT(u_1) = a LT(u_2) \) for some \( a \in K \), \( \text{val}(a) = 0 \). Since \( u_1 \) and \( u_2 \) are nonzero, we can write \( u_1 = LT(u_1) + r_1 \) and \( u_2 = LT(u_2) + r_2 \). Eliminating the leading terms, we obtain a new element \((u', v') = (r_1 - ar_2, u_1 - ar_2)\). By assumption, \( LM(u') = LM(v') \), and \( LM(u') < LM(u_1) \). Observe that \((u', v')\) cannot be top-reduced by \( G \) as otherwise, \((u_1, v_1)\) would also be top-reducible by \( G \). Hence \((u', v') \in W_1\), contradicting the minimality of \( LM(u_1) \). \( \Box \)

Let now \( p_0 = (u_0, v_0) \in W_2 \). From Lemma 3.4, there exists \( p_1 = (u_1, v_1) \in G \) and a term \( t \) such that \( LT(tu_1) = LT(u_0) \) and \( tp_1 \) is not regular top-reducible by \( G \). We define

\[
p_s = (u_s, v_s) = p_s - tp_1 = (u_0, v_0) - t(u_1, v_1).
\]

We remark that \( LM(u_s) < LM(u_0) \). Moreover \( LM(u_0) \neq LM(tu_1) \) since otherwise \( p_0 \) would be top-reducible by \( p_1 \), contradicting the fact that \( p_0 \in W \).

We first examine the case where \( LM(u_0) < LM(tu_1) \). It implies that \( LM(u_s) = LM(tu_1) \). Let us prove first that \( p_s \notin W \). We argue by contradiction. From \( p_s \notin W \), we would derive \( \text{val}(v_s) \geq v = \text{val}(v_0) \) and then \( \text{val}(v_s) = \text{val}(v_0) \) since the inequality in the other direction holds by assumption. We conclude by noticing that \( LM(u_s) < LM(u_0) \) contradicts the minimality of \( LM(u_0) \). So \( p_s \notin W \), i.e. \( p_s \) is top-reducible by \( G \). Let \( p_2 = (u_2, v_2) \in G \) top-reducing \( p_s \). If \( v_2 = 0 \), then \( LM(u_2) \) divides \( LM(u_s) \). Besides, the pair:

\[
p_s' = (u_s', v_s') = (u_s - LT(u_0) - LT(u_2), v_s)
\]

satisfies \( LM(u_s') < LM(u_s) \) and thus cannot be in \( W \) either. We iterate this process until we can only find a reductor \( q = (U, V) \in G \) with \( V \neq 0 \). Let \( t_2 = LM(u_s)/LM(V) \).

Then:

\[
\begin{aligned}
t_2 LM(V) &= LM(u_s) = LM(tu_1), \\
t_2 LM(U) &\leq LM(u_s) < LM(tu_1) \quad \text{if } U \neq 0
\end{aligned}
\]

Thus \( q \) regularly top-reduces \( tp_1 \), which contradicts Lemma 3.4.

Let us now move to the case where \( LM(u_0) > LM(tu_1) \). Then \( LM(u_0) = LM(u_0) \). Combining this with \( LM(u_s) < LM(u_0) \), we deduce \( p_s \notin W \), i.e. \( p_s \) is top-reducible by \( G \). As in the previous case, we construct \( q = (U, V) \in G \) with \( V \neq 0 \) and a term \( t_2 \) such that:

\[
\begin{aligned}
t_2 LM(V) &= LM(u_s) = LM(u_0), \\
t_2 LM(U) &\leq LM(u_s) < LM(u_0) \quad \text{if } U \neq 0
\end{aligned}
\]

Thus \( q \) regularly top-reduces \( p_0 \), which contradicts \( p_0 \in W \).

