Signature Gröbner bases algorithms over Tate algebras

Xavier Caruso[1] Tristan Vaccon[2] Thibaut Verron[3]

1. Université de Bordeaux, CNRS, Inria, Bordeaux, France
2. Université de Limoges, CNRS, XLIM, Limoges, France
3. Johannes Kepler University, Institute for Algebra, Linz, Austria

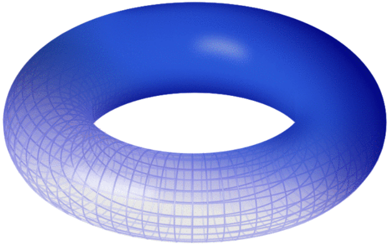
International Symposium on Symbolic and Algebraic Calculation 2020

Note: the present slides are images extracted from the presentation video.
For the best viewing experience, please watch the video!

GAGA [Serre, 1956]

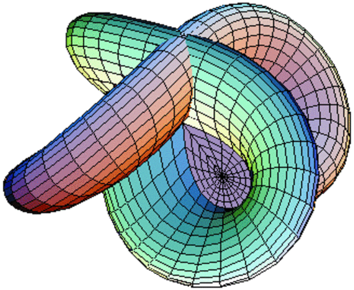
Algebraic Geometry

Multivariate polynomials



Analytic Geometry

Analytic functions



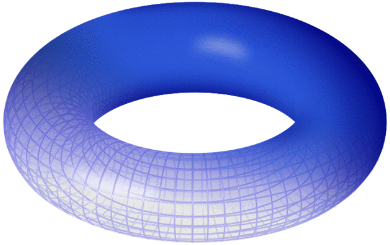
GAGA [Serre, 1956]

Algebraic Geometry

Multivariate polynomials

Analytic Geometry

Analytic functions:
Multivariate power series
convergent on an open ball

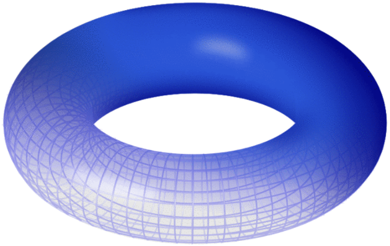


Analytic geometry
in p-adic setting

$\xleftrightarrow{???}$

Algebraic Geometry

Multivariate polynomials



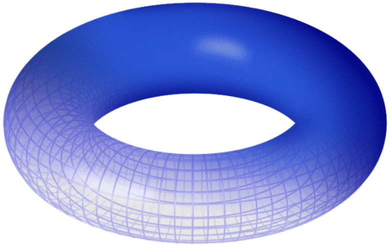
[Tate, 1962]

Rigid Geometry

Tate series:
Multivariate power series
convergent on a closed ball

Algebraic Geometry

Multivariate polynomials



GAGA [Serre, 1956]

Algebraic Geometry $\longleftrightarrow$ Analytic Geometry

Multivariate polynomials

Analytic functions:
Multivariate power series
convergent on an open ball

Effective techniques and software

Exact ideal arithmetic (intersection, union...):

Gröbner bases...

Rigid Geometry

Tate series:
Multivariate power series
convergent on a closed ball

[Tate, 1962]

Algebraic Geometry

Multivariate polynomials

Our work:

- Theory of Gröbner bases
- Buchberger's algorithm
- Signature-based algorithms

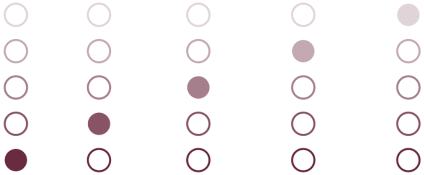
Effective techniques and software

Exact ideal arithmetic (intersection, uni

Gröbner bases...

Tate series: Multivariate power series with coefficients in a valued ring, convergent on a closed ball

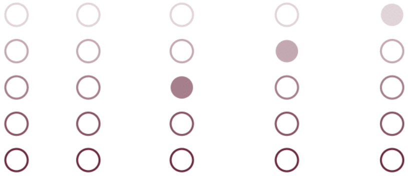
Valued ring: $\mathbb{Q}_p$, $\mathbb{Z}_p$, $\mathbb{C}[[X]]$, $\mathbb{C}((X))$...



$$a = a_0 + a_1\pi + a_2\pi^2 + a_3\pi^3 + a_4\pi^4 + \cdots$$

Tate series: Multivariate power series with coefficients in a valued ring, convergent on a closed ball

Valued ring: $\mathbb{Q}_p$, $\mathbb{Z}_p$, $\mathbb{C}[[X]]$, $\mathbb{C}((X))$...



$$a = \underbrace{0 + 0\pi}_{\text{Valuation}=2} + a_2\pi^2 + a_3\pi^3 + a_4\pi^4 + \cdots$$

Tate series: Multivariate power series with coefficients in a valued ring, convergent on a closed ball

Valued ring: $\mathbb{Q}_p$, $\mathbb{Z}_p$, $\mathbb{C}[[X]]$, $\mathbb{C}((X))$…

Valuation=2

$a \in K^o$

Tate series: Multivariate power series with coefficients in a valued ring, convergent on a closed ball

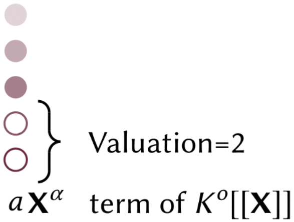
Valued ring: $\mathbb{Q}_p$, $\mathbb{Z}_p$, $\mathbb{C}[[X]]$, $\mathbb{C}((X))\ldots$

Valuation=2

$a\,X_1^{\alpha_1}\cdots X_n^{\alpha_n}$ term of $K^o[[\mathbf{X}]]$

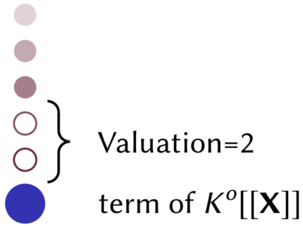
Tate series: Multivariate power series with coefficients in a valued ring, convergent on a closed ball

Valued ring: $\mathbb{Q}_p$, $\mathbb{Z}_p$, $\mathbb{C}[[X]]$, $\mathbb{C}((X))\dots$



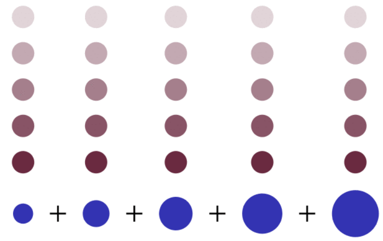
Tate series: Multivariate power series with coefficients in a valued ring, convergent on a closed ball

Valued ring: $\mathbb{Q}_p$, $\mathbb{Z}_p$, $\mathbb{C}[[X]]$, $\mathbb{C}((X))\dots$



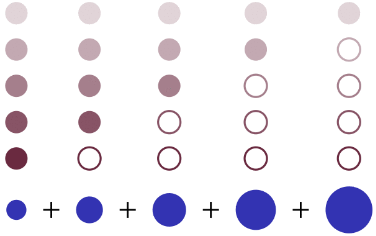
Tate series: Multivariate power series with coefficients in a valued ring, convergent on a closed ball

Valued ring: $\mathbb{Q}_p$, $\mathbb{Z}_p$, $\mathbb{C}[[X]]$, $\mathbb{C}((X))\dots$



Tate series: Multivariate power series with coefficients in a valued ring, convergent on a closed ball

Valued ring: $\mathbb{Q}_p$, $\mathbb{Z}_p$, $\mathbb{C}[[X]]$, $\mathbb{C}((X))\dots$

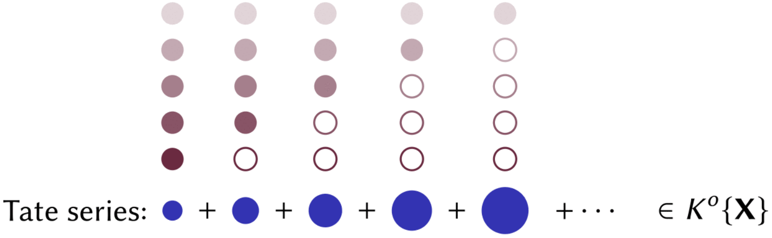


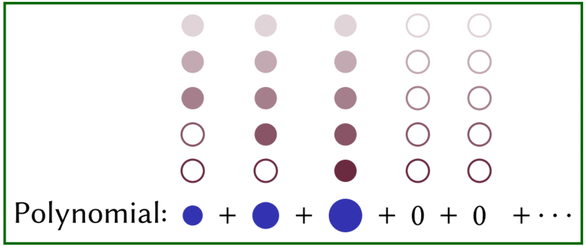
Tate series: $+ \cdots \quad \in K^{o}\{\mathbf{X}\}$

