Implementation Techniques for Power, Laurent, and Puiseux Series in Several Variables

Marc Moreno Maza

Ontario Research Center for Computer Algebra
Departments of Computer Science and Mathematics
University of Western Ontario, Canada

CASC 2022, Gebze, Turkey, 23 August 2022

Acknowledgements

- Many thanks to the CASC 2022 organizers, in particular François Boulier and Matthew England.
- I would have loved to visit Gebze Technical University and our local colleagues.
- This tutorial is based on research projects in which some of my former and current graduate students have played an essential role. By alphabetic order: Parisa Alvandi, Masoud Ataei, Ali Asadi, Alexander Brandt Changbo Chen, Mahsa Kazemi, Juan-Pablo Gonzàlez Trochez.
- This tutorial is based on my collaboration with Erik Postma, with funding support from Maplesoft, MITACS and NSERC of Canada.
- Special thanks go to Juan-Pablo Gonzàlez Trochez who helped me prepare the *Maple* worksheetp illustrating this talk.

- The *Maple* package `MultivariatePowerSeries` implement power, Laurent and Puiseux series, as presented in this talk, see [5, 8].
- Multivariate power series, as presented in this talk, are implemented in the Basic Polynomial Algebra Subprograms (BPAS) [4].

Talk features and tentative plan

- We will not review the theory of formal power series, Laurent series and Puiseux series, but we will have examples ☺.
- A detailed review can be found in my CASC 20218 tutorial.
- This **talk** is dedicated to the key implementation strategies of **MultivariatePowerSeries** and its BPAS counterpart.
- We will not cover benchmarks and complexity analysis of the underlying algorithms; they can be found in our papers [1, 2, 5–8, 13].
- But we will use this **worksheet**

Tentative plan

- Part 1: Motivations
- Part 2: formal power series
- Part 3: Laurent series
- Part 4: Puiseux series in *Maple*'s `MultivariatePowerSeries` library.

Outline

Outline

A first example: computing limit lines

The question

Consider the following set of polynomials

$$F = \begin{cases} x^2 a + y + 1 \\ y^2 b + x + 1 \end{cases}$$

and the following regular chain for $x > y > a > b$:

$$T = \begin{cases} x + y^2 b + 1 \\ a\,b^2 y^4 + 2ab\,y^2 + y + a + 1 \end{cases}$$

It turns that we have

$$V(F) = \overline{W(T)} \text{ where } W(T) = V(T) \setminus V(h_T) \text{ and } h_T = a\,b.$$

How to compute $\overline{W(T)} \setminus W(T)$?

A first example: computing limit lines

Unsatisfactory answers

With $F = \{x^2a + y + 1, y^2b + x + 1\}$, existing algorithms for decomposing polynomial systems:

- either return T and do not compute $\overline{W(T)} \setminus W(T)$ explicitly,
- or returns T with an explicit decomposition of $\overline{W(T)} \setminus W(T)$, obtained by recursively decomposing $V(F \cup \{h_T\})$.

`RegularChains:-Triangularize` produces

$$V(F) = W(T) \cup W(T_a) \cup W(T_b), \quad \text{where}$$

$$T = \begin{cases} x + y^2b + 1 \\ a\,b^2y^4 + 2ab\,y^2 + y + a + 1 \end{cases}, \; T_a = \begin{cases} x + b + 1 \\ y + 1 \\ a \end{cases}, \text{and } \; T_b = \begin{cases} x + 1 \\ y + a + 1 \\ b \end{cases}.$$

Ideally

One would like to obtain $W(T_a)$ and $W(T_b)$ as *limit solutions* of $W(T)$.

A second example: computing limit points

A 1-dimensional regular chain T in $\mathbb{C}[x_1 < x_2 < \cdots < x_n]$ typically looks like

$$T : \begin{cases} t_2(x_1, x_2) & = & h_2(x_1) x_2^{d_2} + \cdots \\ t_3(x_1, x_2, x_3) & = & h_3(x_1) x_3^{d_3} + \cdots \\ \vdots & \vdots & \vdots \\ t_n(x_1, x_2, \ldots, x_n) & = & h_n(x_1) x_n^{d_n} + \cdots \end{cases} \quad (1)$$

- T can be seen as a parametrization of a space curve C, namely $C = \overline{W(T)}$, where $W(T) = V(T) \setminus V(h)$ and $h_T = \prod_{i=2}^{n} h_i$
- $\overline{W(T)} \setminus W(T) = \{$limits points of C when x_1 approaches $\zeta \mid \zeta$ a root of $h\}$.
- We can compute these limit points by factorizing $t_2, t_3, \ldots, t_n$ over the field $\mathbb{C}((x_1^*))$ of univariate Puiseux series in x_1, see [2].

