Maple Transactions

The Multivariate Power Series Package in Maple 2024

JUAN PABLO GONZÁLEZ TROCHEZ, The University of Western Ontario, Canada
ALI ASADI, Xanadu, Canada
ALEXANDER BRANDT, Dalhousie University, Canada
ERIK POSTMA, Maplesoft, Canada

Recommended Reference Format:
Juan Pablo González Trochez, Ali Asadi, Alexander Brandt, and Erik Postma. 2024. The Multivariate Power Series Package in Maple 2024. 1, 1 (June 2024), 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Overview

While symbolic computation is the realm of exact methods, this field was able to develop solutions to approximate problems, which, in turn, can be used to provide approximate answers to problems that are either intractable or too expensive to solve exactly. A well-known example is the so-called symbolic Newton iteration method for approximating the solutions of algebraic equations, see Chapter 9 in the landmark textbook *Modern Computer Algebra* [18].

At the heart of these exact, but approximate, methods is the manipulation of formal multivariate power series. Power series form an active research area of symbolic computation, since the early days of that discipline [8]. Power series are available in various computer algebra systems: Mathemagix [17] Maple [9], Mathematica [19], and SageMath [14], to name a few.

Considering that a power series has potentially an infinite number of terms naturally brings computational challenges. For that reason, software implementation often restrict power series to being either univariate or truncated. By truncated, we mean reduced modulo the power of a monomial ideal.

A "truncated" implementation, while simple, may be unsatisfactory in practice. For instance, modern algorithms for polynomial system solving make an intensive use of modular methods based on Hensel lifting. In those lifting procedures, appropriate degrees of truncation may not be known a priori, thus leading to truncated power series being used in a non-optimal manner.

The limitations of an implementation based on truncated power series are overcome by using instead the technique of *lazy evaluation*, aka *call-by-need*. With this paradigm, a power series is represented as a procedure which, given a particular (total) degree, produces and memorizes the terms of that degree, if they have not been computed yet. The usefulness of lazy evaluation in computer algebra has been studied for a few decades: see the work of Karczmarczuk [7], discussing different mathematical objects with an infinite length, as well as the work of Burge and Watt in [6]. See also the landmark paper *Relax, but don't be too lazy* by van der Hoeven [16], improving the paradigm of lazy evaluation for univariate power series.

Section 2 of the present paper highlights the relations between the algebraic foundation of multivariate formal power series and the algorithmic scheme of lazy evaluation.

Authors' addresses: Juan Pablo González Trochez, The University of Western Ontario, 1151 Richmond St, London, Canada, jgonza55@uwo.ca; Ali Asadi, Xanadu, 777 Bay Street, Toronto, Canada, epostma@maplesoft.com; Alexander Brandt, Dalhousie University, 6050 University Avenue, Halifax, Canada, ABrandt@dal.ca; Erik Postma, Maplesoft, 615 Kumpf Dr, Waterloo, Canada, epostma@maplesoft.com.

https://doi.org/10.1145/nnnnnnn.nnnnnnn

A second implementation challenge arises when implementing Laurent and Puiseux series. While arithmetic operations (addition, multiplication, inversion) have natural and straightforward algorithms for multivariate formal power series, the state of affairs changes dramatically with multivariate Laurent and Puiseux series. In fact, arithmetic operations on Laurent and Puiseux require the manipulations of polyhedral cones that are trivial in the univariate case and research problems in the multivariate one, see [10, 15], as well as Section 3.

The `MultivariatePowerSeries` library in *Maple*, to which this paper is dedicated, provides formal, Laurent and Puiseux series in several variables. The implementation of those series is based on the paradigm of lazy evaluation. Not only arithmetic operations (addition, multiplication, inversion) are supported, but univariate polynomials over multivariate series are provided by this library. One of the main motivations is to factor such polynomials in a number of ways, using either Weierstrass Preparation Theorem, Hensel Lemma, Puiseux Theorem and the Extended Hensel Construction. Section 4 covers these features.

The `MultivariatePowerSeries` library was introduced in *Maple* in 2021, see [3]. Its initial version was an adaptation to the *Maple* language of an implementation [4, 5] realized in the BPAS library [2]. In 2022 and 2023, the `MultivariatePowerSeries` library was enhanced with Laurent and Puiseux series, respectively, see [15]. In 2024, the `MultivariatePowerSeries` library was enhanced with the Extended Hensel Construction, invented by T. Sasaki and F. Kako, see [13].