Convergence condition: the valuation goes to infinity

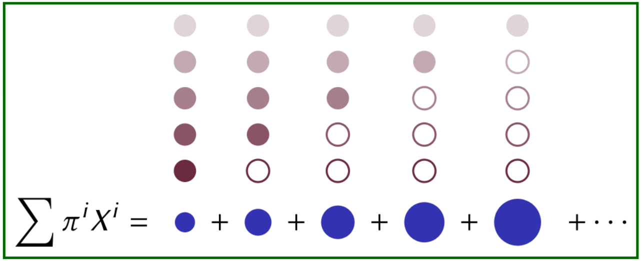
Tate series: Multivariate power series with coefficients in a valued ring, convergent on a closed ball

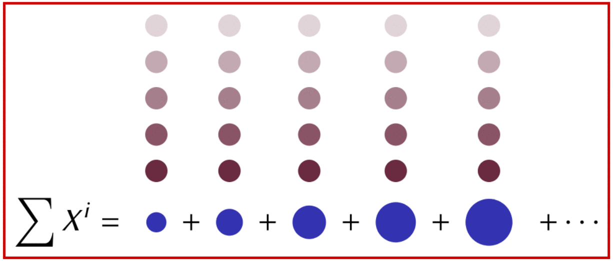
Valued ring: $\mathbb{Q}_p$, $\mathbb{Z}_p$, $\mathbb{C}[[X]]$, $\mathbb{C}((X))$...





Convergence condition: the valuation goes to infinity





Gröbner bases in finite precision:

- Need to work around error propagation
- Need to perform tests to zero to find leading terms

Gröbner bases in finite precision:

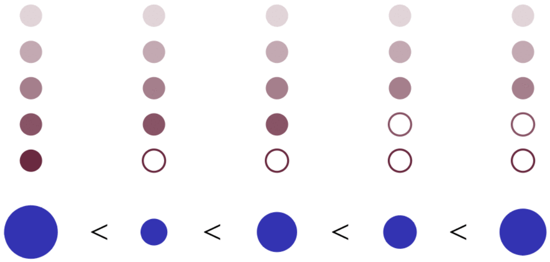
- No error propagation in valued rings
- Need to perform tests to zero to find leading terms

Gröbner bases in finite precision:

- No error propagation in valued rings
- Need to perform tests to zero to find leading terms

Solution: term ordering such that "close to zero" means small

- Order the terms with their coefficients
- First compare the valuations
- Then break ties with a monomial order

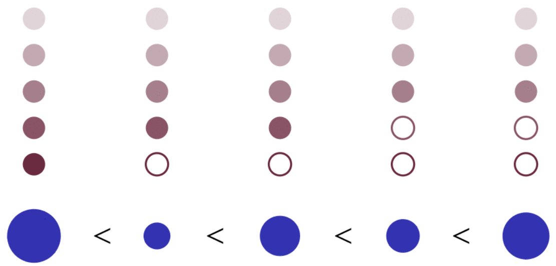


Gröbner bases in finite precision:

- No error propagation in valued rings
- Need to perform tests to zero to find leading terms

Solution: term ordering such that "close to zero" means small

- Order the terms with their coefficients
- First compare the valuations
- Then break ties with a monomial order



Properties:

- Terms with smaller valuation are larger
- Infinite reductions have increasing valuation
- Tate series have a leading term

Buchberger's algorithm

Input: F list of Tate series

Output: G Gröbner basis of the ideal $\langle F \rangle$

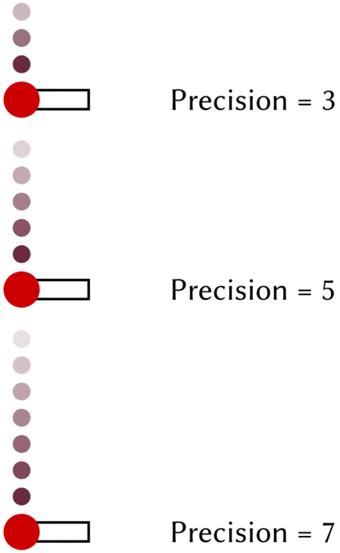
1. $G \leftarrow F$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(g, g') : g \neq g' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad h \leftarrow$ an element of $\mathcal{P}$
5. $\quad h \leftarrow \text{Reduce}(h, G)$
6. $\quad$ If $h \neq 0$:
7. $\quad\quad$ Add h to G
8. $\quad\quad$ Add to $\mathcal{P}$ all $\text{S-Pol}(g, h)$ for $g \in G$
9. Return G

Buchberger's algorithm

Input: F list of Tate series

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow F$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(g, g') : g \neq g' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad h \leftarrow$ an element of $\mathcal{P}$
5. $\quad h \leftarrow \text{Reduce}(h, G)$
6. $\quad$ If $h \neq 0$:
7. $\qquad$ Add h to G
8. $\qquad$ Add to $\mathcal{P}$ all $\text{S-Pol}(g, h)$ for $g \in G$
9. Return G



Bottleneck: reductions to zero

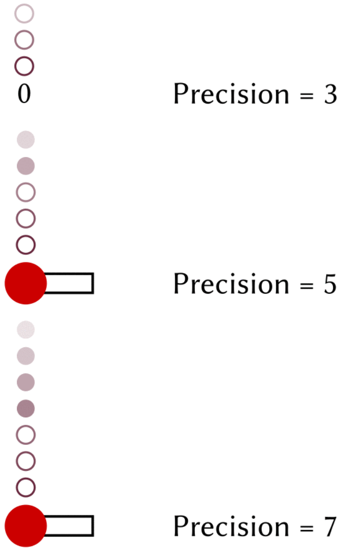
Increasing the precision makes it worse!

Buchberger's algorithm

Input: F list of Tate series

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow F$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(g, g') : g \neq g' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad h \leftarrow$ an element of $\mathcal{P}$
5. $\quad h \leftarrow \text{Reduce}(h, G)$
6. $\quad$ If $h \neq 0$:
7. $\quad\quad$ Add h to G
8. $\quad\quad$ Add to $\mathcal{P}$ all $\text{S-Pol}(g, h)$ for $g \in G$
9. Return G



Bottleneck: reductions to zero

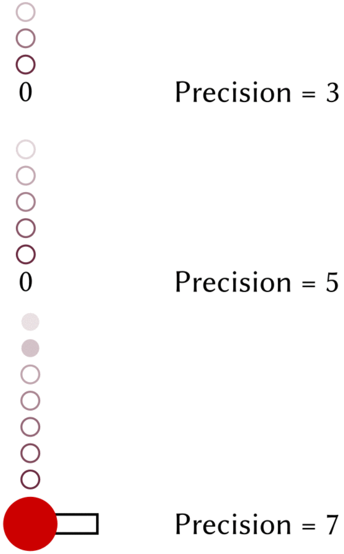
Increasing the precision makes it worse!

Buchberger's algorithm

Input: F list of Tate series

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow F$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(g, g') : g \neq g' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad h \leftarrow$ an element of $\mathcal{P}$
5. $\quad h \leftarrow \text{Reduce}(h, G)$
6. $\quad$ If $h \neq 0$:
7. $\qquad$ Add h to G
8. $\qquad$ Add to $\mathcal{P}$ all S-Pol(g, h) for $g \in G$
9. Return G



Bottleneck: reductions to zero

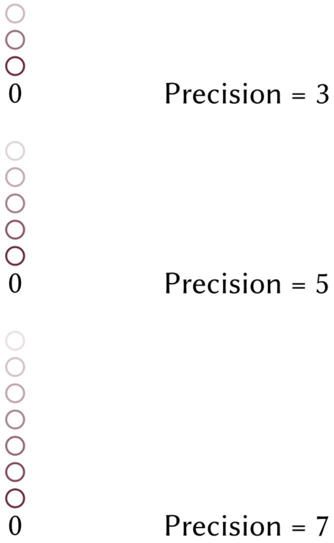
Increasing the precision makes it worse!