Example

Let $T \subseteq \mathbb{K}[x > y > z]$ be a regular chain

$$T := \left\{ \begin{array}{l} z\,x - y^2 \\ y^5 - z^4 \end{array} \right. .$$

In this case: $h = z$ and $\zeta = 0$.
Then, over $\mathbb{C}((z^*))$

$$\mathcal{V}(T) = \{(x = z^{3/5}, y = z^{4/5})\}$$

Thus we have:

$$\overline{W(T)} \setminus W(T) = \{(0,0,0)\}.$$

Example

Let $T \subseteq \mathbb{K}[x > y > z]$ be a regular chain

$$T := \begin{cases} z\,x - y^2 \\ y^5 - z^4 \end{cases}.$$

In this case: $h = z$ and $\zeta = 0$. Then, over $\mathbb{C}((z^*))$

$$\mathcal{V}(T) = \{(x = z^{3/5}, y = z^{4/5})\}$$

Thus we have:

$$\overline{W(T)} \setminus W(T) = \{(0,0,0)\}.$$

Example

Consider $T \subseteq \mathbb{K}[x > y > z]$:

$$T := \begin{cases} z\,x - y^2 = 0 \\ y^5 - z^2 = 0 \end{cases}.$$

In this case: $h = z$ and $\zeta = 0$. Then, over $\mathbb{C}((z^*))$

$$\mathcal{V}(T) = \{(x = z^{-1/5}, y = z^{2/5})\}$$

Since the Puiseux series $z^{-1/5}$ has a **negative order**, we have:

$$\overline{W(T)} \setminus W(T) = \varnothing.$$

Outline

Notations

Formal power series and formal Laurent series

- Let $\mathbb{K}$ be a field and $\overline{\mathbb{K}}$ its algebraic closure.
- $\mathbb{K}[[X_1, \ldots, X_n]]$ denotes the ring (actually UFD) of formal power series in $X_1, \ldots, X_n$ over $\mathbb{K}$ and $\mathcal{M} = \langle X_1, \ldots, X_n \rangle$ its unique maximal ideal.
- $\mathbb{K}((X_1, \ldots, X_n))$ is the fraction field of $\mathbb{K}[[X_1, \ldots, X_n]]$.

Univariate Puiseux series

- $\mathbb{K}[[U^*]] = \bigcup_{\ell=1}^{\infty} \mathbb{K}[[U^{\frac{1}{\ell}}]]$ the ring of *formal univariate Puiseux series*.
- Hence, given $\varphi \in \mathbb{K}[[U^*]]$, there exists $\ell \in \mathbb{N}_{>0}$ such that $\varphi \in \mathbb{K}[[U^{\frac{1}{\ell}}]]$ holds. Thus, we can write $\varphi = \sum_{m=0}^{\infty} a_m U^{\frac{m}{\ell}}$, for $a_0, \ldots, a_m, \ldots \in \mathbb{K}$.
- We denote by $\mathbb{K}((U^*))$ the quotient field of $\mathbb{K}[[U^*]]$.
- Puiseux's theorem: if $\mathbb{K}$ is an algebraically closed field of characteristic zero, then $\mathbb{K}((U^*))$ is the algebraic closure of $\mathbb{K}((U))$.

Algorithms for factoring in $\mathbb{C}((X_1^*, \ldots X_n^*))[Y]$

With non-explicit use of Puiseux series

- the original Newton-Puiseux algorithm and its variants manipulate every $\varphi \in \mathbb{K}[[U^*]]$ as a pair $(\ell, \psi \in \mathbb{K}[[T]])$, where $T^\ell = U$.
- the Extended Hensel Construction (EHC), invented by T. Sasaki and his students, allows to factor in $\mathbb{C}((X_1^*, \ldots X_n^*))[Y]$, see [16].
- The EHC can be implemented efficiently and outperform theoretically faster algorithms, see [1].

With explicit use of Puiseux series

- In [15], K. J. Nowak reduces factorization in $\mathbb{K}((U^*))[Y]$ to Hensel lifting in $\mathbb{K}[[U]][Y]$ and avoids the “corner cases” of the EHC.
- Explicit computations in $\mathbb{K}((U^*))[Y]$ are needed before the reduction.