As a conclusion, in both cases, we have reached a contradiction. This ensures that \( v = +\infty \). In particular, \( W \) contains an element \( p_0 \) of the form \( (u_0, 0) \). Let \( p_1 = (u_1, v_1) \in G \) be given by Lemma 3.4. If \( v_1 = 0 \), this pair would be a reductor of \((u_0, 0) \in W \), which is a contradiction. So \( v_1 \neq 0 \). Set \( t = \frac{LT(u_0)}{LT(u_1)} \). Let:

\[
p_s = (u_s, v_s) = (u_0, 0) - t(u_1, v_1) = (u_0 - tu_1, -v_1)
\]

Then \( LM(u_s) < LM(u_0) \) and \( LM(v_s) = tLM(v_1) \). From \( v_1 \neq 0 \), we deduce \( p_s \notin W \). So \( p_s \) is top-reducible by \( p_2 = (u_2, v_2) \in G \), meaning that there exists a term \( t_2 \) such that \( t_2 LM(u_2) = LM(u_s) = tLM(v_1) \) and \( t_2 LM(u_2) \leq LM(u_s) < tLM(u_1) \). So \( p_2 \) is a regular top-reducer of \( tp_1 \), which contradicts Lemma 3.4.

Finally, we conclude that \( W \) is empty. By construction, \( G \) is an SGB of \( M \).

4 VALUATION OVER POSITION OVER TERM

In this section, we design a variant of the PoTe algorithm in which, roughly speaking, signatures are first ordered by increasing valuations.
4.1 The VaPoTe algorithm

The VaPoTe algorithm is Algorithm 2. It is striking to observe that it looks formally very similar to the PoTe Algorithm (Algorithm 1) as they only differ on lines 3–4 and, more importantly, on lines 13–14. However, these slight changes may have significant consequences on the order in which the inputs are processed, implying possibly important differences in the behaviour of the algorithms.

The VaPoTe algorithm has a couple of interesting features. First, if we stop the execution of the algorithm at the moment when we first reach a series $f$ of valuation greater than $N$ on line 4, the value of $\text{GBasis}$ is a GB of the image of $I = \langle f_1, \ldots, f_m \rangle$ in $K[X]^{\nu}/\pi^N K[X]^{\nu}$. In other words, the VaPoTe algorithm can be used to compute GB of ideals of $K[X]^{\nu}/\pi^N K[X]^{\nu}$ (for our modified order) as well.

Secondly, Algorithm 2 remains correct if the reduction on line 12 is interrupted as soon as the valuation rises. The property allows for delaying some reductions, which might be expensive at one time but cheaper later (because more reductors are available). It also has a theoretical interest because the reduction process may a priori hang forever (if we are working at infinite precision); interrupting it prematurely removes this defect and leads to more satisfying termination results.

4.2 Proof of correctness and termination

We introduce some notations. For a series $f \in K[X]^\nu$, we write $v(f) = \nu - \text{val}(f)$ (which has valuation 0 by construction) and define $\rho(f)$ as the image of $v(f)$ in $K[X]^\nu/\pi K[X]^\nu \approx k[X]$. More generally if $A$ is a subset of $K[X]^\nu$, we define $v(A)$ and $\rho(A)$ accordingly.

We consider $f_1, \ldots, f_m \in K[X]^\nu$ and write $I$ for the ideal generated by $f_1, \ldots, f_m$. For any integer $N$, we set $I_N = I \cap (\pi^N K[X]^\nu)$. Clearly $I_{N+1} \subseteq I_N$ for all $N$. Let $I_N$ be the image of $\pi^{-N} I_N$ in $k[X]$; we have a canonical isomorphism $I_N \approx I_N/I_{N+1}$. Besides, the morphism $I_N \to I_{N+1}$, $f \mapsto \pi f$ induces an inclusion $I_N \to I_{N+1}$. Hence, the $I_N$ form a nondecreasing sequence of ideals of $k[X]$.

We define $Q_{\text{all}}$ as the set of all series that are popped from $Q$ on line 13 during the execution of Algorithm 2. Since the algorithm terminates when $Q$ is empty, $Q_{\text{all}}$ is also the set of all series that has been in $Q$ at some moment. For an integer $N$, we further define:

\[
Q_N = \{ f \in Q_{\text{all}} \text{ s.t. } \text{val}(f) = N \},
\]

\[
Q_{\leq N} = \{ f \in Q_{\text{all}} \text{ s.t. } \text{val}(f) \leq N \},
\]

\[
Q_{> N} = \{ f \in Q_{\text{all}} \text{ s.t. } \text{val}(f) > N \}.
\]

Let also $\tau_N$ be the first time we enter in the while loop on line 3 with $Q \subset \pi^N K[X]^\nu$. If this event never occurs, $\tau_N$ is defined as the time the algorithm exits the main while loop. We finally let $\text{GBasis}$ be the value of the variable $\text{GBasis}$ at the checkpoint $\tau_N$.