Buchberger's algorithm

Input: F list of Tate series

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow F$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(g, g') : g \neq g' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad h \leftarrow$ an element of $\mathcal{P}$
5. $\quad h \leftarrow \text{Reduce}(h, G)$
6. $\quad$ If $h \neq 0$:
7. $\qquad$ Add h to G
8. $\qquad$ Add to $\mathcal{P}$ all $\text{S-Pol}(g, h)$ for $g \in G$
9. Return G



Bottleneck: reductions to zero

Increasing the precision makes it worse!

Some reductions to zero are predictable, how to detect them?

Some reductions to zero are predictable, how to detect them?

$$
\begin{array}{rl}
p & \in \mathbb{C}[\mathbf{X}] \\
+ & \\
q & \in \mathbb{C}[\mathbf{X}] \\
\shortparallel & \\
p + q & \in \mathbb{C}[\mathbf{X}] \\
{\color{red}0} &
\end{array}
$$

Some reductions to zero are predictable, how to detect them?

$$
\begin{array}{ccccccccccc}
p & = & p_1 & f_1 + & p_2 & f_2 + \cdots + & p_m & f_m \in \mathbb{C}[\mathbf{X}] \\
+ & & & & & & & \\
q & = & q_1 & f_1 + & q_2 & f_2 + \cdots + & q_m & f_m \in \mathbb{C}[\mathbf{X}] \\
\shortparallel & & & & & & & \\
p + q & = & (p_1 + q_1) & f_1 + & (p_2 + q_2) & f_2 + \cdots + & (p_m + q_m) & f_m \in \mathbb{C}[\mathbf{X}] \\
0 & & f_2 & & -f_1 & & 0 & f_2 f_1 - f_1 f_2 = 0
\end{array}
$$

Some reductions to zero are predictable, how to detect them?

Idea 1: keep track of the module representation of the elements [Möller, Mora, Traverso, 1992]

$$
\begin{array}{ccccccccc}
p & & p_1 \mathbf{e}_1 & + & p_2 \mathbf{e}_2 & + \cdots + & p_m \mathbf{e}_m & \in \mathbb{C}[\mathbf{X}]^m \\
+ & & & & & & & \\
q & & q_1 \mathbf{e}_1 & + & q_2 \mathbf{e}_2 & + \cdots + & q_m \mathbf{e}_m & \in \mathbb{C}[\mathbf{X}]^m \\
\parallel & & & & & & & \\
p+q & & (p_1+q_1)\mathbf{e}_1 & + & (p_2+q_2)\mathbf{e}_2 & + \cdots + & (p_m+q_m)\mathbf{e}_m & \in \mathbb{C}[\mathbf{X}]^m \\
0 & & f_2 & & -f_1 & & 0 & f_2\mathbf{e}_1 - f_1\mathbf{e}_2 : \text{known syzygy}
\end{array}
$$

Some reductions to zero are predictable, how to detect them?

Idea 1: keep track of the module representation of the elements [Möller, Mora, Traverso, 1992]

$$
\begin{array}{ccccccccc}
p & & p_1 \mathbf{e}_1 & + & p_2 \mathbf{e}_2 & + \cdots + & p_m \mathbf{e}_m & \in \mathbb{C}[\mathbf{X}]^m & \\
+ & & & & & & & & \\
q & & q_1 \mathbf{e}_1 & + & q_2 \mathbf{e}_2 & + \cdots + & q_m \mathbf{e}_m & \in \mathbb{C}[\mathbf{X}]^m & \\
\shortparallel & & & & & & & & \\
p+q & & (p_1+q_1)\mathbf{e}_1 & + & (p_2+q_2)\mathbf{e}_2 & + \cdots + & (p_m+q_m)\mathbf{e}_m & \in \mathbb{C}[\mathbf{X}]^m & \\
0 & & f_2 & & -f_1 & & 0 & & f_2\mathbf{e}_1 - f_1\mathbf{e}_2 : \text{known syzygy} \\
\uparrow & & \uparrow & & \uparrow & & \uparrow & &
\end{array}
$$

Cost: $m+1$ polynomial additions

Some reductions to zero are predictable, how to detect them?

Idea 1: keep track of the module representation of the elements [Möller, Mora, Traverso, 1992]

$$
\begin{array}{ccccccccc}
p & = & p_1 & f_1 + & p_2 & f_2 + \cdots + & p_m & f_m \in \mathbb{C}[\mathbf{X}] \\
+ & & & & & & & \\
q & = & q_1 & f_1 + & q_2 & f_2 + \cdots + & q_m & f_m \in \mathbb{C}[\mathbf{X}] \\
\| & & & & & & & \\
p + q & = & (p_1 + q_1) & f_1 + & (p_2 + q_2) & f_2 + \cdots + & (p_m + q_m) & f_m \in \mathbb{C}[\mathbf{X}] \\
0 & & f_2 & & -f_1 & & 0 & f_2 f_1 - f_1 f_2 = 0
\end{array}
$$

Cost: 1 polynomial addition

Some reductions to zero are predictable, how to detect them?

Idea 1: keep track of the module representation of the elements [Möller, Mora, Traverso, 1992]

$$
\begin{array}{ccc}
p & p_1 \mathbf{e}_1 + p_2 \mathbf{e}_2 + \cdots + p_m \mathbf{e}_m & \in \mathbb{C}[\mathbf{X}]^m \\
+ & & \\
q & q_1 \mathbf{e}_1 + q_2 \mathbf{e}_2 + \cdots + q_m \mathbf{e}_m & \in \mathbb{C}[\mathbf{X}]^m \\
\shortparallel & & \\
p+q & (p_1+q_1)\mathbf{e}_1 + (p_2+q_2)\mathbf{e}_2 + \cdots + (p_m+q_m)\mathbf{e}_m & \in \mathbb{C}[\mathbf{X}]^m \\
0 & f_2 \qquad -f_1 \qquad\qquad 0 & f_2\mathbf{e}_1 - f_1\mathbf{e}_2 : \text{known syzygy}
\end{array}
$$

Cost: $m+1$ polynomial additions

Some reductions to zero are predictable, how to detect them?

Idea 1: keep track of the module representation of the elements [Möller, Mora, Traverso, 1992]

Idea 2: only keep some terms of the module elements

$$
\begin{array}{ccl}
p & & \mathrm{LT}(p_1)\,\mathbf{e}_1 + \mathrm{LT}(p_2)\,\mathbf{e}_2 + \cdots + \mathrm{LT}(p_m)\,\mathbf{e}_m + \text{ smaller terms} \in \mathbb{C}[\mathbf{X}]^m \\
+ & & \\
q & & \mathrm{LT}(q_1)\,\mathbf{e}_1 + \mathrm{LT}(q_2)\,\mathbf{e}_2 + \cdots + \mathrm{LT}(q_m)\,\mathbf{e}_m + \text{ smaller terms} \in \mathbb{C}[\mathbf{X}]^m \\
\shortparallel & & \\
p+q & & \mathrm{LT}(p_1+q_1)\mathbf{e}_1 + \mathrm{LT}(p_2+q_2)\mathbf{e}_2 + \cdots + \mathrm{LT}(p_m+q_m)\mathbf{e}_m + \text{ smaller terms} \in \mathbb{C}[\mathbf{X}]^m \\
0 & & \mathrm{LT}(f_2) \qquad -\mathrm{LT}(f_1) \qquad\qquad 0 \qquad \mathrm{LT}(f_2)\mathbf{e}_1 - \mathrm{LT}(f_1)\mathbf{e}_2 :
\end{array}
$$

LT of a known syzygy

Cost: 1 polynomial addition and m term comparisons : $\mathrm{LT}(p+q) = \begin{cases} \mathrm{LT}(p) & \text{if } \mathrm{LT}(p) > \mathrm{LT}(q) \\ \mathrm{LT}(q) & \text{if } \mathrm{LT}(p) < \mathrm{LT}(q) \\ ??? & \text{otherwise} \end{cases}$

Some reductions to zero are predictable, how to detect them?