Algorithms for factoring in $\mathbb{C}((X_1^*, \ldots X_n^*))[Y]$

```
> alias(T = RootOf(_Z^2 + y)) :
> P := PowerSeries([y, z]) :
  U := UnivariatePolynomialOverPowerSeries([y, z], x) :
  poly := y·x^3 + (-2 · y + z + 1)·x + y :
  U:-ExtendedHenselConstruction(poly, [0, 0], 3);
```

$$\left[\left[x = \frac{-T + T\,y - \frac{1}{2}\,T\,z + \frac{1}{2}\,y^2}{y}\right], \left[x = \frac{T - T\,y + \frac{1}{2}\,T\,z + \frac{1}{2}\,y^2}{y}\right], [x = -y]\right]$$

In the above *Maple* session:

- we factor $f = yx^3 + (-2y + z + 1)x + y$ in $\mathbb{C}((y^*, z^*))[x]$ using the EHC implemented in the `RegularChains` library.
- Note that the factors are not expanded over the monomial basis of $\mathbb{C}((y^*, z^*))[x]$.
- Instead, algebraic functions are introduced dynamically by the algorithm.
- For *pretty printing* reasons only, we introduce the alias $T = y^{1/2}$ before the call to the EHC.

Outline

Outline

Power series: notations

$\mathbb{A} = \mathbb{K}[[X_1, \ldots, X_n]]$ is the **ring of multivariate formal power series**

- $\mathbb{K}$ an algebraically closed.
- $f = \sum_e a_e X^e \in \mathbb{K}[[X_1, \ldots, X_n]]$
- $X^e = X_1^{e_1} \cdots X_n^{e_n}$, $|e| = e_1 + \cdots + e_n$
- $\mathcal{M} = \langle X_1, \ldots, X_n \rangle$ is the maximal ideal of $\mathbb{A}$
- **homogeneous part** of degree k: $f_{(k)} = \sum_{|e|=k} a_e X^e$ and we have $f_{(k)} \in \mathcal{M}^k \setminus \mathcal{M}^{k+1}$

Example:

$f = 1 + X_1 + X_1 X_2 + X_2^2 + X_1 X_2^2 + X_1^3 + \cdots$ is known to **precision** 3

$$f_{(1)} = X_1 \qquad f_{(2)} = X_1 X_2 + X_2^2 \qquad f_{(3)} = X_1 X_2^2 + X_1^3$$

$\mathbb{A}[Y]$ is the ring of **Univariate Polynomials over Power Series** (UPoPS)

- $f = \sum_{i=0}^{d} a_i Y^i$, $a_i \in \mathbb{A}$, $a_d \neq 0$, is a UPoPS of degree d

Lazy evaluation scheme

Motivations:

1. Only compute terms explicitly needed:
   ↳ requested by user or needed for subsequent operations
2. Ability to **resume** and increase precision of an existing power series

Lazy evaluation scheme

Motivations:

1. Only compute terms explicitly needed:
   ↳ requested by user or needed for subsequent operations
2. Ability to **resume** and increase precision of an existing power series

Data-structure:

1. stores previously computed homogeneous parts;
2. returns previously computed homogeneous parts and, otherwise,
3. uses an **update function** to compute homogeneous parts as needed;
4. captures parameters required for the update function and effectively create a **closure**.

Where update parameters are power series, they are called **ancestors**.

Addition, $f = g + h$

- $f_{(k)} = g_{(k)} + h_{(k)}$

Multiplication $f = gh$

- $f_{(k)} = \sum_{i=0}^{k} g_{(i)} h_{(k-i)}$

Ancestry example

$$p = fg + ab$$

$f = 1 + x + yz + \dots$

$g = 1 + z + y + \dots$

$a = 1 + y + x^2 + \dots$

$b = 1 + yz + xz + \dots$

$f \times g \longrightarrow h = 1 + z + y + x + yz + xz + xy + \dots$

$a \times b \longrightarrow c = 1 + y + yz + xz + x^2 + \dots$

$h + c \longrightarrow p = 2 + z + 2y + x + 2yz + 2xz + xy + x^2 + \dots$

Why lazy evaluation works?

- $\mathcal{M} = \langle X_1, \ldots, X_n \rangle$ is the unique maximal ideal of $\mathbb{K}[[X_1, \ldots, X_n]]$.
- For $d \geq 0$, $\mathcal{M}^d$ is generated by the monomials of degree d.
- We have $\mathcal{M}^{d+1} \subseteq \mathcal{M}^d$ and $\bigcap_{k \in \mathbb{N}} \mathcal{M}^k = \langle 0 \rangle$.

