

Computing Gröbner Bases – a short overview

Christian Eder, Jean-Charles Faugère,
Fayssal Martani, John Perry and Bjarke Hammersholt Roune

Seminar of the CARAMEL Team in Nancy, France

September 11, 2014



Conventions

- ▶ $\mathcal{R} = \mathcal{K}[x_1, \dots, x_n]$, \mathcal{K} field, $<$ well-ordering on $\text{Mon}(x_1, \dots, x_n)$
- ▶ $f \in \mathcal{R}$ can be represented in a unique way by $<$.
⇒ Definitions as $\text{lc}(f)$, $\text{lm}(f)$, and $\text{lt}(f)$ make sense.
- ▶ An ideal I in \mathcal{R} is an additive subgroup of \mathcal{R} such that for $f \in I$, $g \in \mathcal{R}$ it holds that $fg \in I$.
- ▶ $G = \{g_1, \dots, g_s\} \subset \mathcal{R}$ is a Gröbner basis for $I = \langle f_1, \dots, f_m \rangle$ w.r.t. $<$
$$G \subset I \text{ and } L_<(G) = L_<(I)$$

: \iff

$$G \subset I \text{ and } L_<(G) = L_<(I)$$

Buchberger's criterion

S-polynomials

Let $f \neq 0, g \neq 0 \in \mathcal{R}$ and let $\lambda = \text{lcm}(\text{lt}(f), \text{lt}(g))$ be the least common multiple of $\text{lt}(f)$ and $\text{lt}(g)$. The **S-polynomial** between f and g is given by

$$\text{spol}(f, g) := \frac{\lambda}{\text{lt}(f)} f - \frac{\lambda}{\text{lt}(g)} g.$$

Buchberger's criterion

S-polynomials

Let $f \neq 0, g \neq 0 \in \mathcal{R}$ and let $\lambda = \text{lcm}(\text{lt}(f), \text{lt}(g))$ be the least common multiple of $\text{lt}(f)$ and $\text{lt}(g)$. The **S-polynomial** between f and g is given by

$$\text{spol}(f, g) := \frac{\lambda}{\text{lt}(f)} f - \frac{\lambda}{\text{lt}(g)} g.$$

Buchberger's criterion [5]

Let $I = \langle f_1, \dots, f_m \rangle$ be an ideal in \mathcal{R} . A finite subset $G \subset \mathcal{R}$ is a **Gröbner basis for I** if $G \subset I$ and for all $f, g \in G$: $\text{spol}(f, g) \xrightarrow{G} 0$.

Buchberger's algorithm

Input: Ideal $I = \langle f_1, \dots, f_m \rangle$

Output: Gröbner basis G for I

1. $G \leftarrow \emptyset$
2. $G \leftarrow G \cup \{f_i\}$ for all $i \in \{1, \dots, m\}$
3. Set $P \leftarrow \{\text{spol}(f_i, f_j) \mid f_i, f_j \in G, i > j\}$

Buchberger's algorithm

Input: Ideal $I = \langle f_1, \dots, f_m \rangle$

Output: Gröbner basis G for I

1. $G \leftarrow \emptyset$
2. $G \leftarrow G \cup \{f_i\}$ for all $i \in \{1, \dots, m\}$
3. Set $P \leftarrow \{\text{spol}(f_i, f_j) \mid f_i, f_j \in G, i > j\}$
4. Choose $p \in P$, $P \leftarrow P \setminus \{p\}$

Buchberger's algorithm

Input: Ideal $I = \langle f_1, \dots, f_m \rangle$

Output: Gröbner basis G for I

1. $G \leftarrow \emptyset$
2. $G \leftarrow G \cup \{f_i\}$ for all $i \in \{1, \dots, m\}$
3. Set $P \leftarrow \{\text{spol}(f_i, f_j) \mid f_i, f_j \in G, i > j\}$
4. Choose $p \in P$, $P \leftarrow P \setminus \{p\}$
 - (a) If $p \xrightarrow{G} 0 \blacktriangleright \text{no new information}$
Go on with the next element in P .
 - (b) If $p \xrightarrow{G} q \neq 0 \blacktriangleright \text{new information}$
Build new S-pair with q and add them to P .
Add q to G .
Go on with the next element in P .
5. When $P = \emptyset$ we are done and G is a Gröbner basis for I .

How to improve computations?

- ▶ Modular computations (modStd et al.)
- ▶ Predict zero reductions (Buchberger, Gebauer-Möller, Möller-Mora-Traverso, Faugère.)
- ▶ Sort pair set (Buchberger, Giovini et al., Möller et al.)
- ▶ Homogenize: d -Gröbner bases
- ▶ Change of ordering (FGLM, Gröbner Walk)
- ▶ Linear Algebra: Gauss Elimination (Lazard, Faugère)
- ▶ Sparse Gröbner Bases: Use sparsity and exploit Newton polygons (Faugère, Spaenlehauer, Svartz)
- ▶ ...

How to improve computations?

- ▶ Predict zero reductions (Buchberger, Gebauer-Möller, Möller-Mora-Traverso, Faugère.)
- ▶ Linear Algebra: Gauss Elimination (Lazard, Faugère)

- Predicting zero reductions
- Fast linear algebra for computing Gröbner bases

How to detect zero reductions in advance?

Let $I = \langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z]$ and let $<$ denote the reverse lexicographical ordering. Let

$$g_1 = xy - z^2, \quad g_2 = y^2 - z^2$$

How to detect zero reductions in advance?

Let $I = \langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z]$ and let $<$ denote the reverse lexicographical ordering. Let

$$g_1 = xy - z^2, \quad g_2 = y^2 - z^2$$

$$\begin{aligned} \text{spol}(g_2, g_1) &= xg_2 - yg_1 = \mathbf{xy^2} - xz^2 - \mathbf{xy^2} + yz^2 \\ &= -xz^2 + yz^2. \end{aligned}$$

$$\implies g_3 = \mathbf{xz^2} - \mathbf{yz^2}.$$

How to detect zero reductions in advance?