Lemmas 4.1. Between the checkpoints $\tau_N$ and $\tau_{N+1}$:
1. the elements popped from $Q$ are exactly those of $Q_N$, and
2. the “reduction modulo $\pi^{N+1}$” of the VaPoTe algorithm behaves like the G2V algorithm, with input polynomials $\rho(Q_N)$ and initial value of $\text{GBasis}$ set to $\rho(\text{GBasis}_N)$.

\vspace{1cm}

\textbf{Proof.} We observe that, after the time $\tau_N$, only elements with valuation at least $N+1$ are added to $Q$. The first statement then follows from the fact that the elements of $Q$ has popped by increasing valuation. The second statement is a consequence of (1) together with the fact that all $f$ and $v$ manipulated by Algorithm 2 between the times $\tau_N$ and $\tau_{N+1}$ have valuation $N$. \hfill $\Box$

Since the G2V algorithm terminates for polynomials over a field, Lemma 4.1 ensures that each checkpoint $\tau_N$ is reached in finite time if the call to regular reduce does not hang forever. This latter property holds when we are working at finite precision and is also guaranteed if we interrupt the reduction as soon as the valuation raises.

We are now going to relate the ideals $I_N$ with the sets $Q_N, Q_{\leq N}$ and $Q_{> N}$. For this, we introduce the syzygies between the elements of $\rho(Q_{\leq N})$. More precisely, we set:

\[
S_N = \{ (af)_{f \in Q_{\leq N}} \text{ s.t. } \sum_{f \in Q_{\leq N}} a_f v(f) \equiv 0 \pmod{\pi} \},
\]

and let $S_N$ be the image of $S_N$ under the projection $K[X]^\nu \to k[X]$; in other words, $S_N$ is the module of syzygies of the set $\rho(Q_{\leq N})$, i.e. $S_N = \text{Syg}(\rho(Q_{\leq N}))$ with the notation of Definition 2.7. We also define the linear mapping:

\[
\varphi_N : (K[X]^\nu)^{Q_{\leq N}} \to K[X]^\nu \cup \bigcup_{f \in Q_{\leq N}} a_f v(f).
\]

By definition, $\varphi_N$ takes its values in the ideal generated by $\nu(Q_{\leq N})$ and $\varphi_N(S_N) \subset \pi K[X]^\nu$.

\textbf{Proposition 4.2.} For any integer $N$, the following holds:

(a) The family $\rho(\text{GBasis}_{N+1})$ is a GB of $I_N$.

(b) $\varphi_N(S_N) \subset (\pi v(Q_{\leq N}), \pi^{-N} Q_{> N})$.

(c) $I_{N+1} = (\pi^{N+1} v(Q_{\leq N+1}), Q_{> N+1})$.

(d) $I_{N+1} = (\rho(Q_{\leq N+1}))$.

\textbf{Proof.} When $N < 0$, we have $S_N = 0$, $I_{N+1} = I$ and $I_N = 0$, so that the proposition is obvious. We now consider a nonnegative integer $N$ and assume that the proposition holds for $N-1$. By the induction hypothesis, we know that $\rho(\text{GBasis}_N)$ is a GB of $I_{N-1}$. It then follows from Lemma 4.1 that $\rho(\text{GBasis}_{N+1})$ is a GB of the ideal generated by $I_{N-1}$ and $\rho(Q_N)$, which is equal to $I_N$ by the induction hypothesis. The assertion (a) is then proved.

Between the checkpoints $\tau_N$ and $\tau_{N+1}$, each signature $u$ added to $S$ on line 14 corresponds to a family $(af)_{f \in Q_{\leq N}}$ for which the sum $\sum_f a_f f$ equals the element $v_0$ added to $Q$ on the same line. Rescaling the $a_f$’s, we cook up an element $z \in S_N$ with the property that $\varphi_N(z) = \pi^{-N} v_0$. Let $Z \subset S_N$ be the set of those elements. From Proposition 2.3 and Lemma 2.8, we derive that $S_N$ is generated by $S_{N-1}$ (viewed as a submodule of $S_N$ by filling new coordinates with zeroes) and $Z$. Thus:

\[
\varphi_N(S_N) = \varphi_{N-1}(S_{N-1}) + \langle \varphi_N(Z), \pi v(Q_{\leq N}) \rangle \subset \varphi_{N-1}(S_{N-1}) + \langle \pi^{-N} Q_{N-1}, \pi v(Q_{\leq N}) \rangle.
\]

The assertion (b) now follows from the induction hypothesis, once we have observed that $Q_{N-1} = \pi^N v(Q_N) \cup Q_{> N}$.