Krull topology

- 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 $\mathbb{K}[[\underline{X}]]$ and let $f \in \mathbb{K}[[\underline{X}]]$. The sequence $(f_n)_{n \in \mathbb{N}}$ *converges* to f if for all $k \in \mathbb{N}$ there exists $N \in \mathbb{N}$ s.t. for all $n \in \mathbb{N}$ we have $n \geq N \quad \Rightarrow \quad f - f_n \in \mathcal{M}^k$,
- Therefore, a bivariate function $f : \mathbb{K}[[\underline{X}]] \times \mathbb{K}[[\underline{X}]] \longmapsto \mathbb{K}[[\underline{X}]]$ is continuous at (p, q) if for every $d \in \mathbb{N}$ we can find $b, c \in \mathbb{N}$ such that
$$f(p + \mathcal{M}^b, q + \mathcal{M}^c,) - f(p, q) \subseteq \mathcal{M}^d.$$
- Continuous functions are those which can be implemented by lazy evaluation; this is the case for addition, multiplication, inversion.

Outline

Weierstrass Preparation Theorem in $\mathbb{K}[[\underline{X}]][[Y]]$

Theorem (Weierstrass Preparation)

Let $f \in \mathbb{K}[[X_1, \ldots, X_n]][Y]$*. Assume* $f \not\equiv 0 \mod \mathcal{M}[Y]$*. Write* $f = \sum_{i=0}^{d+m} a_i Y^i \in \mathbb{K}[[X_1, \ldots, X_n]][Y]$ *where* $d \geq 0$ *be the smallest integer such that* $a_d \notin \mathcal{M}$ *and* $m \in \mathbb{Z}^+$*.*
Then, there exists a unique pair (p, α) *satisfying the following:*

1. $f = p\,\alpha$,
2. α *is an invertible element of* $\mathbb{K}[[X_1, \ldots, X_n]][[Y]]$,
3. p *is a monic polynomial of degree* d,
4. *writing* $p = Y^d + b_{d-1}Y^{d-1} + \cdots b_1 Y + b_0$*, we have* $b_{d-1}, \ldots, b_0 \in \mathcal{M}$.

Lazy evaluation for Weierstrass Preparation

Let $f = \sum_{\ell}^{d+m} a_{\ell=0} Y^{\ell}$, $p = Y^d + \sum_{j=0}^{d-1} b_j Y^j$, $\alpha = \sum_{i=0}^{m} c_i Y^i$ be UPoPS.

↳ a_ℓ, b_j, c_i are power series ↳ $b_j \in \mathcal{M}$ for $j = 0, \ldots, d-1$

$$f = \alpha p \implies \begin{array}{rcl} a_0 & = & b_0 c_0 \\ a_1 & = & b_0 c_1 + b_1 c_0 \\ & \vdots & \\ a_{d-1} & = & b_0 c_{d-1} + b_1 c_{d-2} + \cdots + b_{d-2} c_1 + b_{d-1} c_0 \\ a_d & = & b_0 c_d + b_1 c_{d-1} + \cdots + b_{d-1} c_1 + c_0 \\ & \vdots & \\ a_{d+m-1} & = & b_{d-1} c_m + c_{m-1} \\ a_{d+m} & = & c_m \end{array}$$

We update p and α by solving these equations modulo $\mathcal{M}^k$, $k = 1, 2, \ldots$

Moreover, the structure of those equations yields interesting parallel patterns (parallel mao-reduce) and a fine complexity analysis, see [7].

Outline

Hensel's Lemma

Theorem (Hensel's Lemma)

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:*

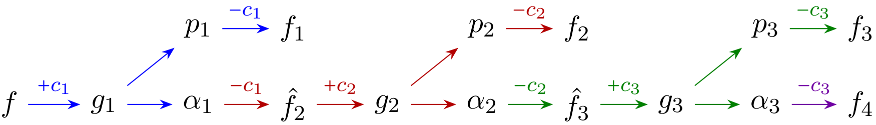
1. $f = f_1 \cdots f_r$,
2. $\deg(f_i, Y) = d_i$ *for* $1 \le i \le r$, *and*
3. $\bar{f}_i = (Y - c_i)^{d_i}$ *for* $1 \le i \le r$.