Idea 1: keep track of the module representation of the elements [Möller, Mora, Traverso, 1992]

Idea 2: only keep some terms of the module elements

$$
\begin{array}{ccl}
p & \mathrm{LT}(p_i) & \mathbf{e}_i + \text{ smaller terms} \in \mathbb{C}[\mathbf{X}]^m \\
+ & & \\
q & \mathrm{LT}(q_i) & \mathbf{e}_i + \text{ smaller terms} \in \mathbb{C}[\mathbf{X}]^m \\
\parallel & & \\
p+q & \mathrm{LT}(p_i+q_i) & \mathbf{e}_i + \text{ smaller terms} \in \mathbb{C}[\mathbf{X}]^m \\
0 & \mathrm{LT}(f_2) & \qquad \mathrm{LT}(f_2)\mathbf{e}_1 :
\end{array}
$$

LT of a known syzygy

Cost: 1 polynomial addition and 1 term comparison : $\mathrm{LT}(p+q) = \begin{cases} \mathrm{LT}(p) & \text{if } \mathrm{LT}(p) > \mathrm{LT}(q) \\ \mathrm{LT}(q) & \text{if } \mathrm{LT}(p) < \mathrm{LT}(q) \\ ??? & \text{otherwise} \end{cases}$

Some reductions to zero are predictable, how to detect them?

Idea 1: keep track of the module representation of the elements [Möller, Mora, Traverso, 1992]

Idea 2: only keep some terms of the module elements

Idea 3: skip operations which cannot be done [Faugère, 2002]

$$
\begin{array}{ccl}
p & \mathrm{LT}(p_i) & \mathbf{e}_i + \text{ smaller terms} \in \mathbb{C}[\mathbf{X}]^m \\
+ & & \\
q & \mathrm{LT}(q_i) & \mathbf{e}_i + \text{ smaller terms} \in \mathbb{C}[\mathbf{X}]^m \\
\shortparallel & & \\
p+q & \mathrm{LT}(p_i+q_i) & \mathbf{e}_i + \text{ smaller terms} \in \mathbb{C}[\mathbf{X}]^m \\
0 & \mathrm{LT}(f_2) & \qquad \mathrm{LT}(f_2)\mathbf{e}_1 : \\
\uparrow & \uparrow & \qquad \text{LT of a known syzygy}
\end{array}
$$

Cost: 1 polynomial addition and 1 term comparison : $\mathrm{LT}(p+q) = \begin{cases} \mathrm{LT}(p) & \text{if } \mathrm{LT}(p) > \mathrm{LT}(q) \\ \mathrm{LT}(q) & \text{if } \mathrm{LT}(p) < \mathrm{LT}(q) \\ \cancel{???} & \cancel{\text{otherwise}} \end{cases}$

Some reductions to zero are predictable, how to detect them?

Idea 1: keep track of the module representation of the elements [Möller, Mora, Traverso, 1992]

Idea 2: only keep some terms of the module elements

Idea 3: skip operations which cannot be done [Faugère, 2002]

- Consider pairs instead of polynomials:

$$(s, p)$$

$$p = \sum p_i f_i \in \mathbb{C}[\mathbf{X}]$$

Signature: $s = \mathrm{LT}(\sum p_i \mathbf{e}_i)$ term of $\mathbb{C}[\mathbf{X}]^m$

- Only allow regular operations:

$$(s, f) + (t, g) = \begin{cases} (s, f+g) & \text{if } s > t \\ (t, f+g) & \text{if } s < t \\ \text{non-regular otherwise} \end{cases}$$

Some reductions to zero are predictable, how to detect them?

Idea 1: keep track of the module representation of the elements [Möller, Mora, Traverso, 1992]

Idea 2: only keep some terms of the module elements

Idea 3: skip operations which cannot be done [Faugère, 2002]

- Consider pairs instead of polynomials: Survey: [Eder, Faugère, 2017]

$$(s, p)$$

$$\longrightarrow p = \sum p_i f_i \in \mathbb{C}[\mathbf{X}]$$

$\longrightarrow$ Signature: $s = \mathrm{LT}(\sum p_i \mathbf{e}_i)$ term of $\mathbb{C}[\mathbf{X}]^m$

- Only allow regular operations:

$$(s, f) + (t, g) = \begin{cases} (s, f+g) & \text{if } s > t \\ (t, f+g) & \text{if } s < t \\ \text{non-regular otherwise} \end{cases}$$

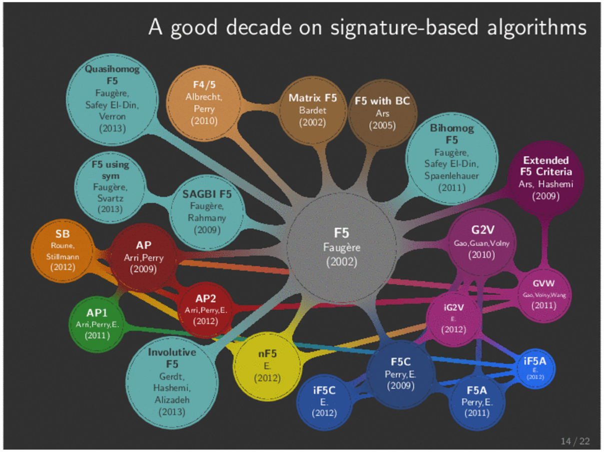


Image: Christian Eder, 2013

Buchberger's algorithm

Input: F list of Tate series

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow F$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(g, g') : g \neq g' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad h \leftarrow$ an element of $\mathcal{P}$
5. $\quad h \leftarrow \text{Reduce}(h, G)$
6. $\quad$ If $h \neq 0$:
7. $\quad\quad$ Add h to G
8. $\quad\quad$ Add to $\mathcal{P}$ all $\text{S-Pol}(g, h)$ for $g \in G$
9. Return G

Signature-based algorithm

Input: $F = \{f_1, \dots, f_n\}$ list of polynomials

Output: G Gröbner basis of the ideal $\langle F \rangle$

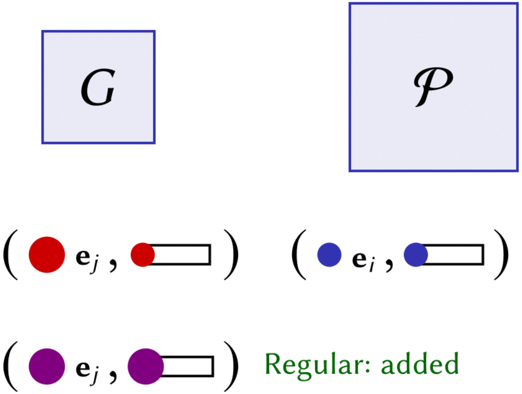
1. $G \leftarrow \{(\mathbf{e}_i, f_i) : f_i \in F\}$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(p, p') : p \neq p' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad (s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
5. $\quad h \leftarrow$Regular-Reduce$((s, h), G)$
6. $\quad$ If $h \neq 0$:
7. $\qquad$ Add (s, h) to G
8. $\qquad$ Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
9. Return G

Signature-based algorithm

Input: $F = \{f_1, \dots, f_n\}$ list of polynomials

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow \{(\mathbf{e}_i, f_i) : f_i \in F\}$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(p, p') : p \neq p' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad (s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
5. $\quad h \leftarrow$ Regular-Reduce$((s, h), G)$
6. $\quad$ If $h \neq 0$:
7. $\qquad$ Add (s, h) to G
8. $\qquad$ Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
9. Return G

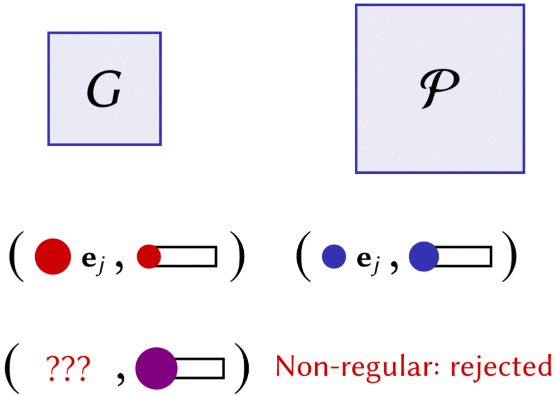


Signature-based algorithm

Input: $F = \{f_1, \ldots, f_n\}$ list of polynomials

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow \{(\mathbf{e}_i, f_i) : f_i \in F\}$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(p, p') : p \neq p' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad (s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
5. $\quad h \leftarrow$ Regular-Reduce$((s, h), G)$
6. $\quad$ If $h \neq 0$:
7. $\quad\quad$ Add (s, h) to G
8. $\quad\quad$ Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
9. Return G