Let $I = \langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z]$ and let $<$ denote the reverse lexicographical ordering. Let

$$g_1 = xy - z^2, \quad g_2 = y^2 - z^2$$

$$\begin{aligned} \text{spol}(g_2, g_1) &= xg_2 - yg_1 = \mathbf{xy^2} - xz^2 - \mathbf{xy^2} + yz^2 \\ &= -xz^2 + yz^2. \end{aligned}$$

$$\implies g_3 = \mathbf{xz^2} - \mathbf{yz^2}.$$

$$\text{spol}(g_3, g_1) = \mathbf{xyz^2} - y^2z^2 - \mathbf{xyz^2} + z^4 = -y^2z^2 + z^4.$$

How to detect zero reductions in advance?

Let $I = \langle g_1, g_2 \rangle \in \mathbb{Q}[x, y, z]$ and let $<$ denote the reverse lexicographical ordering. Let

$$g_1 = xy - z^2, \quad g_2 = y^2 - z^2$$

$$\begin{aligned} \text{spol}(g_2, g_1) &= xg_2 - yg_1 = \mathbf{xy^2} - xz^2 - \mathbf{xy^2} + yz^2 \\ &= -xz^2 + yz^2. \end{aligned}$$

$$\implies g_3 = \mathbf{xz^2} - \mathbf{yz^2}.$$

$$\text{spol}(g_3, g_1) = \mathbf{xyz^2} - y^2z^2 - \mathbf{xyz^2} + z^4 = -y^2z^2 + z^4.$$

We can reduce further using $z^2 g_2$:

$$-y^2z^2 + z^4 + y^2z^2 - z^4 = 0.$$

How to detect zero reductions in advance?

Can we see something? How are the generators of the S-polynomials related to each other?

How to detect zero reductions in advance?

Can we see something? How are the generators of the S-polynomials related to each other?

$$\begin{aligned}\text{spol}(g_3, g_2) &= \mathbf{y^2} (xz^2 - yz^2) - \mathbf{xz^2} (y^2 - z^2) \\ &= \text{lt}(\mathbf{g_2})g_3 - \text{lt}(\mathbf{g_3})g_2 \\ &= \text{lt}(\mathbf{g_2})\text{lot}(g_3) - \text{lt}(\mathbf{g_3})\text{lot}(g_2)\end{aligned}$$

How to detect zero reductions in advance?

Can we see something? How are the generators of the S-polynomials related to each other?

$$\begin{aligned}\text{spol}(g_3, g_2) &= \mathbf{y^2} (xz^2 - yz^2) - \mathbf{xz^2} (y^2 - z^2) \\ &= \text{lt}(\mathbf{g}_2)g_3 - \text{lt}(\mathbf{g}_3)g_2 \\ &= \text{lt}(\mathbf{g}_2)\text{lot}(g_3) - \text{lt}(\mathbf{g}_3)\text{lot}(g_2)\end{aligned}$$

For all $u \in \text{support}(\text{lot}(g_3))$ we can reduce with ug_2 :

$$\begin{aligned}\implies &\text{lt}(g_2)\text{lot}(g_3) - \mathbf{g}_2\text{lot}(\mathbf{g}_3) - \text{lt}(g_3)\text{lot}(g_2) \\ &= -\text{lot}(g_2)\text{lot}(g_3) - \text{lt}(g_3)\text{lot}(g_2) \\ &= -g_3\text{lot}(g_2).\end{aligned}$$

How to detect zero reductions in advance?

Can we see something? How are the generators of the S-polynomials related to each other?

$$\begin{aligned}\text{spol}(g_3, g_2) &= \mathbf{y^2} (xz^2 - yz^2) - \mathbf{xz^2} (y^2 - z^2) \\ &= \text{lt}(\mathbf{g}_2)g_3 - \text{lt}(\mathbf{g}_3)g_2 \\ &= \text{lt}(\mathbf{g}_2)\text{lot}(g_3) - \text{lt}(\mathbf{g}_3)\text{lot}(g_2)\end{aligned}$$

For all $u \in \text{support}(\text{lot}(g_3))$ we can reduce with ug_2 :

$$\begin{aligned}&\implies \text{lt}(g_2)\text{lot}(g_3) - \mathbf{g}_2\text{lot}(\mathbf{g}_3) - \text{lt}(g_3)\text{lot}(g_2) \\ &= -\text{lot}(g_2)\text{lot}(g_3) - \text{lt}(g_3)\text{lot}(g_2) \\ &= -g_3\text{lot}(g_2).\end{aligned}$$

So we can reduce this to zero by vg_3 for all $v \in \text{support}(\text{lot}(g_2))$.

Buchberger's criteria

Product criterion [6, 7]

If $\text{lcm}(\text{lt}(f), \text{lt}(g)) = \text{lt}(f)\text{lt}(g)$ then $\text{spol}(f, g) \xrightarrow{\{f, g\}} 0$.

Buchberger's criteria

Product criterion [6, 7]

If $\text{lcm}(\text{lt}(f), \text{lt}(g)) = \text{lt}(f)\text{lt}(g)$ then $\text{spol}(f, g) \xrightarrow{\{f, g\}} 0$.

Couldn't we remove $\text{spol}(g_3, g_2)$ in a different way?

Buchberger's criteria

Product criterion [6, 7]

If $\text{lcm}(\text{lt}(f), \text{lt}(g)) = \text{lt}(f)\text{lt}(g)$ then $\text{spol}(f, g) \xrightarrow{\{f, g\}} 0$.