Proof:

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, $b_0, \ldots, b_{d_r - 1} \in \mathcal{M}$ and Weierstrass preparation can be applied to produce $g = p\,\alpha$ with $\deg p = d_r, \deg \alpha = d - d_r$.

Reversing the shift, $f_r = p(Y - c_r)$.
Induction on $\hat{f} = \alpha(Y - c_r)$ completes the proof. □

Hensel lifting in a pipeline



- The output of one Weierstrass becomes input to another
- $f_{i+i(k)}$ relies on $f_{i(k)}$
- Can compute $f_{i(k+1)}$ and $f_{i+i(k)}$ concurrently in a **pipeline**
- See [7] for complexity analysis and implementation report.

	Stage 1 (f_1)	Stage 2 (f_2)	Stage 3 (f_3)	Stage 4 (f_4)
Time 1	$f_{1(1)}$			
Time 2	$f_{1(2)}$	$f_{2(1)}$		
Time 3	$f_{1(3)}$	$f_{2(2)}$	$f_{3(1)}$	
Time 4	$f_{1(4)}$	$f_{2(3)}$	$f_{3(2)}$	$f_{4(1)}$
Time 5	$f_{1(5)}$	$f_{2(4)}$	$f_{3(3)}$	$f_{4(2)}$

Hensel or EHC: how to decide which one to use?

The problem

Hensel's lemma requires $f \in \mathbb{K}[[X_1, \ldots, X_n]][Y]$ to be monic. How to check that requirement and avoids calling the EHC which handles the non-monic case but does potentially more work.

The answer: closed-form expression

The `MultivariatePowerSeries` library package frequently has some information beyond just the homogeneous-component procedure:

- the user can specify a closed-form expression for the power series:

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

- this closed-form expression is also automatically given for `MultivariatePowerSeries` objects constructed by most other commands, in particular rational functions.

Outline

Substituting non-units into power series

- This is well-defined, but can it be done by lazy evaluation?
- For simplicity, we describe the univariate case and refer to [13] for the multivariate one.
- The input is $a := \sum_i a_i X^i$ and $b := \sum_j b_j X^j$. We want $a|_{X=b}$

Substituting non-units into power series

- This is well-defined, but can it be done by lazy evaluation?
- For simplicity, we describe the univariate case and refer to [13] 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:

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

Substituting non-units into power series

- This is well-defined, but can it be done by lazy evaluation?
- For simplicity, we describe the univariate case and refer to [13] 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:

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

- 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.

Substituting non-units into power series

- Up to elementary expansions, we had above:

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

Substituting non-units into power series

- Up to elementary expansions, we had above:

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

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

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

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 $\sum_j j m_j = i$.

Substituting non-units into power series

- Up to elementary expansions, we had above:

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

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

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

  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 $|\underline{m}| \le \sum_j j m_j = i$.

Substituting non-units into power series

- Up to elementary expansions, we had above:

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

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

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

  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 $a|_{X=b}$.

Substituting non-units into power series

- Up to elementary expansions, we had above:

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

- 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 $a|_{X=b}$.
- In a sum, this process is continuous in the Krull topology.

Substituting units into power series

- Consider $a = b = \sum_{i=0}^{\infty} \frac{X^i}{i!}$, which we traditionally view as e^X.

Substituting 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

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

Substituting 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

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

- Because of the factor of e, substituting units into power series cannot be continuous in the Krull topology.

Substituting 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

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

- Because of the factor of e, substituting units into power series cannot be continuous in the Krull topology.
- There is a work-around, however. Do you guess which one?

Substituting 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

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

- Because of the factor of e, substituting units into power series cannot be continuous in the Krull topology.
- There is a work-around, however. Do you guess which one?
- If two multivariate power series a, b have closed form 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

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

Substituting 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

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

- Because of the factor of e, substituting units into power series cannot be continuous in the Krull topology.
- There is a work-around, however. Do you guess which one?
- If two multivariate power series a, b have closed form 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 [13].

Outline

Outline

The domain $\mathbb{K}_C[[\mathbf{x}]]$

- In this sub-section, we follow Monforte and Kauers, see [12].
- 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 domain $\mathbb{K}_C[[\mathbf{x}]]$

- In this sub-section, we follow Monforte and Kauers, see [12].
- 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 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 $\mathbb{K}$, 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 domain $\mathbb{K}_C[[\mathbf{x}]]$

- In this sub-section, we follow Monforte and Kauers, see [12].
- 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 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 $\mathbb{K}$, 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.
- Let $C \subseteq \mathbb{R}^p$ be a cone. All cones here are **line-free**, polyhedral and generated by **integer** vectors.