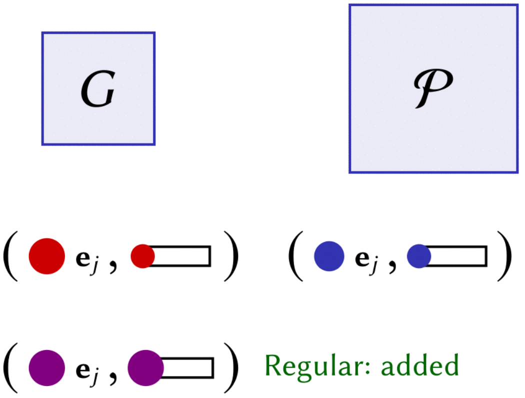


Signature-based algorithm

Input: $F = \{f_1, \ldots, f_n\}$ list of polynomials

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow \{(\mathbf{e}_i, f_i) : f_i \in F\}$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(p, p') : p \neq p' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad (s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
5. $\quad h \leftarrow$ Regular-Reduce$((s, h), G)$
6. $\quad$ If $h \neq 0$:
7. $\quad\quad$ Add (s, h) to G
8. $\quad\quad$ Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
9. Return G

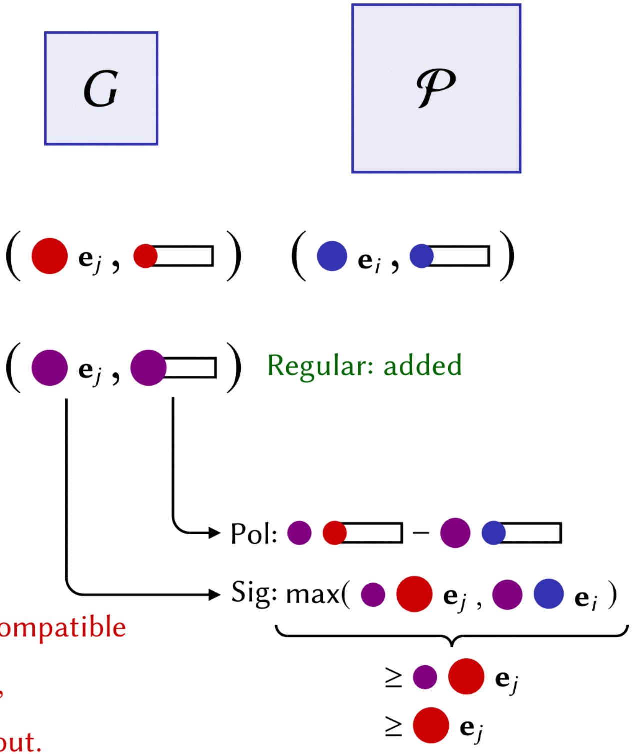


Signature-based algorithm

Input: $F = \{f_1, \dots, f_n\}$ list of polynomials

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow \{(\mathbf{e}_i, f_i) : f_i \in F\}$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(p, p') : p \neq p' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad (s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
5. $\quad h \leftarrow$ Regular-Reduce$((s, h), G)$
6. $\quad$ If $h \neq 0$:
7. $\qquad$ Add (s, h) to G
8. $\qquad$ Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
9. Return G



If the signature ordering is compatible

with the monomial ordering,

signatures increase throughout.

Signature-based algorithm

Input: $F = \{f_1, \dots, f_n\}$ list of polynomials

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow \{(\mathbf{e}_i, f_i) : f_i \in F\}$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(p, p') : p \neq p' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad (s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
5. $\quad h \leftarrow$ Regular-Reduce$((s, h), G)$
6. $\quad$ If $h \neq 0$:
7. $\quad\quad$ Add (s, h) to G
8. $\quad\quad$ Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
9. Return G

Signature-based algorithm

Input: $F = \{f_1, \ldots, f_n\}$ list of polynomials

Output: G Gröbner basis of the ideal $\langle F \rangle$

$G \leftarrow \{(\mathbf{e}_i, f_i) : f_i \in F\}$

$\mathcal{P} \leftarrow \{\text{S-pol}(p, p') : p \neq p' \in G\}$

While $\mathcal{P} \neq \emptyset$:

  $(s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s

  $h \leftarrow$ Regular-Reduce$((s, h), G)$

  If $h \neq 0$:

    Add (s, h) to G

    Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$

Return G

Choice of a signature ordering:

- Compatible with the monomial ordering
- Gives a loop invariant for the algorithm

Easiest example: PoTe: Position over Term

$$\bullet\,\mathbf{e}_1 < \bullet\,\mathbf{e}_2 < \bullet\,\mathbf{e}_2 < \bullet\,\mathbf{e}_3 < \bullet\,\mathbf{e}_3$$

Signature-based algorithm

Input: $F = \{f_1, \dots, f_n\}$ list of polynomials

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow \{(\mathbf{e}_i, f_i) : f_i \in F\}$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(p, p') : p \neq p' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad (s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
5. $\quad h \leftarrow$ Regular-Reduce$((s, h), G)$
6. $\quad$ If $h \neq 0$:
7. $\quad\quad$ Add (s, h) to G
8. $\quad\quad$ Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
9. Return G

Choice of a signature ordering:

- Compatible with the monomial ordering
- Gives a loop invariant for the algorithm

Easiest example: PoTe: Position over Term

$$\bullet\, \mathbf{e}_1 < \bullet\, \mathbf{e}_2 < \large\bullet\, \mathbf{e}_2 < \bullet\, \mathbf{e}_3 < \large\bullet\, \mathbf{e}_3$$

Current signature: $\bullet\, \mathbf{e}_i$

Current polynomial: $\bullet\, f_i$ + sum of previous

Current basis: basis of $\{f_1, \dots, f_{i-1}\}$

Incremental signature-based algorithm [G2V, 2010]

Input: $F = \{f_1, \dots, f_n\}$ list of polynomials

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow \emptyset$
2. For i from 1 to n:
3. Add $(\mathbf{e}_i, f_i)$ to G
4. Add to $\mathcal{P}$ all S-Pol$(p, (s, h))$ for $p \in G$
5. While $\mathcal{P} \neq \emptyset$:
6. $(s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
7. $h \leftarrow$Regular-Reduce$((s, h), G)$
8. If $h \neq 0$:
9. Add (s, h) to G
10. Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
11. Return G

Choice of a signature ordering:

- Compatible with the monomial ordering
- Gives a loop invariant for the algorithm

Easiest example: PoTe: Position over Term

$$\bullet\, \mathbf{e}_1 < \bullet\, \mathbf{e}_2 < \bullet\, \mathbf{e}_2 < \bullet\, \mathbf{e}_3 < \bullet\, \mathbf{e}_3$$

Current signature: $\bullet\, \mathbf{e}_i$

Current polynomial: $\bullet\, f_i$ + sum of previous

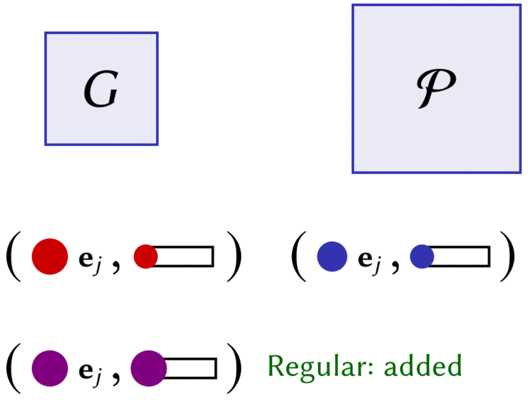
Current basis: basis of $\{f_1, \dots, f_{i-1}\}$

Signature-based algorithm

Input: $F = \{f_1, \dots, f_n\}$ list of polynomials

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow \{(\mathbf{e}_i, f_i) : f_i \in F\}$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(p, p') : p \neq p' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad (s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
5. $\quad h \leftarrow$ Regular-Reduce$((s, h), G)$
6. $\quad$ If $h \neq 0$:
7. $\quad\quad$ Add (s, h) to G
8. $\quad\quad$ Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
9. Return G

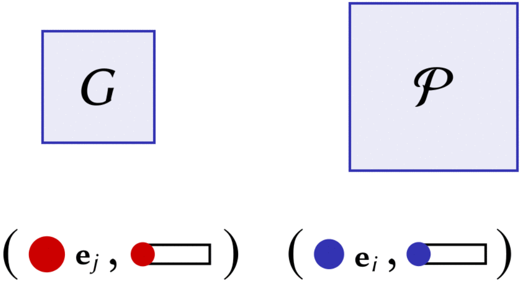


Signature-based algorithm

Input: $F = \{f_1, \dots, f_n\}$ list of polynomials

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow \{(\mathbf{e}_i, f_i) : f_i \in F\}$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(p, p') : p \neq p' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad (s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
5. $\quad h \leftarrow$ Regular-Reduce$((s, h), G)$
6. $\quad$ If $h \neq 0$:
7. $\quad\quad$ Add (s, h) to G
8. $\quad\quad$ Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
9. Return G