2 The algebraic foundation of lazy evaluation for multivariate power series

The goal of this section is to show how the `MultivariatePowerSeries` handles the arithmetic operations of addition, multiplication, inversion and composition, with the machinery of lazy evaluation. We refer to [5] for the concepts that we use without recalling their definitions.

Let $\mathbb{K}$ be a field and $X_1, \ldots, X_n$ be independent indeterminates. We denote by $\mathbb{A}$ the ring $\mathbb{K}[[X_1, \ldots, X_n]]$ of multivariate formal power series in $X_1, \ldots, X_n$ with coefficients in $\mathbb{K}$.

For an integer $k \geq 0$ and $f = \sum_e a_e X^e \in \mathbb{K}[[X_1, \ldots, X_n]]$, the *homogeneous part* of degree k of f is $f_{(k)} = \sum_{|e|=k} a_e X^e$.

For $g, h \in \mathbb{K}[[X_1, \ldots, X_n]]$ their sum s and their product p are given by:

$$s_{(k)} = g_{(k)} + h_{(k)} \text{ and } p_{(k)} = \sum_{i=0}^{k} g_{(i)} h_{(k-i)}, \tag{1}$$

for all integer $k \geq 0$.

Let $\mathcal{M} = \langle X_1, \ldots, X_n \rangle$ be the maximal ideal of $\mathbb{A}$. For $d \geq 0$, the ideal $\mathcal{M}^d$ is generated by the monomials of degree d. We have:

$$\mathcal{M}^{d+1} \subseteq \mathcal{M}^d \text{ and } \bigcap_{k \in \mathbb{N}} \mathcal{M}^k = \langle 0 \rangle. \tag{2}$$

Such a *filtration* yields a topology, the *Krull topology*, where the neighbourhoods of a power series f are of the form $f + \mathcal{M}^d$.

Let $(f_n)_{n \in \mathbb{N}}$ be a sequence of elements of $\mathbb{A}$ and let $f \in \mathbb{A}$. The sequence $(f_n)_{n \in \mathbb{N}}$ *converges* to f if for all $k \in \mathbb{N}$ there exists $N \in \mathbb{N}$ such that for all $n \in \mathbb{N}$ we have:

$$n \geq N \Rightarrow f - f_n \in \mathcal{M}^k, \tag{3}$$

Therefore, a bivariate function

$$F : \mathbb{A} \times \mathbb{A} \longmapsto \mathbb{A} \tag{4}$$

is *continuous* at (p, q) if for every $d \in \mathbb{N}$ we can find $b, c \in \mathbb{N}$ such that

$$F(p + \mathcal{M}^b, q + \mathcal{M}^c) - F(p, q) \subseteq \mathcal{M}^d. \tag{5}$$

Continuous functions are those which can be implemented by lazy evaluation; This is clearly the case for addition, multiplication. Similarly, one can prove that inversion of formal power series is continuous in Krull topology.

We turn our intention to the composition of power series, starting with the substitution of non-units (that is, elements of $\mathcal{M}$) into power series (that is, elements of $\mathbb{A}$). This substitution is well-defined, but can it be done via lazy evaluation?

For simplicity, we describe the univariate case and refer to [12] for the multivariate one. The input is $a := \sum_i a_i X^i$ and $b := \sum_j b_j X^j$. We want $a|_{X=b}$ By definition, we have:

$$a|_{X=b} = \sum_i a_i \, X^i|_{X=\sum_j b_j X^j} = \sum_i a_i \left(\sum_j b_j X^j \right)^i .$$

By the multinomial formula, we have:

$$a|_{X=b} = \sum_i a_i \left(\sum_{\underline{m} \in M_i} \left(\binom{i}{\underline{m}} \prod \left(b_j X^j\right)^{m_j} \right) \right)$$

where M_i is the set of all infinite non-negative integer sequences $(m_1, m_2, \ldots,)$ with finitely many nonzero entries and whose sum is i. Note the multinomial coefficients.

Up to elementary expansions, we also have:

$$a|_{X=b} = \sum_i a_i \left(\sum_{\underline{m} \in M_i} \binom{i}{\underline{m}} \left(\prod {b_j}^{m_j} \right) X^{\sum_j j m_j} \right)$$

By grouping terms of equal degree in X, we obtain:

$$a|_{X=b} = \sum_i \left(\sum_{\underline{m} \in M_i} a_{|\underline{m}|} \binom{|\underline{m}|}{\underline{m}} \left(\prod {b_j}^{m_j} \right) \right) X^i$$

where M_i is **now** the set of all infinite non-negative integer sequences $(m_1, m_2, \ldots,)$ with finitely many nonzero entries and such that $\sum_j j m_j = i$. Because $b_0 = 0$, we can start numbering such a sequence $\underline{m}$ at m_1 and we have $|\underline{m}| \leq \sum_j j m_j = i$. Therefore, only finitely many coefficients of a and b contribute to each coefficient of $a|_{X=b}$. Consequently, this process is continuous in the Krull topology.