The domain $\mathbb{K}_C[[\mathbf{x}]]$

- In this sub-section, we follow Monforte and Kauers, see [12].
- 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 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 $\mathbb{K}$, 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.

- 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:

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

The domain $\mathbb{K}_C[[\mathbf{x}]]$

- In this sub-section, we follow Monforte and Kauers, see [12].
- 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 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 $\mathbb{K}$, 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.

- 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 $a_{\mathbf{0}} \neq 0$.

The field $\mathbb{K}_{\preceq}((\mathbf{x}))$

- 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:

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

The field $\mathbb{K}_{\preceq}((\mathbf{x}))$

- 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 $\mathbf{0} \preceq \mathbf{k}$.

The field $\mathbb{K}_{\preceq}((\mathbf{x}))$

- 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:

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

The field $\mathbb{K}_{\preceq}((\mathbf{x}))$

- 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:

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

- 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 field $\mathbb{K}_{\preceq}((\mathbf{x}))$

- 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:

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

- 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 $\frac{x}{1-x}$.
- Then, $\mathbb{K}_{\preceq}[[\mathbf{x}]]$ is a **ring** and $\mathbb{K}_{\preceq}((\mathbf{x}))$ is a **field**.

The field $\mathbb{K}_{\preceq}((\mathbf{x}))$

- 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:

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

- 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 $\frac{x}{1-x}$.
- 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 $<_{glex}$.

The field $\mathbb{K}_{\preceq}((\mathbf{x}))$

- 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:

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

- 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 $\frac{x}{1-x}$.
- 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 $<_{glex}$.
- Recall that $<_{glex}$ first compares **total degrees** before using **reverse lexicographic order** as tie-breaker.

Outline

Data-structure

Lemma

Recall $\mathbf{x} = x_1, \ldots, x_p$ *and* $\boldsymbol{u} = u_1, \ldots, u_m$. *Let* $g \in \mathbb{K}[[\boldsymbol{u}]]$, $\boldsymbol{e} \in \mathbb{Z}^p$ *be a point, and* $\boldsymbol{R} := \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_m\} \subset \mathbb{Z}^p$ *be a set of* ***grevlex non-negative*** *rays. Then,*

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

is a ***Laurent series*** *in* $\mathbf{x}^{\boldsymbol{e}}\mathbb{K}_C[[\mathbf{x}]]$, *where* C *is the cone generated by* $\boldsymbol{R}$.

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

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))`.

Outline

Addition and multiplication

- 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 $m \geq p$.

Addition and multiplication

- 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 $\mathbf{e}_1, \mathbf{e}_2 \in \mathbb{Z}^p$.

Addition and multiplication

- 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:

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

Addition and multiplication

- 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:

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

Addition and multiplication

- 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. $\mathrm{supp}(f_1 + f_2)$).

Addition and multiplication

- 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 $\operatorname{supp}(f_1 f_2)$ (resp. $\operatorname{supp}(f_1 + f_2)$).
- 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, see [8].

Outline

Inversion: understanding the challenge

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. Let us see an example.

Inversion: illustrating the challenge

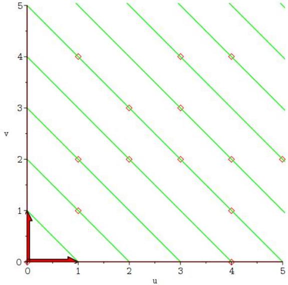
Consider a power series $g \in \mathbb{K}[[u,v]]$ with support equal to

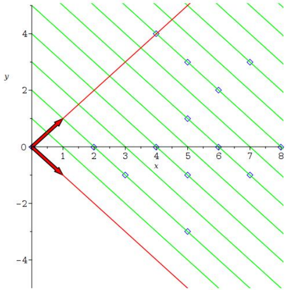
$$\{(0,0),(1,1),(1,2),(1,4),(2,2),\\ (2,3),(3,2),(3,3),(3,4),(4,0),\\ (4,1),(4,2),(4,4),(5,2),\ldots\},$$

a random infinite set.

Then, the support of $g(xy, xy^{-1})$ is going to be equal to

$$\{(0,0),(2,0),(3,-1),(5,-3),(4,0),\\ (5,-1),(5,1),(6,0),(7,-1),(4,4),\\ (5,3),(6,2),(8,0),(7,3),\ldots\}.$$





Inversion: our solutions

- As just illustrated, knowing $\min(\mathrm{supp}(\mathrm{g}))$ would not guarantee finding the **grevlex-minimum** element of $\mathrm{supp}(f)$, if **R** has rays with null total degree.