Couldn't we remove $\text{spol}(g_3, g_2)$ in a different way?

$$\text{lt}(g_1) = xy \mid xy^2z^2 = \text{lcm}(\text{lt}(g_3), \text{lt}(g_2))$$

Buchberger's criteria

Product criterion [6, 7]

If $\text{lcm}(\text{lt}(f), \text{lt}(g)) = \text{lt}(f)\text{lt}(g)$ then $\text{spol}(f, g) \xrightarrow{\{f, g\}} 0$.

Couldn't we remove $\text{spol}(g_3, g_2)$ in a different way?

$$\text{lt}(g_1) = xy \mid xy^2z^2 = \text{lcm}(\text{lt}(g_3), \text{lt}(g_2))$$

\implies We can rewrite $\text{spol}(g_3, g_2)$:

$$\begin{aligned} \text{spol}(g_3, g_2) &= y\underbrace{\text{spol}(g_3, g_1)}_{\xrightarrow{G} 0} - z^2 \underbrace{\text{spol}(g_2, g_1)}_{\xrightarrow{G} -g_3} = y(yg_3 - z^2 g_1) - z^2(xg_2 - yg_1) \end{aligned}$$

Buchberger's criteria

Product criterion [6, 7]

If $\text{lcm}(\text{lt}(f), \text{lt}(g)) = \text{lt}(f)\text{lt}(g)$ then $\text{spol}(f, g) \xrightarrow{\{f, g\}} 0$.

Couldn't we remove $\text{spol}(g_3, g_2)$ in a different way?

$$\text{lt}(g_1) = xy \mid xy^2z^2 = \text{lcm}(\text{lt}(g_3), \text{lt}(g_2))$$

\implies We can rewrite $\text{spol}(g_3, g_2)$:

$$\begin{aligned} \text{spol}(g_3, g_2) &= y \underbrace{\text{spol}(g_3, g_1)}_{\xrightarrow{G} 0} - z^2 \underbrace{\text{spol}(g_2, g_1)}_{\xrightarrow{G} -g_3} = y(yg_3 - z^2g_1) - z^2(xg_2 - yg_1) \end{aligned}$$

Standard representations of $\text{spol}(g_2, g_1)$ and $\text{spol}(g_3, g_1)$

\implies Standard representation of $\text{spol}(g_3, g_2)$.

Buchberger's criteria

Chain criterion [8]

Let $f, g, h \in \mathcal{R}$, $G \subset \mathcal{R}$ finite. If

1. $\text{lt}(h) \mid \text{lcm}(\text{lt}(f), \text{lt}(g))$, and
2. $\text{spol}(f, h)$ and $\text{spol}(h, g)$ have a standard representation w.r.t. G respectively,

then $\text{spol}(f, g)$ has a standard representation w.r.t. G .

Buchberger's criteria

Chain criterion [8]

Let $f, g, h \in \mathcal{R}$, $G \subset \mathcal{R}$ finite. If

1. $\text{lt}(h) \mid \text{lcm}(\text{lt}(f), \text{lt}(g))$, and
2. $\text{spol}(f, h)$ and $\text{spol}(h, g)$ have a standard representation w.r.t. G respectively,

then $\text{spol}(f, g)$ has a standard representation w.r.t. G .

Note

Do not remove too much information! If $\lambda = 1$ and

$$\text{spol}(f, g) = \lambda \text{spol}(f, h) + \sigma \text{spol}(h, g),$$

then we can remove $\text{spol}(f, g)$ or $\text{spol}(f, h)$ but not both!

Buchberger's criteria

Chain criterion [8]

Let $f, g, h \in \mathcal{R}$, $G \subset \mathcal{R}$ finite. If

1. $\text{lt}(h) \mid \text{lcm}(\text{lt}(f), \text{lt}(g))$, and
2. $\text{spol}(f, h)$ and $\text{spol}(h, g)$ have a standard representation w.r.t. G respectively,

then $\text{spol}(f, g)$ has a standard representation w.r.t. G .

Note

Do not remove too much information! If $\lambda = 1$ and

$$\text{spol}(f, g) = \lambda \text{spol}(f, h) + \sigma \text{spol}(h, g),$$

then we can remove $\text{spol}(f, g)$ or $\text{spol}(f, h)$ but not both!

How to combine Product and Chain criterion?

Gebauer-Möller installation [32]

We add a new element h to G and generate new pairs $P' := \{(f, h) \mid f \in G\}$.

Gebauer-Möller installation [32]

We add a new element h to G and generate new pairs $P' := \{(f, h) \mid f \in G\}$.

We update the pairs in 4 steps:

Gebauer-Möller installation [32]

We add a new element h to G and generate new pairs $P' := \{(f, h) \mid f \in G\}$.

We update the pairs in 4 steps:

1. If $(f, g) \in P$ s.t.

- ▷ $\text{lt}(h) \mid \text{lcm}(\text{lt}(f), \text{lt}(g))$,
- ▷ $\text{lcm}(\text{lt}(f), \text{lt}(h)) \neq \text{lcm}(\text{lt}(f), \text{lt}(g))$,
- ▷ $\text{lcm}(\text{lt}(g), \text{lt}(h)) \neq \text{lcm}(\text{lt}(f), \text{lt}(g))$

⇒ Remove (f, g) from P . [**P done**]

Gebauer-Möller installation [32]

We add a new element h to G and generate new pairs $P' := \{(f, h) \mid f \in G\}$.

