Algorithms for multivariate Laurent series

Matt Calder[1] , Juan Pablo González Trochez[2], Marc Moreno Maza[2] and Erik Postma[1]

[1]Maplesoft, [2]University of Western Ontario

Overview

A Laurent series is a generalization of a power series in which negative degrees are allowed. Following the ideas of Monforte and Kauers in **[2]**, we present a **first implementation** of multivariate Laurent series in MAPLE. Since we rely on MAPLE's `MultitivariatePowerSeries` package [1] and its lazy evaluation scheme, the minimal element of the support of a given Laurent series object may not be known when we compute with that object. We show how to deal with this challenge when performing arithmetic operations on Laurent series.

Construction

Let $\mathbb{K}$ be a field, $\mathbf{x} = x_1, \ldots, x_p$ and $\mathbf{u} = u_1, \ldots, u_m$ be **ordered indeterminates** with $m \geq p$. The elements $g(\mathbf{u})$ of the ring $\mathbb{K}[[\mathbf{u}]]$ of **multivariate formal power series** look like

$$g(\mathbf{u}) = \Sigma_{\mathbf{k}\in\mathbb{N}^m}\, a_{\mathbf{k}}\mathbf{u}^{\mathbf{k}},$$

for $a_{\mathbf{k}}$ in $\mathbb{K}$, 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 non-negative integers.

The elements $f(\mathbf{x})$ of the field $\mathbb{K}((\mathbf{x}))$ of **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 $\mathbb{K}$.

Let $C \subseteq \mathbb{R}^p$ be a cone. 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 $\mathbb{Z}^p$ and let $\mathcal{C}$ be the set of all cones $C \subseteq \mathbb{R}^p$ which are **compatible** with $\preceq$. Define:

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

Then, $\mathbb{K}_{\preceq}[[\mathbf{x}]]$ is a **ring** and $\mathbb{K}_{\preceq}((\mathbf{x}))$ is a **field**. Our goal is to implement $\mathbb{K}_{\preceq}((\mathbf{x}))$, where $\preceq$ is $<_{glex}$.

Graded reverse lexicographic order

The ***graded reverse lexicographic order*** or ***grevlex*** denoted by $<_{glex}$, for two vectors of $\mathbb{Z}^p$:

- first compares their **total degrees** and
- then uses a **reverse lexicographic order** as tie-breaker.

Example

Set $\mathbf{v}_1 = (1, 0, -1)$, $\mathbf{v}_2 = (0, 0, 0)$, $\mathbf{v}_3 = (1, 1, -1)$, and $\mathbf{v}_4 = (2, -1, -1)$. Then, we have:

$$\mathbf{v}_2 <_{glex} \mathbf{v}_1 <_{glex} \mathbf{v}_4 <_{glex} \mathbf{v}_3.$$

The Laurent series object

Our implementation **encodes** multivariate Laurent series as a ***Laurent series object***, LSO for short, that is, **quintuple** $(\mathbf{x}, \mathbf{u}, \mathbf{e}, \mathbf{R}, g)$, based on the proposition below.

Example

Consider $f := x^{-4}y^5 \sum_{i=0}^{\infty} x^{2i}y^{-i}$. To encode f as an LSO, one can choose:

$$\begin{aligned}\mathbf{x} &= [x, y],\ \mathbf{u} = [u, v],\\ g &= \mathrm{Inverse}(\mathrm{PowerSeries}(1 + uv)),\\ \mathbf{R} &= [[1, 0], [1, -1]],\\ \mathbf{e} &= [x = -4, y = 5].\end{aligned}$$

Proposition: the Laurent series object

Let $g \in \mathbb{K}[[\mathbf{u}]]$ be a power series, $\mathbf{e} \in \mathbb{Z}^p$ 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** living in $\mathbf{x}^{\mathbf{e}}\mathbb{K}_C[[\mathbf{x}]]$, where C is the cone generated by $\mathbf{R}$.

Addition and multiplication

Let $C_1, C_2 \subseteq \mathbb{Z}^p$ be two cones generated, respectively, by two sets of **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. $\mathrm{supp}(f_1 + f_2)$). To this end, we developed an algorithm which takes as input a number of cones $C_1, C_2, \ldots$ all generated by grevlex non-negative rays and returns a cone C generated by p grevlex non-negative rays and such that C contains the union of $C_1, C_2, \ldots$.

Inversion

For an LSO $f = (\mathbf{x}, \mathbf{u}, \mathbf{e}, \mathbf{R}, g)$, knowing $\min(\mathrm{supp(g)})$ would not guarantee finding the **grevlex-minimum** element of $\mathrm{supp}(f)$, if $\mathbf{R}$ has rays with null total degree. However, if $\mathbf{R}$ is a set of **grevlex-positive** rays, $\min\mathrm{supp}(g(\mathbf{x}^{\mathbf{R}}))$ equals

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

where $\bar{\mathbf{k}} = \min(\mathrm{supp(g)})$ and $\mathbf{R} = (\mathbf{r}_1^T, \ldots, \mathbf{r}_m^T)$. When $\mathbf{R}$ has rays with null total degree, we replace $\left|\mathbf{R}\cdot\bar{\mathbf{k}}^T\right|$ 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 a known analytic form G, see [1], 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.