Inversion: our solutions

- As just illustrated, knowing $\min(\mathrm{supp}(\mathrm{g}))$ would not guarantee finding the **grevlex-minimum** element of $\mathrm{supp}(f)$, if **R** has rays with null total degree.
- However, if **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 $\overline{\mathbf{k}} = \min(\mathrm{supp}(\mathrm{g}))$ and $\overline{\mathbf{R}} = (\mathbf{r}_1^T, \ldots, \mathbf{r}_m^T)$.

Inversion: our solutions

- As just illustrated, knowing $\min(\text{supp}(g))$ would not guarantee finding the **grevlex-minimum** element of $\text{supp}(f)$, if **R** has rays with null total degree.
- However, if **R** is a set of **grevlex-positive** rays, then

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

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

- When **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.

Inversion: our solutions

- As just illustrated, knowing $\min(\mathrm{supp}(\mathrm{g}))$ would not guarantee finding the **grevlex-minimum** element of $\mathrm{supp}(f)$, if **R** has rays with null total degree.
- However, if **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 $\overline{\mathbf{k}} = \min(\mathrm{supp}(\mathrm{g}))$ and $\overline{\mathbf{R}} = (\mathbf{r}_1^T, \ldots, \mathbf{r}_m^T)$.
- When **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 a closed-form expression G, and if G is a rational function, then $\min(\mathrm{supp}())g(\mathbf{x}^{\mathbf{R}})$ is always computable, even if **R** has rays with null total degree.

Concluding remarks

- Lazy evaluation is simply the implementation of continuity in the Krull topology

Concluding remarks

- Lazy evaluation is simply the implementation of continuity in the Krull topology
- How many times have we been saved by the *closed-form expression* trick?

Concluding remarks

- Lazy evaluation is simply the implementation of continuity in the Krull topology
- How many times have we been saved by the *closed-form expression* trick?
- Three times: Hensel's lemma, composition and Puiseux series inversion.

Concluding remarks

- Lazy evaluation is simply the implementation of continuity in the Krull topology
- How many times have we been saved by the *closed-form expression* trick?
- Three times: Hensel's lemma, composition and Puiseux series inversion.
- In the case of composition, it allowed us to go beyond the well-known case of substituting non-units into power series.

Concluding remarks

- Lazy evaluation is simply the implementation of continuity in the Krull topology
- How many times have we been saved by the *closed-form expression* trick?
- Three times: Hensel's lemma, composition and Puiseux series inversion.
- In the case of composition, it allowed us to go beyond the well-known case of substituting non-units into power series.
- Our lazy scheme for Weierstrass Preparation and Hensel lifting can be finely analyzed and provide interesting parallel patterns.

Concluding remarks

- Lazy evaluation is simply the implementation of continuity in the Krull topology
- How many times have we been saved by the *closed-form expression* trick?
- Three times: Hensel's lemma, composition and Puiseux series inversion.
- In the case of composition, it allowed us to go beyond the well-known case of substituting non-units into power series.
- Our lazy scheme for Weierstrass Preparation and Hensel lifting can be finely analyzed and provide interesting parallel patterns.
- The same is expected to be true for Nowak's construction.

Concluding remarks

- Lazy evaluation is simply the implementation of continuity in the Krull topology
- How many times have we been saved by the *closed-form expression* trick?
- Three times: Hensel's lemma, composition and Puiseux series inversion.
- In the case of composition, it allowed us to go beyond the well-known case of substituting non-units into power series.
- Our lazy scheme for Weierstrass Preparation and Hensel lifting can be finely analyzed and provide interesting parallel patterns.
- The same is expected to be true for Nowak's construction.
- Today, *Maple*'s language and *Maple*'s supporting kernel libraries allow for very effective implementation: `MultivariatePowerSeries` is only an an order of magnitude slower than its BPAS counterpart (which is essentially pure C code) when both execute serially, see [5].

Work in progress

- In general, power series produced by Weierstrass Preparation and Hensel's lemma are not rational expressions in the input power series. Hence, currently, their closed-forms are not computed.

Work in progress

- In general, power series produced by Weierstrass Preparation and Hensel's lemma are not rational expressions in the input power series. Hence, currently, their closed-forms are not computed.
- However, those (former) power series are algebraic in the input power series. Hence, something can be done.