\vspace{1cm}

\textsuperscript{VaPoTe means “Valuation over Position over Term”}
Let us now prove (c). Let \( h \in I_{N+1} \). Then \( h \in I_N \) and we can use the induction hypothesis to write:

\[
    h = \pi^N \sum_{f \in Q_N} a_f \nu(f) + \sum_{g \in Q_N} b_g g
\]

for some \( a_f, b_g \in k[X] \). Reducing modulo \( \pi^{N+1} \), we find that the family \((a_f)_{f \in Q_N}\) belongs to \( S_N \). From (b), we deduce that:

\[
    \sum_{f \in Q_N} a_f \nu(f) \in \{ \nu(v(Q_N)), \pi^{-N}Q_{N} \}.
\]

Hence \( h \in \langle \pi^{N+1}v(Q_N), Q_{N+1} \rangle \) and we conclude by noticing that \( Q_{N+1} = \pi^{N+1}v(Q_{N+1}) \cup (Q_{N+1}) \).

Finally, (d) follows from (c) by dividing by \( \pi^{N+1} \) and reducing modulo \( \pi \). \( \square \)

Termination. Since \( k[X] \) is noetherian, the sequence of ideals \((I_N)\) is eventually constant. This implies that \( G\text{Basis} \) cannot grow indefinitely; in other words, the final value of \( G\text{Basis} \) is reached in finite time. However, the reader should be careful that this does not mean that Algorithm 2 terminates. Indeed, once the final value of \( G\text{Basis} \) has been computed, one still has to check that the remaining series in \( Q \) reduce to zero; this is achieved by performing divisions and can hang forever if we are working at infinite precision. Nevertheless, this misfeature seems very difficult to avoid since, when working at infinite precision, the input series contain themselves an infinite number of coefficients and any modification on one of them could have a strong influence on the final result.

Correctness. Let \( G \) be the output of Algorithm 2, that is the limit of the ultimately constant sequence \((G\text{Basis}_N)\). For a positive integer \( N \), we define:

\[
    G_{\leq N} = \{ f \in G \text{ s.t. } \nu(f) \leq N \}.
\]

Since only elements of valuation at least \( N+1 \) are added to \( G\text{Basis} \) after the checkpoint \( \tau_{N+1} \), we deduce that \( G_{\leq N} = G\text{Basis}_{SN+1} \). Hence, by Proposition 4.2, \( \rho(G_{\leq N}) \) is a GB of \( I_N \) for all \( N \geq 0 \). We are going to show that this sole property implies that \( G \) is indeed a GB of \( I \). For this, we consider \( f \in I \). We write \( N = \nu(f) \), so that \( \rho(f) \) is the image in \( [X] / \pi^{-N} f \). Moreover, we know that \( L(f)(\rho(f)) \) is divisible by \( L(f)(\rho(g)) \) for some \( g \in G_{\leq N} \), i.e. there exists \( i \in \mathbb{N}^n \) such that \( L(f)(\rho(f)) = \sum_{j} X^i \cdot L(f)(\rho(g)) \). This readily implies that:

\[
    L(f) = \pi^{N-\nu(g)} \cdot X^i \cdot L(g)
\]

showing that \( L(g) \) divides \( L(f) \) in \( \mathbb{T}[X]^n \) given that \( \nu(g) \leq N \). We have then proved that the leading monomial of any element of \( I \) is divisible by some \( L(g) \) with \( g \in G \), i.e. that \( G \) is a GB of \( I \).