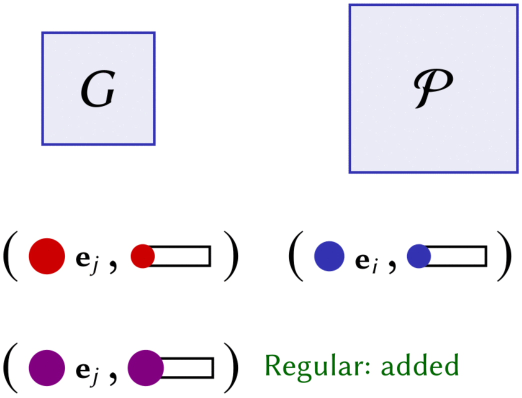


Signature-based algorithm

Input: $F = \{f_1, \dots, f_n\}$ list of polynomials

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow \{(\mathbf{e}_i, f_i) : f_i \in F\}$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(p, p') : p \neq p' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad (s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
5. $\quad h \leftarrow$ Regular-Reduce$((s, h), G)$
6. $\quad$ If $h \neq 0$:
7. $\qquad$ Add (s, h) to G
8. $\qquad$ Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
9. Return G

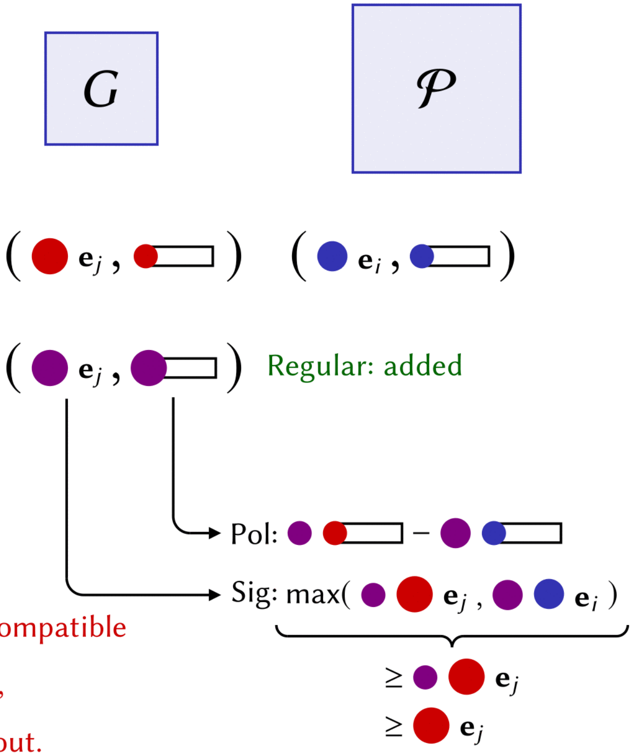


Signature-based algorithm

Input: $F = \{f_1, \dots, f_n\}$ list of polynomials

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow \{(\mathbf{e}_i, f_i) : f_i \in F\}$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(p, p') : p \neq p' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad (s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
5. $\quad h \leftarrow$ Regular-Reduce$((s, h), G)$
6. $\quad$ If $h \neq 0$:
7. $\quad\quad$ Add (s, h) to G
8. $\quad\quad$ Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
9. Return G



If the signature ordering is compatible

with the monomial ordering,

signatures increase throughout.

Signature-based algorithm

Input: $F = \{f_1, \dots, f_n\}$ list of polynomials

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow \{(\mathbf{e}_i, f_i) : f_i \in F\}$
2. $\mathcal{P} \leftarrow \{\text{S-pol}(p, p') : p \neq p' \in G\}$
3. While $\mathcal{P} \neq \emptyset$:
4. $\quad (s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
5. $\quad h \leftarrow$ Regular-Reduce$((s, h), G)$
6. $\quad$ If $h \neq 0$:
7. $\quad\quad$ Add (s, h) to G
8. $\quad\quad$ Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
9. Return G

Signature-based algorithm

Input: $F = \{f_1, \dots, f_n\}$ list of polynomials

Output: G Gröbner basis of the ideal $\langle F \rangle$

$G \leftarrow \{(\mathbf{e}_i, f_i) : f_i \in F\}$
$\mathcal{P} \leftarrow \{\text{S-pol}(p, p') : p \neq p' \in G\}$
While $\mathcal{P} \neq \emptyset$:
    $(s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
    $h \leftarrow$ Regular-Reduce$((s, h), G)$
    If $h \neq 0$:
        Add (s, h) to G
        Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
Return G

Choice of a signature ordering:

- Compatible with the monomial ordering
- Gives a loop invariant for the algorithm

Easiest example: PoTe: Position over Term

$$\bullet\, \mathbf{e}_1 < \bullet\, \mathbf{e}_2 < \bullet\, \mathbf{e}_2 < \bullet\, \mathbf{e}_3 < \bullet\, \mathbf{e}_3$$

Incremental signature-based algorithm [G2V, 2010]

Input: $F = \{f_1, \dots, f_n\}$ list of polynomials

Output: G Gröbner basis of the ideal $\langle F \rangle$

$G \leftarrow \emptyset$
For i from 1 to n:
  Add $(\mathbf{e}_i, f_i)$ to G
  Add to $\mathcal{P}$ all S-Pol$(p, (s, h))$ for $p \in G$
  While $\mathcal{P} \neq \emptyset$:
    $(s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
    $h \leftarrow$Regular-Reduce$((s, h), G)$
    If $h \neq 0$:
      Add (s, h) to G
      Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
Return G

Choice of a signature ordering:

- Compatible with the monomial ordering
- Gives a loop invariant for the algorithm

Easiest example: PoTe: Position over Term

$$\bullet\, \mathbf{e}_1 < \bullet\, \mathbf{e}_2 < \bullet\, \mathbf{e}_2 < \bullet\, \mathbf{e}_3 < \bullet\, \mathbf{e}_3$$

Current signature: $\bullet\, \mathbf{e}_i$

Current polynomial: $\bullet\, f_i$ + sum of previous

Current basis: basis of $\{f_1, \dots, f_{i-1}\}$

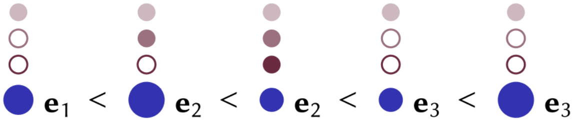
Incremental signature-based algorithm “PoTe”

Input: $F = \{f_1, \dots, f_n\}$ list of Tate series

Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow \emptyset$
2. For i from 1 to n:
3.   Add $(\mathbf{e}_i, f_i)$ to G
4.   Add to $\mathcal{P}$ all S-Pol$(p, (s, h))$ for $p \in G$
5.   While $\mathcal{P} \neq \emptyset$:
6. $(s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
7. $h \leftarrow$Regular-Reduce$((s, h), G)$
8.   If $h \neq 0$:
9.   Add (s, h) to G
10.   Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
11. Return G

Signature ordering 1: Position over Term



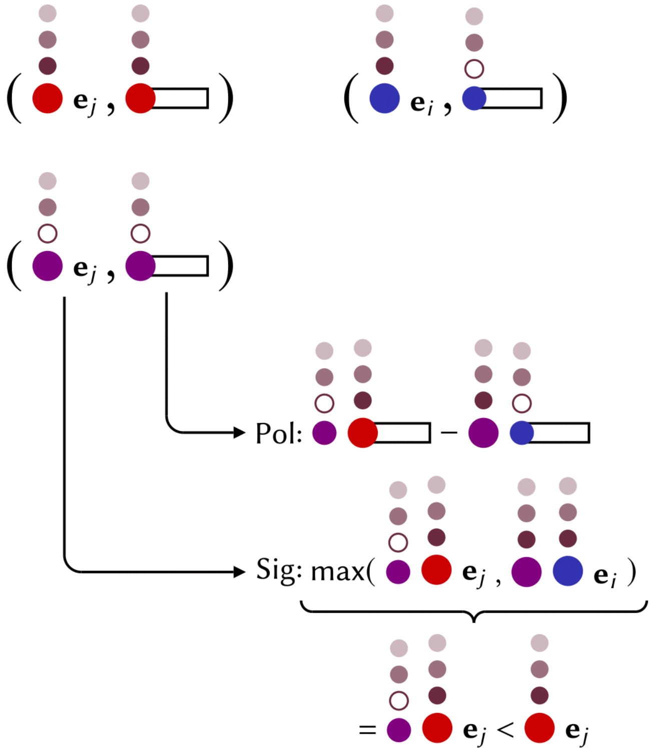
Incremental signature-based algorithm “PoTe”