Work in progress

- In general, power series produced by Weierstrass Preparation and Hensel's lemma are not rational expressions in the input power series. Hence, currently, their closed-forms are not computed.
- However, those (former) power series are algebraic in the input power series. Hence, something can be done.
- Nowak's construction is conceptually much simpler than the EHC but the EHC avoids expanding expressions by introducing algebraic functions. Who will win? Can we combine the best of both worlds? We should know soon 🙂.

References

[1] P. Alvandi, M. Ataei, M. Kazemi, and M. Moreno Maza. "On the Extended Hensel Construction and its application to the computation of real limit points". In: *Journal of Symbolic Computation* 98 (2020), pp. 120–162.

[2] P. Alvandi, C. Chen, and M. M. Maza. "Computing the Limit Points of the Quasi-component of a Regular Chain in Dimension One". In: *Proc. of CASC 2013*. Vol. 8136. Lecture Notes in Computer Science. Springer, 2013, pp. 30–45.

[3] P. Alvandi, M. Kazemi, and M. Moreno Maza. "Computing limits with the regularchains and powerseries libraries: from rational functions to Zariski closure". In: *ACM Communications in Computer Algebra* 50.3 (2016), pp. 93–96.

[4] M. Asadi, A. Brandt, C. Chen, S. Covanov, M. Kazemi, F. Mansouri, D. Mohajerani, R. H. C. Moir, M. Moreno Maza, D. Talaashrafi, L. Wang, N. Xie, and Y. Xie. *Basic Polynomial Algebra Subprograms (BPAS) v. 1.791*. http://www.bpaslib.org. 2022.

[5] M. Asadi, A. Brandt, M. Kazemi, M. Moreno Maza, and E. Postma. "Multivariate Power Series in Maple". In: *Proc. of MC 2020*. Vol. 1414. Communications in Computer Information Science. 2020, pp. 48–66.

[6] A. Brandt, M. Kazemi, and M. Moreno Maza. "Power Series Arithmetic with the BPAS Library". In: *Proc. of CASC 2020*. Vol. 12291. Lecture Notes in Computer Science. Springer, 2020, pp. 108–128.

[7] A. Brandt and M. Moreno Maza. "On the Complexity and Parallel Implementation of Hensel's Lemma and Weierstrass Preparation". In: *Proc. of CASC 2021*. Vol. 12865. Lecture Notes in Computer Science. Springer, 2021, pp. 78–99.

[8] M. Calder, J. P. González Trochez, M. Moreno Maza, and E. Postma. *Algorithms for multivariate Laurent series*. Chapter 4 in the MSc Thesis of Juan Pablo González Trochez, submitted to MC 2022. 2022.

[9] D. V. Chudnovsky and G. V. Chudnovsky. “On expansion of algebraic functions in power and Puiseux series I”. In: *Journal of Complexity* 2.4 (1986), pp. 271–294.

[10] M. Kazemi and M. Moreno Maza. “Detecting Singularities Using the PowerSeries Library”. In: *Proc. of MC 2019*. Springer, 2019, pp. 145–155.

[11] H. T. Kung and J. F. Traub. “All Algebraic Functions Can Be Computed Fast”. In: *Journal of the ACM* 25.2 (1978), pp. 245–260.

[12] A. A. Monforte and M. Kauers. “Formal Laurent series in several variables”. In: *Expositiones Mathematicae* 31.4 (2013), pp. 350–367.

[13] M. Moreno Maza and E. Postma. “Substituting Units into Multivariate Power Series”. In: *Proc. of MC 2021*. 2021.

[14] D. Mumford. *The red book of varieties and schemes*. 2nd. Vol. 1358. Lecture Notes in Mathematics. Springer, 1999.

[15] K. J. Nowak. “Some elementary proofs of Puiseux’s theorems”. In: *Univ. Iagel. Acta Math* 38 (2000), pp. 279–282.

[16] T. Sasaki and F. Kako. “Solving multivariate algebraic equation by Hensel construction”. In: *Japan Journal of Industrial and Applied Mathematics* 16.2 (1999), pp. 257–285.

[17] T. Sasaki and D. Inaba. “Enhancing the Extended Hensel Construction by Using Gröbner Bases”. In: *Proc. of CASC 2016*. Vol. 9890. Lecture Notes in Computer Science. Springer, 2016, pp. 457–472.

Thank You!

Basic Polynomial Algebra Subprograms

http://www.bpaslib.org/