Finally, we consider the substitution of units into power series. Consider $a = b = \sum_{i=0}^{\infty} \frac{X^i}{i!}$, which we traditionally view as e^X. We expect that $a|_{X=b}$ evaluates to the power series for e^{e^X}, which is

$$\sum_{i=0}^{\infty} \frac{e B_n}{n!} X^n = e + eX + eX^2 + (5/6) e X^3 + \cdots ,$$

Because of the factor of e, substituting units into power series cannot be continuous in the Krull topology. There is a work-around, however. If two multivariate power series a, b have *analytic expressions* f_a, f_b and we want $a|_{X_n=b}$, then:

(1) we compute $f := f_a|_{X_n=f_b}$, and
(2) we obtain the coefficient of $X_1^{m_1} \cdots X_n^{m_n}$ as

$$\frac{1}{m_1! \cdots m_n!} \left. \frac{\partial^{|m|} f}{\partial {X_1}^{m_1} \cdots \partial {X_n}^{m_n}} \right|_{X_1 = \cdots = X_n = 0}$$

To obtain an efficient implementation, see the details in [12].

```
> a := GeometricSeries([x, y]):
> GetAnalyticExpression(a);
```

$$\frac{1}{1-x-y} \tag{7}$$

```
> b := 1/PowerSeries(3 + 2·x + y);
```

$$b := \left[\text{PowerSeries of } \frac{1}{3+2x+y} : \frac{1}{3} + \ldots\right] \tag{8}$$

```
> e := PowerSeries(d -> (x^d/d!), analytic=exp(x));
```

$$e := \left[\text{PowerSeries of } e^{x} : 1 + \ldots\right] \tag{9}$$

```
> f := UnivariatePolynomialOverPowerSeries([a, b, e], z):
> Truncate(f, 3);
```

$$\left(1+x+\frac{1}{2}x^2+\frac{1}{6}x^3\right)z^2+\left(\frac{1}{3}-\frac{1}{9}y-\frac{2}{9}x+\frac{1}{27}y^2+\frac{4}{27}xy+\frac{4}{27}x^2-\frac{1}{81}y^3-\frac{2}{27}xy^2-\frac{4}{27}x^2y-\frac{8}{81}x^3\right)z+x^3 \\ +3x^2y+3xy^2+y^3+x^2+2xy+y^2+x+y+1 \tag{10}$$

```
> GetAnalyticExpression(f);
```

$$\frac{1}{1-x-y}+\frac{z}{3+2x+y}+e^{x}z^2 \tag{11}$$

Fig. 1. `MultivariatePowerSeries` session showing the use of analytic expressions of power series.

Figure 1 illustrates how analytic expressions of power series, and univariate polynomials over power series, are computed in the `MultivariatePowerSeries` library. These expressions are involved by various commands. We just saw the case of substitution. Another scenario is that of inversion for Laurent and Puiseux to be discussed in the next section.

3 Laurent and Puiseux series

We focus on Laurent series since the case of Puiseux series is essentially similar, as explained in [15]. However, at the end of this section we illustrates the use of multivariate Puiseux series with the `MultivariatePowerSeries` library.

Recall that $\mathbb{K}$ is a field. The sequences $\mathbf{x} = x_1, \ldots, x_p$ and $\mathbf{u} = u_1, \ldots, u_m$ are ordered indeterminates with $m \geq p$. By definition, a multivariate formal Laurent series look like:

$$f(\mathbf{x}) := \Sigma_{\mathbf{k}\in\mathbb{Z}^p}\, a_{\mathbf{k}}\mathbf{x}^{\mathbf{k}},$$

where the $a_{\mathbf{k}}$ are elements of $\mathbb{K}$, and $\mathbf{u}^{\mathbf{k}}$ is a notation for $u_1^{k_1}\cdots u_p^{k_p}$ where $k_1, \ldots, k_p$ are integers.