We update the pairs in 4 steps:

1. If $(f, g) \in P$ s.t.

- ▷ $\text{lt}(h) \mid \text{lcm}(\text{lt}(f), \text{lt}(g))$,
- ▷ $\text{lcm}(\text{lt}(f), \text{lt}(h)) \neq \text{lcm}(\text{lt}(f), \text{lt}(g))$,
- ▷ $\text{lcm}(\text{lt}(g), \text{lt}(h)) \neq \text{lcm}(\text{lt}(f), \text{lt}(g))$

⇒ Remove (f, g) from P . [**P done**]

2. Fix $(f, h) \in P'$. If $(g, h) \in P' \setminus \{(f, h)\}$ s.t.

- ▷ $\exists \lambda > 1$ and $\text{lcm}(\text{lt}(f), \text{lt}(h)) = \lambda \text{lcm}(\text{lt}(g), \text{lt}(h))$

⇒ Remove (g, h) from P' .

Gebauer-Möller installation [32]

We add a new element h to G and generate new pairs $P' := \{(f, h) \mid f \in G\}$.

We update the pairs in 4 steps:

1. If $(f, g) \in P$ s.t.

- ▷ $\text{lt}(h) \mid \text{lcm}(\text{lt}(f), \text{lt}(g))$,
- ▷ $\text{lcm}(\text{lt}(f), \text{lt}(h)) \neq \text{lcm}(\text{lt}(f), \text{lt}(g))$,
- ▷ $\text{lcm}(\text{lt}(g), \text{lt}(h)) \neq \text{lcm}(\text{lt}(f), \text{lt}(g))$

⇒ Remove (f, g) from P . [**P done**]

2. Fix $(f, h) \in P'$. If $(g, h) \in P' \setminus \{(f, h)\}$ s.t.

- ▷ $\exists \lambda > 1$ and $\text{lcm}(\text{lt}(f), \text{lt}(h)) = \lambda \text{lcm}(\text{lt}(g), \text{lt}(h))$

⇒ Remove (g, h) from P' .

3. Fix $(f, h) \in P'$. If $(g, h) \in P' \setminus \{(f, h)\}$ s.t.

- ▷ $\text{lcm}(\text{lt}(f), \text{lt}(h)) = \text{lcm}(\text{lt}(g), \text{lt}(h))$

⇒ Remove (g, h) from P' . [**Chain criterion done**]

Gebauer-Möller installation [32]

We add a new element h to G and generate new pairs $P' := \{(f, h) \mid f \in G\}$.

We update the pairs in 4 steps:

1. If $(f, g) \in P$ s.t.

- ▷ $\text{lt}(h) \mid \text{lcm}(\text{lt}(f), \text{lt}(g))$,
- ▷ $\text{lcm}(\text{lt}(f), \text{lt}(h)) \neq \text{lcm}(\text{lt}(f), \text{lt}(g))$,
- ▷ $\text{lcm}(\text{lt}(g), \text{lt}(h)) \neq \text{lcm}(\text{lt}(f), \text{lt}(g))$

⇒ Remove (f, g) from P . [**P done**]

2. Fix $(f, h) \in P'$. If $(g, h) \in P' \setminus \{(f, h)\}$ s.t.

- ▷ $\exists \lambda > 1$ and $\text{lcm}(\text{lt}(f), \text{lt}(h)) = \lambda \text{lcm}(\text{lt}(g), \text{lt}(h))$

⇒ Remove (g, h) from P' .

3. Fix $(f, h) \in P'$. If $(g, h) \in P' \setminus \{(f, h)\}$ s.t.

- ▷ $\text{lcm}(\text{lt}(f), \text{lt}(h)) = \text{lcm}(\text{lt}(g), \text{lt}(h))$

⇒ Remove (g, h) from P' . [**Chain criterion done**]

4. If $(f, h) \in P'$ s.t. $\text{lcm}(\text{lt}(f), \text{lt}(h)) = \text{lt}(f)\text{lt}(h)$

⇒ Remove (f, h) from P' . [**Product criterion done**]

Can we do even better?

In our example we still need to consider

$$\text{spol}(g_3, g_1) \xrightarrow{G} 0.$$

Can we do even better?

In our example we still need to consider

$$\text{spol}(g_3, g_1) \xrightarrow{G} 0.$$

How to get rid of this useless computation?

Can we do even better?

In our example we still need to consider

$$\text{spol}(g_3, g_1) \xrightarrow{G} 0.$$

How to get rid of this useless computation?

Use more structure of $I \implies \text{Signatures}$

Signatures

Let $I = \langle f_1, \dots, f_m \rangle \subset \mathcal{R}$.

Idea: Give each $f \in I$ a bit more structure:

Signatures

Let $I = \langle f_1, \dots, f_m \rangle \subset \mathcal{R}$.

Idea: Give each $f \in I$ a bit more structure:

1. Let \mathcal{R}^m be generated by e_1, \dots, e_m and let \prec be a compatible monomial order on the monomials of \mathcal{R}^m .

Signatures

Let $I = \langle f_1, \dots, f_m \rangle \subset \mathcal{R}$.

Idea: Give each $f \in I$ a bit more structure:

1. Let \mathcal{R}^m be generated by e_1, \dots, e_m and let \prec be a compatible monomial order on the monomials of \mathcal{R}^m .
2. Let $\alpha \mapsto \bar{\alpha} : \mathcal{R}^m \rightarrow \mathcal{R}$ such that $\bar{e}_i = f_i$ for all i .

Signatures

Let $I = \langle f_1, \dots, f_m \rangle \subset \mathcal{R}$.

Idea: Give each $f \in I$ a bit more structure:

1. Let \mathcal{R}^m be generated by e_1, \dots, e_m and let \prec be a compatible monomial order on the monomials of \mathcal{R}^m .
2. Let $\alpha \mapsto \bar{\alpha} : \mathcal{R}^m \rightarrow \mathcal{R}$ such that $\bar{e}_i = f_i$ for all i .
3. Each $f \in I$ can be represented via some $\alpha \in \mathcal{R}^m$: $f = \bar{\alpha}$

Signatures

Let $I = \langle f_1, \dots, f_m \rangle \subset \mathcal{R}$.