Input: $F = \{f_1, \dots, f_n\}$ list of Tate series

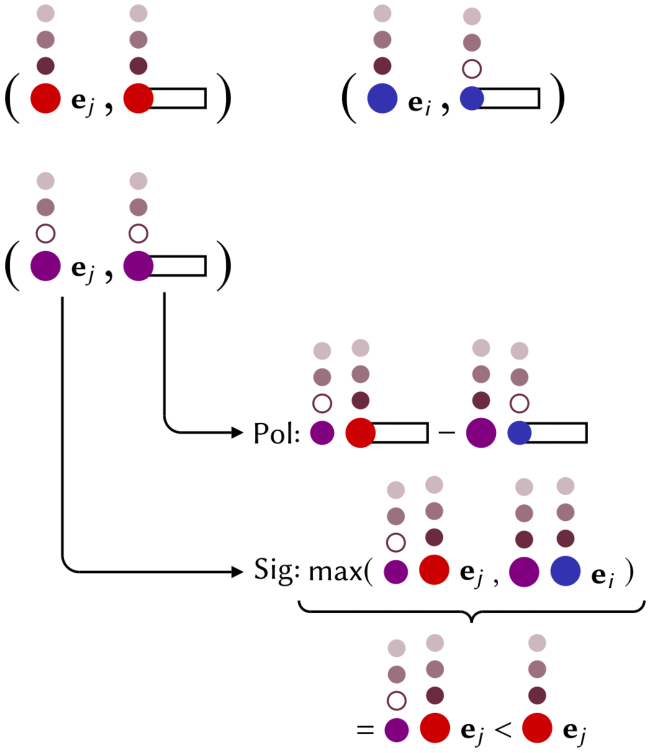
Output: G Gröbner basis of the ideal $\langle F \rangle$

1. $G \leftarrow \emptyset$
2. For i from 1 to n:
3. Add $(\mathbf{e}_i, f_i)$ to G
4. Add to $\mathcal{P}$ all S-Pol$(p, (s, h))$ for $p \in G$
5. While $\mathcal{P} \neq \emptyset$:
6. $(s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
7. $h \leftarrow$Regular-Reduce$((s, h), G)$
8. If $h \neq 0$:
9. Add (s, h) to G
10. Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
11. Return G

But signatures can decrease!



But signatures can decrease!



What is the difference with the polynomial case?

- The polynomial monomial order is global : $1 \leq$
- Most signature-based algorithms require a global order
- There are also local orders : ≤ 1
- and signature-based algorithms for that case

  [Lu et.al. 2018]

- Our order is mixed : $< 1 <$
- The local proof can be adapted

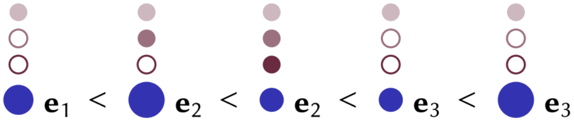
Incremental signature-based algorithm “PoTe”

Input: $F = \{f_1, \dots, f_n\}$ list of Tate series

Output: G Gröbner basis of the ideal $\langle F \rangle$

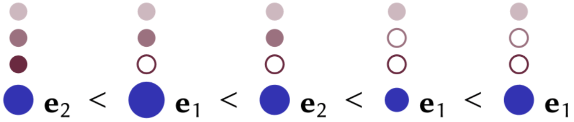
1. $G \leftarrow \emptyset$
2. For i from 1 to n:
3.   Add $(\mathbf{e}_i, f_i)$ to G
4.   Add to $\mathcal{P}$ all S-Pol$(p, (s, h))$ for $p \in G$
5.   While $\mathcal{P} \neq \emptyset$:
6. $(s, h) \leftarrow$ element of $\mathcal{P}$ with minimal signature s
7. $h \leftarrow$Regular-Reduce$((s, h), G)$
8.   If $h \neq 0$:
9.   Add (s, h) to G
10.   Add to $\mathcal{P}$ all regular S-Pol$(p, (s, h))$ for $p \in G$
11. Return G

Signature ordering 1: Position over Term



Signature ordering 2: VaPoTe

Increasing Valuation over Position over Term



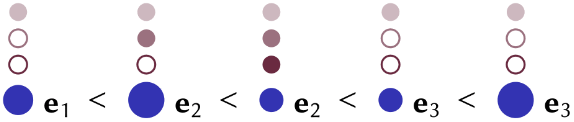
Incremental signature-based algorithm "VaPoTe"

Input: $F = \{f_1, \dots, f_n\}$ list of Tate series

Output: G Gröbner basis of the ideal $\langle F \rangle$

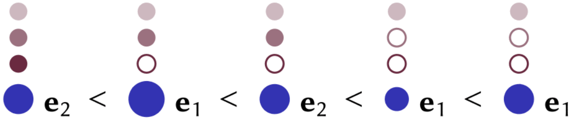
1. (Initialization), $Q \leftarrow \{(\mathbf{e}_i, f_i)\}$
2. For v from 0 to ∞:
3. For each element with valuation v in Q
4. (Update the basis like in PoTe)
5. $\dots$
6. If $\mathrm{val}(h) > v$:
7. Add h to Q
8. Else:
9. Update $\mathcal{P}$
10. Return G

Signature ordering 1: Position over Term



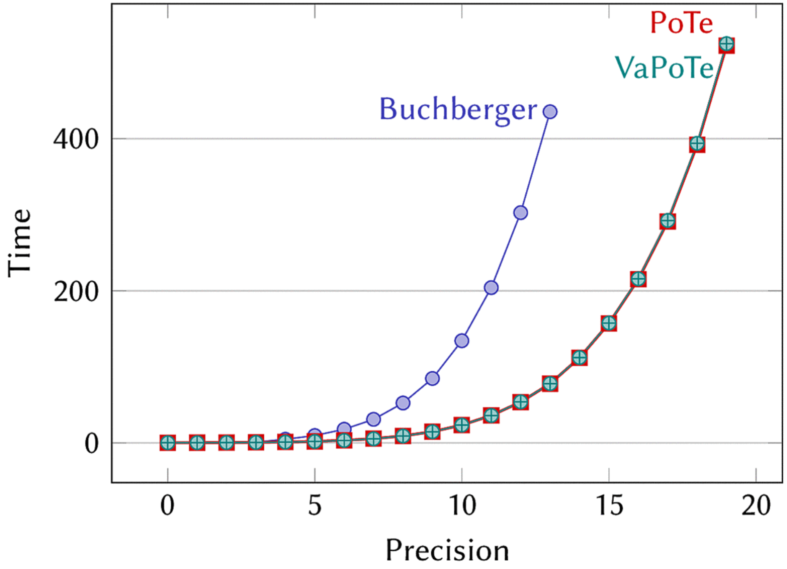
Signature ordering 2: VaPoTe

Increasing Valuation over Position over Term



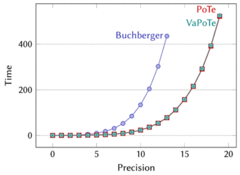
Conclusion

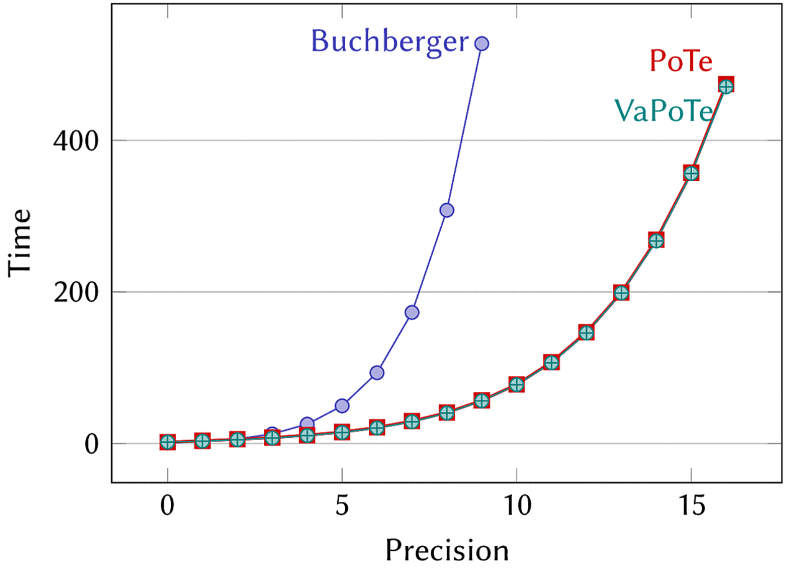
- Two algorithms: PoTe and VaPote
- Incremental, signature-based
- Generically equivalent
- Usually faster than Buchberger
- Distributed with SageMath 9.1