Let $C \subseteq \mathbb{R}^p$ be a cone. C is said to be *line-free* if for every $\mathbf{v} \in C \setminus \{\mathbf{0}\}$, we have $-\mathbf{v} \notin C$. All cones here are line-free, polyhedral and generated by integer vectors. The set of the **Laurent series** $f(\mathbf{x}) \in \mathbb{K}((\mathbf{x}))$ with $\mathrm{supp}(f(\mathbf{x})) \subseteq C$ is an integral domain denoted by $\mathbb{K}_C[[\mathbf{x}]]$, where:

$$\mathrm{supp}(f(\mathbf{x})) := \{\mathbf{k} \in \mathbb{Z}^p \mid a_{\mathbf{k}} \neq 0\}.$$

Note that, there exists $g(\mathbf{x}) \in \mathbb{K}_C[[\mathbf{x}]]$ with $f(\mathbf{x})g(\mathbf{x}) = 1$, if and only if $a_{\mathbf{0}} \neq 0$.

Let $\preceq$ be an **additive order** in $\mathbb{Z}^p$. Thus, for all $\mathbf{i}, \mathbf{j}, \mathbf{k} \in \mathbb{Z}^p$, we have:

$$\mathbf{i} \preceq \mathbf{j} \implies \mathbf{i}+\mathbf{k} \preceq \mathbf{j}+\mathbf{k}.$$

let $\mathcal{C}$ be the set of all cones $C \subseteq \mathbb{R}^p$ which are **compatible** with $\preceq$. Thus, for every $C \in \mathcal{C}$, if for all $\mathbf{k} \in C \cap \mathbb{Z}^p$ we have $\mathbf{0} \preceq \mathbf{k}$. Define:

$$\mathbb{K}_{\preceq}[[\mathbf{x}]] := \cup_{C\in\mathcal{C}}\mathbb{K}_C[[\mathbf{x}]] \quad \text{and} \quad \mathbb{K}_{\preceq}((\mathbf{x})) := \cup_{\mathbf{e}\in\mathbb{Z}^p}\mathbf{x}^{\mathbf{e}}\mathbb{K}_{\preceq}[[\mathbf{x}]].$$

Then, $\mathbb{K}_{\preceq}[[\mathbf{x}]]$ is a ring and $\mathbb{K}_{\preceq}((\mathbf{x}))$ is a field.

If you are puzzled by the factor $\mathbf{x}^{\mathbf{e}}$ think that the inverse of $x^{-1}+1+x+x^2+\cdots$ is the inverse of $x^{-1}(1+x+x^2+x^3+\cdots)$ that is $\frac{x}{1-x}$.

The `MultivariatePowerSeries` library implements $\mathbb{K}_{\preceq}((\mathbf{x}))$, where $\preceq$ is $<_{glex}$. Recall that $<_{glex}$ first compares **total degrees** before using reverse lexicographic order as tie-breaker. We explain our encoding of $\mathbb{K}_{\preceq}((\mathbf{x}))$.

Let $g \in \mathbb{K}[[\mathbf{u}]]$ be a multivariate power series, let $\mathbf{e} \in \mathbb{Z}^p$ be a point, and $\mathbf{R} := \{\mathbf{r}_1, \ldots, \mathbf{r}_m\} \subset \mathbb{Z}^p$ be a set of grevlex non-negative rays. Then,

$$f = \mathbf{x}^{\mathbf{e}} g(\mathbf{x}^{\mathbf{r}_1}, \ldots, \mathbf{x}^{\mathbf{r}_m}),$$

is a *Laurent series object*, which belongs to $\mathbf{x}^{\mathbf{e}}\mathbb{K}_C[[\mathbf{x}]]$, where C is the cone generated by $\mathbf{R}$. Our implementation encodes every multivariate Laurent series as a Laurent series object, LSO for short, that is, a quintuple $(\mathbf{x}, \mathbf{u}, \mathbf{e}, \mathbf{R}, g)$. As an example, consider $f := x^{-4}y^5 \sum_{i=0}^{\infty} x^{2i}y^{-i}$. To encode f as an LSO, one can choose:

$$\mathbf{x} = [x, y], \mathbf{u} = [u, v], \mathbf{R} = [[1, 0], [1, -1]], \mathbf{e} = [x = -4, y = 5]$$

and g = `Inverse(PowerSeries(1+uv))`.