Idea: Give each $f \in I$ a bit more structure:

1. Let \mathcal{R}^m be generated by e_1, \dots, e_m and let \prec be a compatible monomial order on the monomials of \mathcal{R}^m .
2. Let $\alpha \mapsto \bar{\alpha} : \mathcal{R}^m \rightarrow \mathcal{R}$ such that $\bar{e}_i = f_i$ for all i .
3. Each $f \in I$ can be represented via some $\alpha \in \mathcal{R}^m$: $f = \bar{\alpha}$
4. A **signature** of f is given by $s(f) = \text{lt}_{\prec}(\alpha)$ where $f = \bar{\alpha}$.

Signatures

Let $I = \langle f_1, \dots, f_m \rangle \subset \mathcal{R}$.

Idea: Give each $f \in I$ a bit more structure:

1. Let \mathcal{R}^m be generated by e_1, \dots, e_m and let \prec be a compatible monomial order on the monomials of \mathcal{R}^m .
2. Let $\alpha \mapsto \bar{\alpha} : \mathcal{R}^m \rightarrow \mathcal{R}$ such that $\bar{e}_i = f_i$ for all i .
3. Each $f \in I$ can be represented via some $\alpha \in \mathcal{R}^m$: $f = \bar{\alpha}$
4. A **signature** of f is given by $\text{s}(f) = \text{lt}_{\prec}(\alpha)$ where $f = \bar{\alpha}$.
5. An element $\alpha \in \mathcal{R}^m$ such that $\bar{\alpha} = 0$ is called a **syzygy**.

Our example again – with signatures and \prec_{pot}

$$g_1 = xy - z^2, \mathfrak{s}(g_1) = e_1,$$

$$g_2 = y^2 - z^2, \mathfrak{s}(g_2) = e_2.$$

Our example again – with signatures and \prec_{pot}

$$g_1 = xy - z^2, \mathfrak{s}(g_1) = e_1,$$

$$g_2 = y^2 - z^2, \mathfrak{s}(g_2) = e_2.$$

$$g_3 = \text{spol}(g_2, g_1) = xg_2 - yg_1$$

$$\Rightarrow \mathfrak{s}(g_3) = x \mathfrak{s}(g_2) = xe_2.$$

Our example again – with signatures and \prec_{pot}

$$g_1 = xy - z^2, \mathfrak{s}(g_1) = e_1,$$

$$g_2 = y^2 - z^2, \mathfrak{s}(g_2) = e_2.$$

$$g_3 = \text{spol}(g_2, g_1) = xg_2 - yg_1$$

$$\Rightarrow \mathfrak{s}(g_3) = x \mathfrak{s}(g_2) = xe_2.$$

$$\text{spol}(g_3, g_1) = yg_3 - z^2 g_1$$

$$\Rightarrow \mathfrak{s}(\text{spol}(g_3, g_1)) = y \mathfrak{s}(g_3) = xye_2.$$

Our example again – with signatures and \prec_{pot}

$$g_1 = xy - z^2, \mathfrak{s}(g_1) = e_1,$$

$$g_2 = y^2 - z^2, \mathfrak{s}(g_2) = e_2.$$

$$g_3 = \text{spol}(g_2, g_1) = xg_2 - yg_1$$

$$\Rightarrow \mathfrak{s}(g_3) = x \mathfrak{s}(g_2) = xe_2.$$

$$\text{spol}(g_3, g_1) = yg_3 - z^2 g_1$$

$$\Rightarrow \mathfrak{s}(\text{spol}(g_3, g_1)) = y \mathfrak{s}(g_3) = xy e_2.$$

Note that $\mathfrak{s}(\text{spol}(g_3, g_1)) = xy e_2$ and $\text{Im}(g_1) = xy$.

Think in the module

$\alpha \in \mathcal{R}^m \implies$ polynomial $\overline{\alpha}$ with $\text{lt}(\overline{\alpha})$, signature $\mathfrak{s}(\alpha) = \text{lt}(\alpha)$

Think in the module

$\alpha \in \mathcal{R}^m \implies$ polynomial $\overline{\alpha}$ with $\text{lt}(\overline{\alpha})$, signature $\mathfrak{s}(\alpha) = \text{lt}(\alpha)$

S-pairs/S-polynomials:

$$\text{spol}(\overline{\alpha}, \overline{\beta}) = a\overline{\alpha} - b\overline{\beta} \implies \text{spair}(\alpha, \beta) = a\alpha - b\beta$$

Think in the module

$\alpha \in \mathcal{R}^m \implies$ polynomial $\overline{\alpha}$ with $\text{lt}(\overline{\alpha})$, signature $\mathfrak{s}(\alpha) = \text{lt}(\alpha)$

S-pairs/S-polynomials:

$$\text{spol}\left(\overline{\alpha}, \overline{\beta}\right) = a\overline{\alpha} - b\overline{\beta} \implies \text{spair}(\alpha, \beta) = a\alpha - b\beta$$

\mathfrak{s} -reductions:

$$\overline{\gamma} - d\overline{\delta} \implies \gamma - d\delta$$

Think in the module

$\alpha \in \mathcal{R}^m \implies$ polynomial $\overline{\alpha}$ with $\text{lt}(\overline{\alpha})$, signature $\mathfrak{s}(\alpha) = \text{lt}(\alpha)$

S-pairs/S-polynomials:

$$\text{spol}(\overline{\alpha}, \overline{\beta}) = a\overline{\alpha} - b\overline{\beta} \implies \text{spair}(\alpha, \beta) = a\alpha - b\beta$$

\mathfrak{s} -reductions:

$$\overline{\gamma} - d\overline{\delta} \implies \gamma - d\delta$$

Remark

In the following we need one detail from signature-based Gröbner Basis computations:

We pick from P by increasing signature.