Conclusion

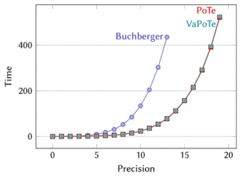
- Two algorithms: PoTe and VaPote
- Incremental, signature-based
- Generically equivalent
- Usually faster than Buchberger
- Distributed with SageMath 9.1

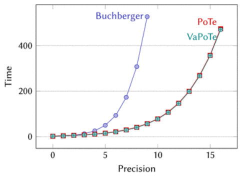




Conclusion

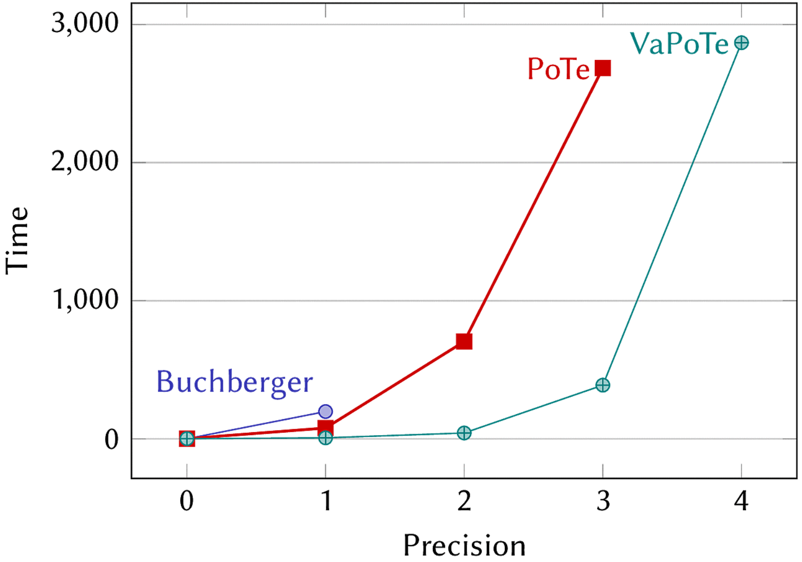
- Two algorithms: PoTe and VaPote
- Incremental, signature-based
- Generically equivalent
- Usually faster than Buchberger
- Distributed with SageMath 9.1





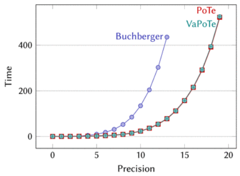
Future work

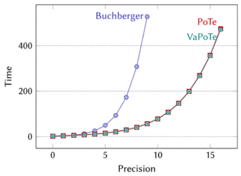
- Understand non-generic performance differences

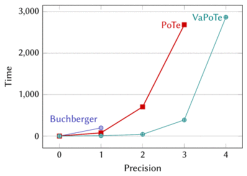


Conclusion

- Two algorithms: PoTe and VaPote
- Incremental, signature-based
- Generically equivalent
- Usually faster than Buchberger
- Distributed with SageMath 9.1

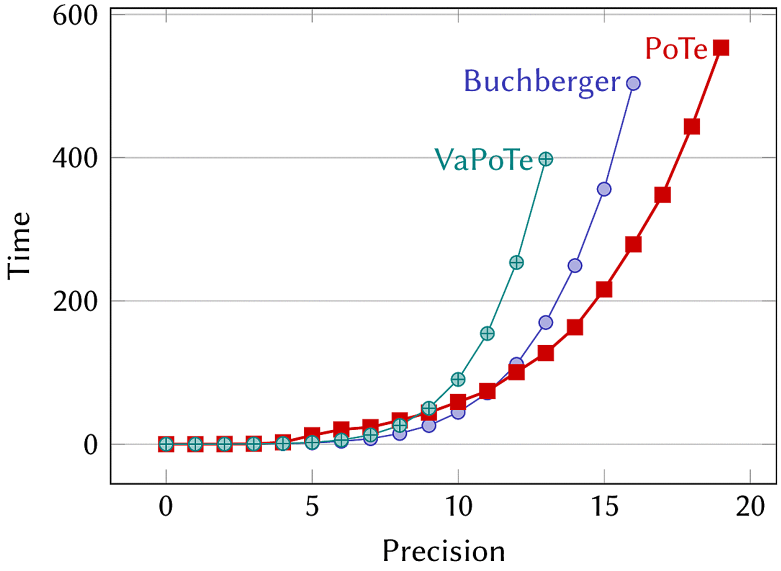






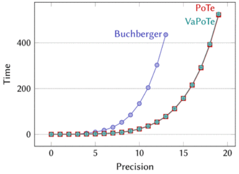
Future work

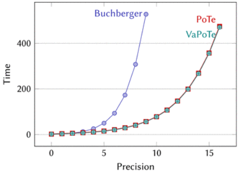
- Understand non-generic performance differences

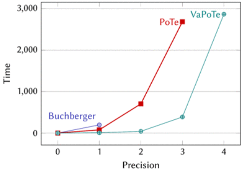


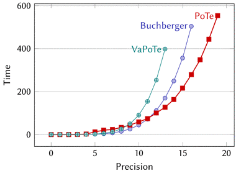
Conclusion

- Two algorithms: PoTe and VaPote
- Incremental, signature-based
- Generically equivalent
- Usually faster than Buchberger
- Distributed with SageMath 9.1



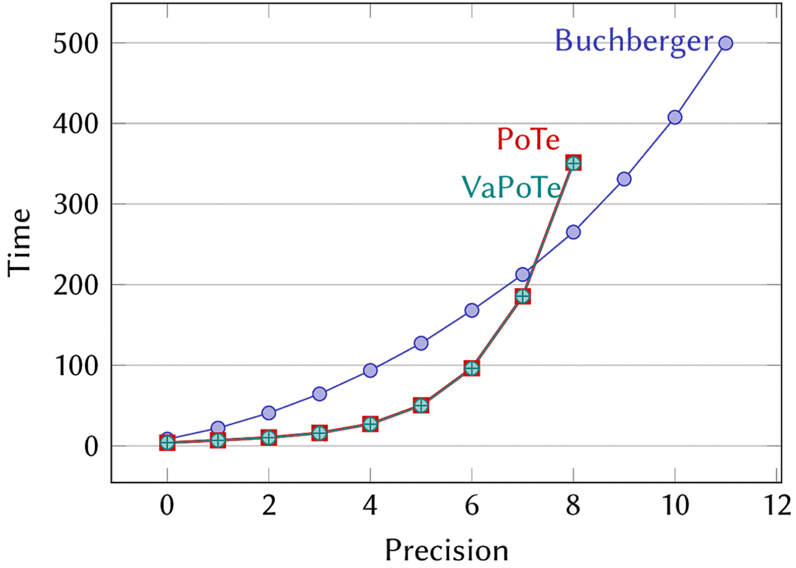






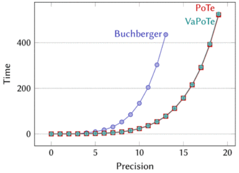
Future work

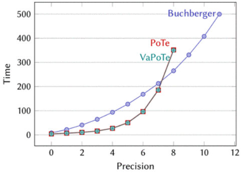
- Understand non-generic performance differences

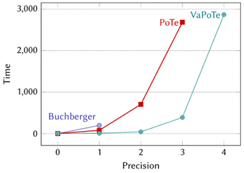


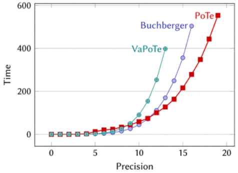
Conclusion

- Two algorithms: PoTe and VaPote
- Incremental, signature-based
- Generically equivalent
- Usually faster than Buchberger
- Distributed with SageMath 9.1









Future work

- Understand non-generic performance differences
- Examine possible optimizations between the loops
- Flatten the curve in precision