We turn our attention to the addition and multiplication of LSOs. Let $C_1, C_2 \subseteq \mathbb{Z}^p$ be generated by grevlex non-negative rays, $\mathbf{R}_1 := \{\mathbf{r}'_1, \ldots, \mathbf{r}'_m\} \subset \mathbb{Z}^p$ and $\mathbf{R}_2 := \{\mathbf{r}''_1, \ldots, \mathbf{r}''_m\} \subset \mathbb{Z}^p$, with $m \geq p$. Consider two Laurent series in $\mathbb{K}_{\preceq}((\mathbf{x}))$, namely:

$$f_1 = \mathbf{x}^{\mathbf{e}_1} g_1(\mathbf{x}^{\mathbf{R}_1}) \text{ and } f_2 = \mathbf{x}^{\mathbf{e}_2} g_2(\mathbf{x}^{\mathbf{R}_2}),$$

with $g_1, g_2 \in \mathbb{K}[[\mathbf{u}]]$ and $\mathbf{e}_1, \mathbf{e}_2 \in \mathbb{Z}^p$. Then, we have:

$$f_1 f_2 = \mathbf{x}^{\mathbf{e}_1+\mathbf{e}_2} \left(g_1(\mathbf{x}^{\mathbf{R}_1}) g_2(\mathbf{x}^{\mathbf{R}_2}) \right).$$

Assume $\mathbf{e} = \mathbf{e}_1$ is the grevlex-minimum of $\mathbf{e}_1$ and $\mathbf{e}_2$. Then, we have:

$$f_1 + f_2 = \mathbf{x}^{\mathbf{e}} \left(g_1(\mathbf{x}^{\mathbf{R}_1}) + \mathbf{x}^{\mathbf{e}_2-\mathbf{e}} g_2(\mathbf{x}^{\mathbf{R}_2}) \right).$$

To make $f_1 f_2$ (resp. $f_1 + f_2$) an LSO object, we need to find a cone containing $\mathrm{supp}(f_1 f_2)$ (resp. $\mathrm{supp}(f_1 + f_2)$). We developed an algorithm which takes several cones C_i's all generated by grevlex non-negative rays (g.n.r.) and returns a cone C generated by p g.n.r. and containing $\bigcup_i C_i$'s.

Finally, we consider the inversion of Laurent series. Let $C \subseteq \mathbb{Z}^p$ be a line-free cone described by a set of grevlex non-negative rays, $\mathbf{R} := \{\mathbf{r}_1, \ldots, \mathbf{r}_m\} \subset \mathbb{Z}^p$, and let $\mathbf{e} \in \mathbb{Z}^p$ be a point. Now, consider

$$0 \neq f = \mathbf{x}^{\mathbf{e}} g(\mathbf{x}^{\mathbf{R}}) \in \mathbf{x}^{\mathbf{e}}\mathbb{K}_C[[\mathbf{x}]],$$

with $g \in \mathbb{K}[[\mathbf{u}]]$. We have:

$$\mathrm{supp}(g(\mathbf{x}^{\mathbf{R}})) = \{(\mathbf{r}_1^T, \ldots, \mathbf{r}_m^T) \cdot \mathbf{k}^T \mid \mathbf{k} \in \mathrm{supp}(g)\} \subseteq \mathbb{Z}^p.$$

Finding the smallest element of the support of the power series g does not guarantee that we can find the **grevlex-minimum** element of the support of he Laurent series f, see [15] for an example.

However, if $\mathbf{R}$ is a set of **grevlex-positive** rays, then

$$\min(\mathrm{supp}(g(\mathbf{x}^{\mathbf{R}}))) = \min(\left\{\overline{\mathbf{R}} \cdot \mathbf{k}^T \mid \mathbf{k} \in \mathrm{supp}(g) \text{ with } \left|\overline{\mathbf{R}} \cdot \mathbf{k}^T\right| \leq \left|\overline{\mathbf{R}} \cdot \overline{\mathbf{k}}^T\right|\right\}),$$

where $\overline{\mathbf{k}} = \min(\mathrm{supp(g)})$ and $\overline{\mathbf{R}} = (\mathbf{r}_1^T, \ldots, \mathbf{r}_m^T)$.

When $\mathbf{R}$ has rays with null total degree, we replace $\left|\overline{\mathbf{R}} \cdot \overline{\mathbf{k}}^T\right|$ by a *guess* bound B and carry computations until the guess is proved to be wrong, in which case B is increased. As an optimization, if g has an analytic expression G, and if G is a rational function, then $\min(\mathrm{supp}(g(\mathbf{x}^{\mathbf{R}})))$ is always computable, even if $\mathbf{R}$ has rays with null total degree.