5 IMPLEMENTATION

We have implemented both the PoTe and VaPoTe algorithms in SageMath\(^3\). Our implementation includes the following optimization: at the end of the loop (i.e. after line 20), we minimize and reduce the current GB in construction. This operation is allowed since all signatures are discarded after each iteration of the loop. Similarly, we reduce each new series \( f \) popped from \( Q \) on line 4 before proceeding it. These ideas were explored in the algorithm

\( ^3 \) https://trac.sagemath.org/ticket/28777

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Buchberger</th>
<th>PoTe</th>
<th>VaPoTe</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p = 5 )</td>
<td>( \ell = 5 ), prec = 12</td>
<td>87.9</td>
<td>72.2</td>
</tr>
<tr>
<td>( p = 11 )</td>
<td>( \ell = 5 ), prec = 12</td>
<td>321</td>
<td>30.5</td>
</tr>
<tr>
<td>( p = 57637 )</td>
<td>( \ell = 5 ), prec = 12</td>
<td>83.2</td>
<td>13.3</td>
</tr>
<tr>
<td>( p = 7 )</td>
<td>( \ell = 7 ), prec = 9</td>
<td>62.3</td>
<td>45.3</td>
</tr>
<tr>
<td>( p = 11 )</td>
<td>( \ell = 7 ), prec = 9</td>
<td>168</td>
<td>36.0</td>
</tr>
</tbody>
</table>

Table 1: Timings for the computation of GBs related to the torsion of points of Tate curves (all times in seconds)

F5-C [EP10] and, as mentioned before, were one of the main motivations for adopting an incremental point of view.

Our implementation is also able to compute GB of ideals in \( K[X] \). For this, we simply use a reduction (for no extra cost) to the case of \( K[X]^n \) (see [CVV19, Proposition 2.23]). We also make monic the signatures in \( S \) after each iteration of the main loop; in the PoTe algorithm, this renormalization gives a stronger cover criterion and thus improves the performances.

As mentioned in Section 4.1, Algorithm 2 remains correct if the reductions are interrupted as soon as the valuation rises. This can be done in the reduction step before processing the next \( f \), before adding elements to the SGB, as well as in the inter-reduction step. Delaying reductions could be interesting, for instance, if the input ideal is saturated: indeed, in this case, the algorithm never considers elements with positive valuation and delayed reductions do not need to be done afterwards. On the other hand, performing more reductions earlier leads to shorter reducers and potentially faster reductions later. In practice, in our current implementation, we have observed all possible scenarios: interrupting the reductions can make the computation faster, slower, or not make any significant difference.

5.1 Some timings

Numerous experimentations on various random inputs show that the VaPoTe algorithm performs slightly better than the PoTe algorithm on average. Besides, both PoTe and VaPoTe algorithms usually perform much better than Buchberger algorithm, although we observed important variations depending on the input system.

As mentioned in the introduction, Tate algebras are the building blocks of \( p \)-adic geometry. One can then cook up interesting systems associated to meaningful geometrical situations. As a basic example, let us look at torsion points on elliptic curves.

We recall briefly that (a certain class of) elliptic curves over \( \mathbb{Q}_p \) are uniformized by the Tate curve (see [Ta95]), which can be seen as the curve defined over \( K(q) \) by the explicit equation \( y^2 = x^3 + a_4 x + a_6 \) with:

\[
    a_4 = 5 \sum_{n=0}^{\infty} n^3 \frac{(pq)^n}{1 - (pq)^n}, \quad a_6 = 5 \sum_{n=0}^{\infty} 7n^5 + 5n^3 \frac{(pq)^n}{12 - (pq)^n}.
\]

Given an auxiliary prime number \( \ell \), we consider the \( \ell \)-th division polynomial \( \Phi_{\ell}(x, q) \in K(q)^n[x] \) associated to the Weierstrass form of the above equation. By definition, its roots are the abscissas of \( \ell \)-torsion points of the Tate curve. We now fix \( p \) and \( \ell \) and consider the system in 3 variables \( \Phi_{\ell}(x, q_1) = \Phi_{\ell}(x, q_2) = 0 \). Its solutions parametrize the pairs of elliptic curves sharing a common
\(\ell\)-torsion point. Computing a GB of it then provides information about torsion points on \(p\)-adic elliptic curves.

Table 1 shows the timings obtained for computing a GB of the above systems for different values of \(p\), \(\ell\) and different precisions. We clearly see on these examples than both PoTe and VaPoTe overperform the Buchberger algorithm.

### 5.2 Towards further improvements

**Faster reductions.** Observing how our algorithms behave, one immediately notices that reductions are very slow. It is not that surprising since our reduction algorithm is currently very naive. For this reason, we believe that several structural improvements are quite possible. An idea in this direction would be to store a well-chosen representative sample of reductions and reuse them later on. Typically, we could cache the reductions of all terms of the form \(x_1^{a_1} \cdots x_n^{a_n}\) (with respect to the current GB in construction) and use them to emulate a fast exponentation algorithm in the quotient ring \(K[X]/I(GB)\).