Algorithm 1 Inverse

Require: Laurent series $f(\mathbf{x}) = \mathbf{x}^{\mathbf{e}}g(\mathbf{x}^{\mathbf{R}})$.
Ensure: The inverse f^{-1} of f.
1: **if** AnalyticExpression(f) = **Undefined** or **non-rational then**
2: **return** $\mathbf{x}^{-\mathbf{e}}$InverseOfUndefinedAnalyticExpression$(g(\mathbf{x}^{\mathbf{R}}))$
3: **else**
4: $q :=$ AnalyticExpression(f) ▷ The analytic expression of f.
5: **return** $\mathbf{x}^{-\mathbf{e}}$InverseOfAnalyticExpression$(q, \mathbf{x}^{\mathbf{R}})$

Maple overview

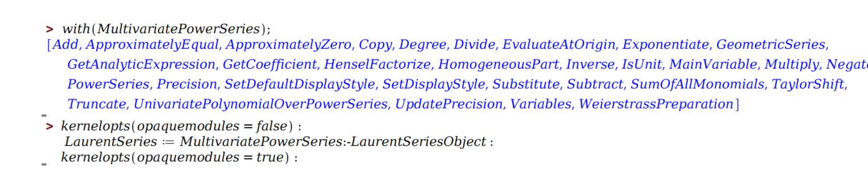

Figure 1: Laurent series object

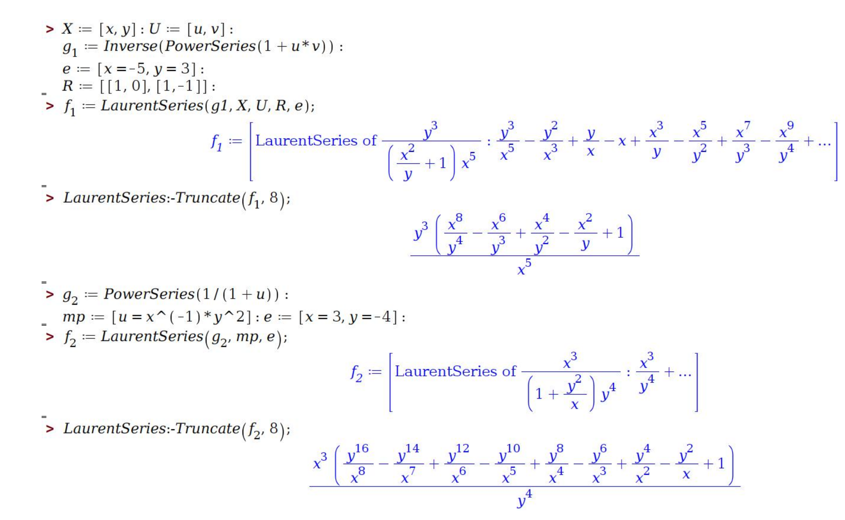

Figure 2: Creation of Laurent series

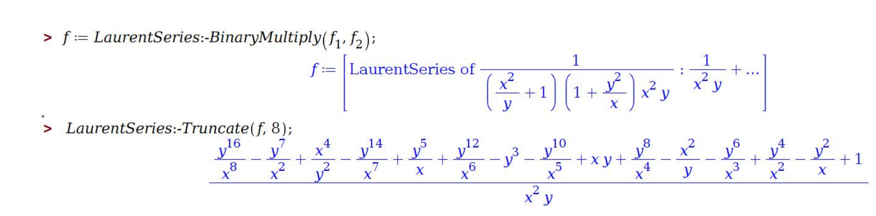

Figure 3: Multiplication of Laurent series

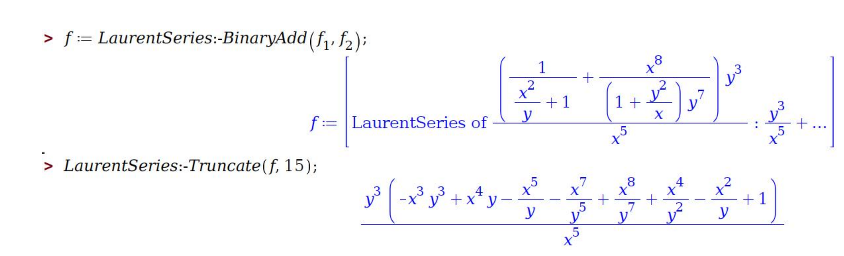

Figure 4: Addition of Laurent series

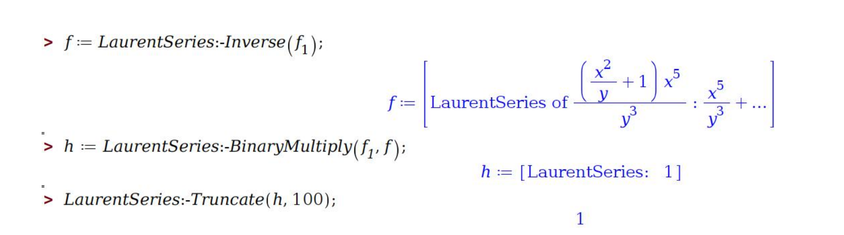

Figure 5: Inversion of a Laurent series

References

[1] Mohammadali Asadi, Alexander Brandt, Mahsa Kazemi, Marc Moreno-Maza, and Erik Postma. Multivariate power series in Maple. Springer International Publishing, 2021.

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