The Puiseux series object

Let **x** = (x1, ..., xp) be order variables, a multivariate power series **g** in the variables **u**=(u1, ...,um), **R** = (**r1**, ..., **rm**)⊆ $\boldsymbol{Q}^{\boldsymbol{p}}$ be a list of grevlex non-negative vectors and **e**=(e1, ...,ep) ∈ $\boldsymbol{Q}^{\boldsymbol{p}}$. Let also q be the lcm between all the denominators of **r1**, ..., **rm.** Then, ($\boldsymbol{r}1 \cdot q$, ..., $\boldsymbol{r}m \cdot q$)⊆ $\boldsymbol{Z}^{\boldsymbol{p}}$. Thus, $g\left(\boldsymbol{x}^{\boldsymbol{r}1 \cdot q}, \ldots, \boldsymbol{x}^{\boldsymbol{r}1 \cdot q}\right)$ is a Laurent series.

Let's define some Puiseux series.

```
> f1 := PuiseuxSeries(PowerSeries(1/(1+u)), [u=x^(-1/3)*y^2], [x=3, y=-4]);
```

$$f1 := \left[\text{PuiseuxSeries of } \frac{x^3}{\left(1+\frac{y^2}{x^{1/3}}\right) y^4} : \frac{x^3}{y^4} + \ldots\right] \quad \textbf{(5.1)}$$

```
> f2 := PuiseuxSeries(2 + 2*(u + v), [u=x^(-1/2)*y, v=y], [x=3, y=2]);
```

$$f2 := \left[\text{PuiseuxSeries of } \left(2+\frac{2\,y}{\sqrt{x}}+2\,y\right) x^3 y^2 : 2\,x^3 y^2 + 2\,x^{5/2} y^3 + 2\,y^3 x^3\right] \quad \textbf{(5.2)}$$

```
>
```

We can compute the inverse of a Puiseux series (as long as it is possible):

```
> Inverse(f1); Inverse(f2);
```

$$\left[\text{PuiseuxSeries of } \frac{\left(1+\frac{y^2}{x^{1/3}}\right) y^4}{x^3} : \frac{y^4}{x^3} + \ldots\right]$$

$$\left[\text{PuiseuxSeries of } \frac{1}{\left(2+\frac{2\,y}{\sqrt{x}}+2\,y\right) x^3 y^2} : \frac{1}{2\,x^3 y^2} + \ldots\right] \quad \textbf{(5.3)}$$

Fig. 2. *Maple* session showing the computation of the inverse of a Puiseux series.

Figure 2 illustrates how one can create and invert a multivariate Puiseux series with the `MultivariatePowerSeries` library.

4 UPoPS factorization based on lazy evaluation

The goal of this section is to show how the `MultivariatePowerSeries` handles the factorization of univariate polynomials over $\mathbb{A} = \mathbb{K}[[X_1, \ldots, X_n]]$, with the machinery of lazy evaluation.

We denote by $\mathbb{A}[Y]$ the ring of *univariate polynomials over power series*, that we call UPoPS for short. We recall *Weierstrass Preparation Theorem.*

Let $f \in \mathbb{A}[Y]$. Assume $f \not\equiv 0 \bmod \mathcal{M}[Y]$. Write $f = \sum_{i=0}^{d+m} a_i Y^i$, where $d \geq 0$ is the smallest integer such that $a_d \notin \mathcal{M}$ and $m \in \mathbb{Z}^+$. Then, there exists a unique pair (p, α) satisfying $f = p\,\alpha$, where α is an invertible element of $\mathbb{A}[Y]$, and $p = Y^d + b_{d-1}Y^{d-1} + \cdots b_1 Y + b_0$ is a monic polynomial of degree d, such that we have $b_{d-1}, \ldots, b_0 \in \mathcal{M}$.