Another attractive idea for accelerating reduction is to incorporate Mora’s reduction algorithm [Mo82, MRW17] in our framework. Let us recall that Mora’s algorithm is a special method for reducing terms with respect to local or mixed orders (i.e. orders for which there exist terms \(t < 1\)), avoiding infinite loops in the reduction process. In our framework, infinite loops of reductions cannot arise since the computations are truncated at a given precision. Nevertheless, we believe that Mora’s algorithm can still be used to short-circuit some reductions.

The situation for Tate terms is actually significantly simpler than that of general local orders. Indeed, Mora’s reduction algorithm roughly amounts to add \(\pi r\) to our list of reductors each time we encounter a remainder \(r\) to our list of reductors each time we encounter a remainder \(r\) (including \(f\) itself) in the reduction process. We believe that this optimization, if it is carefully implemented, could already have some impact on the performances. Besides, observing that the equality \(f = r + \pi q f\) also reads \(f = (1 + \pi q)^{-1} r\), we realize that Mora reduction of a Tate series are somehow related to its Weierstrass decomposition. Moreover, at least in the univariate case, it is well known that Weierstrass decompositions can be efficiently computed using a well-suited Newton iteration. It could be interesting to figure out whether this strategy extends to multivariate series and, more generally, to the computation of arbitrary Mora reductions.

**Using overconvergence properties.** In a different direction, we would like to underline that the orderings we are working with are by design block orders (comparing first the valuation). However, in the classical setting, we all know that graded orders often lead to much more efficient algorithms. Unfortunately, in the setting of this article, the very first definition of a Tate series already forces us to give the priority to the valuation in the comparison of terms; otherwise, the leading term would not be defined in general.

Nonetheless, we emphasize that if graded orders does not exist over \(K[X]\), they do exist over some subrings. Precisely, recall that, given a tuple \(r = (r_1, \ldots, r_n)\), we have defined:

\[
K(X; r) := \left\{ \sum_{i \in I} a_i x^i \mid s.t. \ a_i \in K \text{ and } \text{val}(a_i) - r_i \xrightarrow{|i| \to +\infty} +\infty \right\}
\]

where \(r_i\) denotes the scalar product of the vectors \(r\) and \(i\). When the \(r_i\)'s are all nonnegative, \(K[X; r]\) embeds naturally into \(K[X]\); precisely, elements in \(K[X; r]\) are those series that overconverges over the polydisk of polyradius \(\{|x|^{-r_1}, \ldots, |x|^{-r_n}\}\). Moreover, the algebra \(K[X; r]\) is equipped with the valuation \(\text{val}\) defined by:

\[
\text{val} \left( \sum_{i \in I} a_i x^i \right) = \min_{i \in I} \text{val}(a_i) - r_i.
\]

This valuation defines a new term ordering \(\preceq_r\). We observe that, from the point of view of \(K(X)\), it really looks like a graded order: the quantity \(\text{val}(f)\) plays the role of (the opposite of) a “total degree” which mixes the contribution of the valuation and that of the classical degree.

In light of the above remarks, we formulate the following question. Suppose that we are given an ideal \(I \subseteq K[X]^n\) (say, of dimension 0) generated by some series \(f_1, \ldots, f_m\). If we have the promise that the \(f_i\)'s all overconverge, i.e. all lie in \(K[X; r]\) for a given \(r\), can we imagine an algorithm that computes a GB of \(I\) taking advantage of the term ordering \(\preceq_r\)? As an extreme case, if we have the promise that all the \(f_i\)'s are polynomials (that is \(r_i = +\infty\) for all \(i\)), can one use this assumption to accelerate the computation of a GB of \(I\)?

**REFERENCES**


[I-L18] Liu Dong, Wang Dingkang, Xiao Fanghui, Zhou Jie Extending the GVW Algo- rithm to Local Ring, Proceedings of 46th International Symposium on Symbolic and Algebraic Computation, ISSAC’18, New York, USA


[Ta71] Tate John, Rigid analytic spaces, Inventions Mathematicae 12, 1971, 257–289