Signature-based criteria

$\mathfrak{s}(\alpha) = \mathfrak{s}(\beta) \implies \text{Compute 1, remove 1.}$

Signature-based criteria

$$\mathfrak{s}(\alpha) = \mathfrak{s}(\beta) \implies \text{Compute 1, remove 1.}$$

Sketch of proof

1. $\mathfrak{s}(\alpha - \beta) \prec \mathfrak{s}(\alpha), \mathfrak{s}(\beta)$.
2. All S-pairs are handled by increasing signature.
 \Rightarrow All relations $\prec \mathfrak{s}(\alpha)$ are known:

$\alpha = \beta + \text{elements of smaller signature}$

□

Signature-based criteria

S-pairs in signature T

Signature-based criteria

S-pairs in signature T

What are all possible configurations to reach signature T ?

Signature-based criteria

S-pairs in signature T

$$\mathfrak{R}_T = \left\{ a\alpha \mid \alpha \text{ handled by the algorithm and } s(a\alpha) = T \right\}$$

What are all possible configurations to reach signature T ?

Signature-based criteria

S-pairs in signature T

$$\mathfrak{R}_T = \left\{ a\alpha \mid \alpha \text{ handled by the algorithm and } s(a\alpha) = T \right\}$$

What are all possible configurations to reach signature T ?

Define an order on \mathfrak{R}_T and choose the maximal element.

Special cases

$$\mathfrak{R}_T = \left\{ a\alpha \mid \alpha \text{ handled by the algorithm and } s(a\alpha) = T \right\}$$

Special cases

$$\mathfrak{R}_T = \left\{ a\alpha \mid \alpha \text{ handled by the algorithm and } s(a\alpha) = T \right\}$$

Choose $b\beta$ to be an element of \mathfrak{R}_T maximal w.r.t. an order \trianglelefteq .

Special cases

$$\mathfrak{R}_T = \left\{ a\alpha \mid \alpha \text{ handled by the algorithm and } s(a\alpha) = T \right\}$$

Choose $b\beta$ to be an element of \mathfrak{R}_T maximal w.r.t. an order \trianglelefteq .

1. If $b\beta$ is a syzygy \implies Go on to next signature.

Special cases

$$\mathfrak{R}_T = \left\{ a\alpha \mid \alpha \text{ handled by the algorithm and } s(a\alpha) = T \right\}$$

Choose $b\beta$ to be an element of \mathfrak{R}_T maximal w.r.t. an order \trianglelefteq .

1. If $b\beta$ is a syzygy \implies Go on to next signature.
2. If $b\beta$ is not part of an S-pair \implies Go on to next signature.

Special cases

$$\mathfrak{R}_T = \left\{ a\alpha \mid \alpha \text{ handled by the algorithm and } \mathfrak{s}(a\alpha) = T \right\}$$

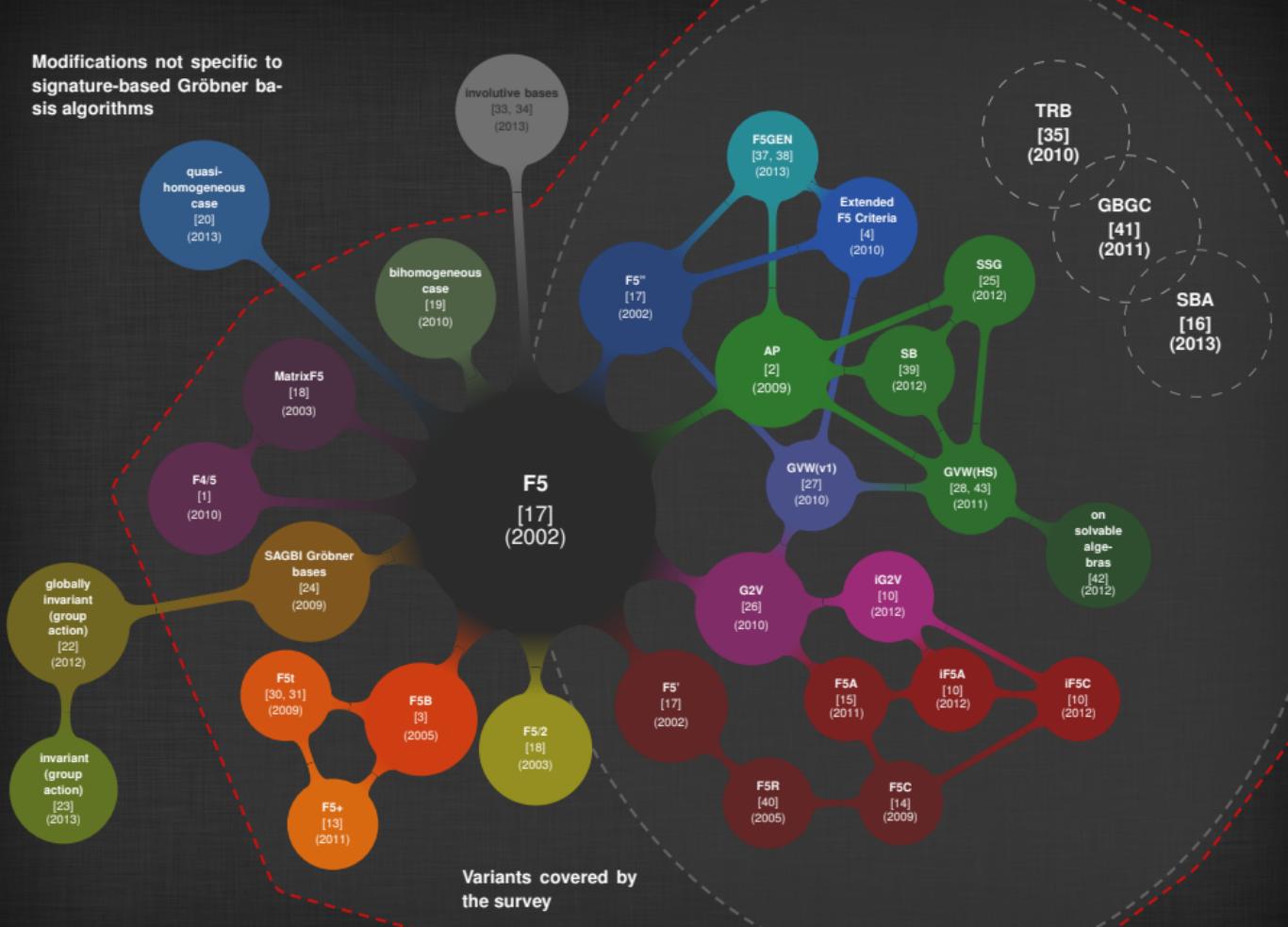
Choose $b\beta$ to be an element of \mathfrak{R}_T maximal w.r.t. an order \trianglelefteq .