One can compute the polynomial p and $\alpha = \sum_{i=0}^{m} c_i Y^i$, where a_ℓ, b_j, c_i are power series by means of the following observation:

$$\begin{array}{rcl} a_0 & = & b_0c_0 \\ a_1 & = & b_0c_1 + b_1c_0 \\ a_2 & = & b_0c_2 + b_1c_1 + b_2c_0 \\ & \vdots & \\ a_{d-1} & = & b_0c_{d-1} + b_1c_{d-2} + \cdots + b_{d-2}c_1 + b_{d-1}c_0 \\ a_d & = & b_0c_d + b_1c_{d-1} + \cdots + b_{d-1}c_1 + c_0 \\ a_{d+1} & = & b_0c_{d+1} + b_1c_d + \cdots + b_{d-1}c_2 + c_1 \\ & \vdots & \\ a_{d+m-3} & = & b_{d-3}c_m + b_{d-2}c_{m-1} + b_{d-3}c_{m-2} + c_{m-3} \\ a_{d+m-2} & = & b_{d-2}c_m + b_{d-1}c_{m-1} + c_{m-2} \\ a_{d+m-1} & = & b_{d-1}c_m + c_{m-1} \\ a_{d+m} & = & c_m \end{array}$$

We update p and α by solving the above system of equations modulo $\mathcal{M}^k$, $k = 1, 2, \ldots$. This implies that the UPoPS p and α are continuous functions in Krull topology of the input UPoPS f. Therefore, Weierstrass Preparation Theorem (WPT) can be implemented in a lazy evaluation manner.

It follows that the same is true for Hensel Lemma. To see this, let us restate this key result and exhibit a (short) proof based on WPT.

Let $f = Y^d + \sum_{i=0}^{d-1} a_i Y^i$ be a monic polynomial in $\mathbb{K}[[X_1, \ldots, X_n]][Y]$. Let $\bar{f} = f(0, \ldots, 0, Y) = (Y - c_1)^{d_1}(Y - c_2)^{d_2} \cdots (Y - c_r)^{d_r}$ for $c_1, \ldots, c_r \in \mathbb{K}$ and positive integers $d_1, \ldots, d_r$. Then, there exists $f_1, \ldots, f_r \in \mathbb{K}[[X_1, \ldots, X_n]][Y]$, all monic in Y, such that:

(1) $f = f_1 \cdots f_r$,
(2) $\deg(f_i, Y) = d_i$ for $1 \le i \le r$, and
(3) $\bar{f}_i = (Y - c_i)^{d_i}$ for $1 \le i \le r$.

Indeed. Let $g = f(X_1, \ldots, X_n, Y + c_r) = Y^d + \sum_{i=0}^{d-1} b_i Y^i$, sending c_r to the origin. By construction, we have $b_0, \ldots, b_{d_r-1} \in \mathcal{M}$ and WPT can be applied to produce $g = p\,\alpha$ with $\deg p = d_r$, $\deg \alpha = d - d_r$. Reversing the shift, we have $f_r = p(Y - c_r)$. Induction on $\hat{f} = \alpha(Y - c_r)$ completes the proof.

Consequently, the UPoPS $f_1, \ldots, f_r$ are continuous functions in Krull topology of the input UPoPS f. Therefore, Hensel Lemma can be implemented in a lazy evaluation manner.

It turns out that the same is true for Puiseux Theorem. This follows easily from the proof that Nowak gives of that theorem in [11] since it derives it from Hensel Lemma.

In a future paper, we shall explain how we have implemented the Extended Hensel Construction [13, 1] in a lazy evaluation manner.

We conclude this section with Figure 3 which illustrates Puiseux Theorem in action. To be precise, we see how a UPoPS over Puiseux series is defined and then factorized.

References

[1] Parisa Alvandi, Masoud Ataei, Mahsa Kazemi, and Marc Moreno Maza. On the extended Hensel construction and its application to the computation of real limit points. *J. Symb. Comput.*, 98:120–162, 2020.

[2] M. Asadi, A. Brandt, C. Chen, S. Covanov, F. Mansouri, D. Mohajerani, R. H. C. Moir, M. Moreno Maza, D. Talaashrafi, Linxiao Wang, Ning Xie, and Yuzhen Xie. Basic Polynomial Algebra Subprograms (BPAS), 2021. www.bpaslib.org.

[3] Mohammadali Asadi, Alexander Brandt, Mahsa Kazemi, Marc Moreno Maza, and Erik J. Postma. Multivariate Power Series in Maple. In Robert M. Corless, Jürgen Gerhard, and Ilias S. Kotsireas, editors, *Maple in Mathematics Education and Research*, pages 48–66, Cham, 2021. Springer International Publishing.