1. If $b\beta$ is a syzygy \implies Go on to next signature.
2. If $b\beta$ is not part of an S-pair \implies Go on to next signature.

Revisiting our example with \prec_{pot}

$$\begin{aligned} \mathfrak{s}(\text{spol}(g_3, g_1)) &= xy\mathbf{e}_2 \\ \left. \begin{array}{l} g_1 = xy - z^2 \\ g_2 = y^2 - z^2 \end{array} \right\} \Rightarrow \text{psyz}(g_2, g_1) &= g_1\mathbf{e}_2 - g_2\mathbf{e}_1 = xy\mathbf{e}_2 + \dots \end{aligned}$$

Modifications not specific to signature-based Gröbner basis algorithms



Variants covered by
the survey

zero reductions (Singular-4-0-0, \mathbb{F}_{32003})

Benchmark	STD	SBA \prec_{pot}	SBA $\prec_{\text{d-pot}}$	SBA \prec_{lt}
cyclic-8	4,284	243	243	671
cyclic-8-h	5,843	243	243	671
eco-11	3,476	0	749	749
eco-11-h	5,429	502	502	749
katsura-11	3,933	0	0	348
katsura-11-h	3,933	0	0	348
noon-9	25,508	0	0	682
noon-9-h	25,508	0	0	682
Random(11,2,2)	6,292	0	0	590
HRandom(11,2,2)	6,292	0	0	590
Random(12,2,2)	13,576	0	0	1,083
HRandom(12,2,2)	13,576	0	0	1,083

Time in seconds (Singular-4-0-0, \mathbb{F}_{32003})

Benchmark	STD	SBA \prec_{pot}	SBA $\prec_{\text{d-pot}}$	SBA \prec_{lt}
cyclic-8	32.480	44.310	100.780	38.120
cyclic-8-h	38.300	35.770	98.440	32.640
eco-11	28.450	3.450	27.360	13.270
eco-11-h	20.630	11.600	14.840	7.960
katsura-11	54.780	35.720	31.010	11.790
katsura-11-h	51.260	34.080	32.590	17.230
noon-9	29.730	12.940	14.620	15.220
noon-9-h	34.410	17.850	20.090	20.510
Random(11,2,2)	267.810	77.430	130.400	28.640
HRandom(11,2,2)	22.970	14.060	39.320	3.540
Random(12,2,2)	2,069.890	537.340	1,062.390	176.920
HRandom(12,2,2)	172.910	112.420	331.680	22.060

Can we combine both attempts?

Buchberger's Product and Chain criterion can be combined with the Rewrite criterion [29, 33, 11]:

Can we combine both attempts?

Buchberger's Product and Chain criterion can be combined with the Rewrite criterion [29, 33, 11]:

Chain criterion is a special case of the Rewrite criterion
⇒ already included.

Can we combine both attempts?

Buchberger's Product and Chain criterion can be combined with the Rewrite criterion [29, 33, 11]:

Chain criterion is a special case of the Rewrite criterion
⇒ already included.

Product criterion is not always (but mostly) included.

Can we combine both attempts?

Buchberger's Product and Chain criterion can be combined with the Rewrite criterion [29, 33, 11]:

Chain criterion is a special case of the Rewrite criterion
⇒ already included.

Product criterion is not always (but mostly) included.

α added to \mathcal{G}



Generate **all** possible
principal syzygies with α .
(e.g. **GVW**)

Can we combine both attempts?

Buchberger's Product and Chain criterion can be combined with the Rewrite criterion [29, 33, 11]:

Chain criterion is a special case of the Rewrite criterion
⇒ already included.

Product criterion is not always (but mostly) included.

α added to \mathcal{G}
▼
Generate **all** possible
principal syzygies with α .
(e.g. **GVW**)

S-pair fulfilling Product criterion
not detected by Rewrite criterion
▼
Add **one** corresponding syzygy.
(e.g. **SBA** in **Singular**)

Experimental results

Implementation done in **Singular** [9]

Benchmark	STD	SBA ↲ _{pot}	SBA ↲ _{lt}	
	ZR		ZR	ZR / PC
cyclic-8	4284	243	671	671 / 0
cyclic-8-h	5843	243	671	671 / 0
eco-11	3476	0	749	749 / 0
eco-11-h	5429	502	749	718 / 0
katsura-11	3933	0	348	304 / 0
katsura-11-h	3933	0	348	304 / 0
noon-9	25508	0	682	646 / 0
noon-9-h	25508	0	682	646 / 0
binomial-6-2	21	6	15	8 / 7
binomial-6-3	20	13	15	9 / 6
binomial-7-3	27	24	21	21 / 0
binomial-7-4	41	16	19	16 / 3
binomial-8-3	53	23	27	27 / 0
binomial-8-4	40	31	26	26 / 0

And what's about SBA using \prec_{pot} ?