Newton-Puiseux factorization

```
> f := UnivariatePolynomialOverPowerSeries([PuiseuxSeries(1), PuiseuxSeries(0), PuiseuxSeries(y, [y = y^(1/3)]),
    PuiseuxSeries(y, [y = y^(1/2)]), PuiseuxSeries(y/(1 + y), [y = y^(1/2)])], x);
```

$$f := [\text{UnivariatePolynomialOverPowerSeries: } (1) + (0)\ x + \left(y^{1/3}\right) x^2 + \left(\sqrt{y}\right) x^3 + (0 + \ldots)\ x^4] \tag{2.1}$$

We compute the Puiseux factorization of f:

```
> F := PuiseuxFactorize(f, returnleadingcoefficient = true);
```

$$F := \left[[\text{UnivariatePolynomialOverPowerSeries: } (0 + \ldots)], \left[\text{UnivariatePolynomialOverPowerSeries: } \left(\frac{1}{\sqrt{y}}\right)\right],\right. \tag{2.2}$$
$$[\text{UnivariatePolynomialOverPowerSeries: } (0 + \ldots) + (0 + \ldots)\ x], [\text{UnivariatePolynomialOverPowerSeries: } (0 + \ldots) + (0 + \ldots)\ x], [\text{UnivariatePolynomialOverPowerSeries: } (0 + \ldots) + (0 + \ldots)\ x],$$
$$\left.[\text{UnivariatePolynomialOverPowerSeries: } (0 + \ldots) + (0 + \ldots)\ x]\right]$$

```
> map(print, map(Truncate, F, 5)):
```

$$y^{5/2} - y^2 + y^{3/2} - y + \sqrt{y}$$
$$\frac{1}{\sqrt{y}}$$
$$y^{1/8}\, x$$
$$y^{1/8}\, x$$
$$y^{1/8}\, x$$
$$y^{1/8}\, x \tag{2.3}$$

Let us see that we got the correct output.

```
> p := Multiply(seq(F));
```

$$p := [\text{UnivariatePolynomialOverPowerSeries: } (0 + \ldots) + (0 + \ldots)\ x + (0 + \ldots)\ x^2 + (0 + \ldots)\ x^3 + (0 + \ldots)\ x^4] \tag{2.4}$$

```
> Truncate(Subtract(f, p), 5);
```

$$0 \tag{2.5}$$

Fig. 3. `MultivariatePowerSeries` session illustrating Puiseux Theorem in action.

[4] Alexander Brandt, Mahsa Kazemi, and Marc Moreno Maza. Power series arithmetic with the BPAS library. In *Computer Algebra in Scientific Computing (CASC '20)*, volume 12291 of *LNCS*, pages 108–128. Springer, 2020.

[5] Alexander Brandt and Marc Moreno Maza. On the complexity and parallel implementation of Hensel's lemma and Weierstrass preparation. In *Computer Algebra in Scientific Computing (CASC '21)*, 2021. (*To appear*).

[6] William H Burge and Stephen M Watt. Infinite structures in scratchpad ii. In *European Conference on Computer Algebra*, pages 138–148. Springer, 1987.

[7] Jerzy Karczmarczuk. Generating power of lazy semantics. *Theoretical Computer Science*, 187(1-2):203–219, 1997.

[8] Wolfram Koepf. Power series in computer algebra. *J. Symb. Comput.*, 13(6):581–604, 1992.

[9] Maplesoft, a division of Waterloo Maple Inc. Maple 2024. www.maplesoft.com/.

[10] Ainhoa Aparicio Monforte and Manuel Kauers. Formal Laurent series in several variables. *Expositiones Mathematicae*, 31(4):350–367, 2013.

[11] Krzysztof Jan Nowak. Some elementary proofs of Puiseux's theorems. *Univ. Iagel. Acta Math*, 38:279–282, 2000.

[12] Erik Postma and Marc Moreno Maza. Substituting units into multivariate power series. *Maple Trans.*, 2(1), 2022.

[13] T. Sasaki and F. Kako. Solving multivariate algebraic equation by Hensel construction. *Japan Journal of Industrial and Applied Mathematics*, 16(2):257–285, 1999.

[14] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.1)*, 2020. `https://www.sagemath.org`.

[15] Juan Pablo González Trochez, Marc Moreno Maza, Erik Postma, and Matt Calder. Laurent series and Puiseux series in maple. *Maple Trans.*, 3(2), 2023.

[16] Joris van der Hoeven. Relax, but don't be too lazy. *Journal of Symbolic Computation*, 34(6):479–542, 2002.

[17] Joris van der Hoeven and Grégoire Lecerf. Mathemagix User Guide. 101 pages, February 2013.

[18] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra (3. ed.)*. Cambridge University Press, 2013.

[19] Wolfram Research Inc. Mathematica, 2024. www.wolfram.com/mathematica.