Conjecture [11]

Every S-polynomial fulfilling the Product criterion is also detected by the Rewrite criterion in **SBA** using \prec_{pot} .

And what's about SBA using \prec_{pot} ?

Conjecture [11]

Every S-polynomial fulfilling the Product criterion is also detected by the Rewrite criterion in **SBA** using \prec_{pot} .

- ▶ We checked several million examples, all fulfilling the conjecture.
- ▶ Until now we cannot prove this.

And what's about SBA using \prec_{pot} ?

Conjecture [11]

Every S-polynomial fulfilling the Product criterion is also detected by the Rewrite criterion in **SBA** using \prec_{pot} .

- ▶ We checked several million examples, all fulfilling the conjecture.
- ▶ Until now we cannot prove this.

Ongoing work:

1. Describe in detail the connection between our conjecture and Moreno-Socías conjecture [36].
2. Try to exploit even more algebraic structures for predicting zero reductions.

- Predicting zero reductions
- Fast linear algebra for computing Gröbner bases

Buchberger's algorithm - revisited

Input: Ideal $I = \langle f_1, \dots, f_m \rangle$

Output: Gröbner basis G for I

1. $G \leftarrow \emptyset$
2. $G \leftarrow G \cup \{f_i\}$ for all $i \in \{1, \dots, m\}$
3. Set $P \leftarrow \{\text{spol}(f_i, f_j) \mid f_i, f_j \in G, i > j\}$
4. Choose $p \in P$, $P \leftarrow P \setminus \{p\}$
 - (a) If $p \xrightarrow{G} 0 \blacktriangleright \text{no new information}$
Go on with the next element in P .
 - (b) If $p \xrightarrow{G} q \neq 0 \blacktriangleright \text{new information}$
Build new S-pair with q and add them to P .
Add q to G .
Go on with the next element in P .
5. When $P = \emptyset$ we are done and G is a Gröbner basis for I .

Faugère's F4 algorithm

Input: Ideal $I = \langle f_1, \dots, f_m \rangle$

Output: Gröbner basis G for I

1. $G \leftarrow \emptyset$
2. $G \leftarrow G \cup \{f_i\}$ for all $i \in \{1, \dots, m\}$
3. Set $P \leftarrow \{(af, bg) \mid f, g \in G\}$
4. $d \leftarrow 0$
5. while $P \neq \emptyset$:

Faugère's F4 algorithm

Input: Ideal $I = \langle f_1, \dots, f_m \rangle$

Output: Gröbner basis G for I

1. $G \leftarrow \emptyset$
2. $G \leftarrow G \cup \{f_i\}$ for all $i \in \{1, \dots, m\}$
3. Set $P \leftarrow \{(af, bg) \mid f, g \in G\}$
4. $d \leftarrow 0$
5. while $P \neq \emptyset$:
 - (a) $d \leftarrow d + 1$
 - (b) $P_d \leftarrow \text{Select}(P)$, $P \leftarrow P \setminus P_d$
 - (c) $L_d \leftarrow \{af, bg \mid (af, bg) \in P_d\}$
 - (d) $L_d \leftarrow \text{Symbolic Preprocessing}(L_d, G)$
 - (e) $F_d \leftarrow \text{Reduction}(L_d, G)$
 - (f) for $h \in F_d$:
 - If $\text{lt}(h) \notin L(G)$ (all other h are “useless”):
 - ▷ $P \leftarrow P \cup \{\text{new pairs with } h\}$
 - ▷ $G \leftarrow G \cup \{h\}$
6. Return G

Differences to Buchberger

1. Select a subset P_d of P , not only one element.
2. Do a **symbolic preprocessing**:
Search and store reducers, but do not reduce.
3. Do a **full reduction of P_d** at once:
Reduce a subset of \mathcal{R} by a subset of \mathcal{R}

Differences to Buchberger

1. Select a subset P_d of P , not only one element.
2. Do a **symbolic preprocessing**:
Search and store reducers, but do not reduce.
3. Do a **full reduction of P_d** at once:
Reduce a subset of \mathcal{R} by a subset of \mathcal{R}

If **Select**(P) selects only 1 pair F4 is just Buchberger's algorithm.
Usually one chooses the normal selection strategy,
i.e. all pairs of lowest degree.

Symbolic preprocessing

Input: L, G finite subsets of \mathcal{R}

Output: a finite subset of \mathcal{R}

1. $F \leftarrow L$
2. $D \leftarrow L(F)$ (S-pairs already reduce lead terms)
3. while $T(F) \neq D$:
 - (a) Choose $m \in T(F) \setminus D$, $D \leftarrow D \cup \{m\}$.
 - (b) If $m \in L(G) \Rightarrow \exists g \in G$ and $\lambda \in \mathcal{R}$ such that $\lambda \text{ lt}(g) = m$
 ▷ $F \leftarrow F \cup \{\lambda g\}$
4. Return F

Symbolic preprocessing

Input: L, G finite subsets of \mathcal{R}

Output: a finite subset of \mathcal{R}

1. $F \leftarrow L$
2. $D \leftarrow L(F)$ (S-pairs already reduce lead terms)
3. while $T(F) \neq D$:
 - (a) Choose $m \in T(F) \setminus D$, $D \leftarrow D \cup \{m\}$.
 - (b) If $m \in L(G) \Rightarrow \exists g \in G$ and $\lambda \in \mathcal{R}$ such that $\lambda \text{ lt}(g) = m$
 ▷ $F \leftarrow F \cup \{\lambda g\}$
4. Return F

We optimize this soon!

Reduction

Input: L, G finite subsets of \mathcal{R}

Output: a finite subset of \mathcal{R}

1. $M \leftarrow$ Macaulay matrix of L
2. $M \leftarrow$ Gaussian Elimination of M (Linear algebra)
3. $F \leftarrow$ polynomials from rows of M
4. Return F

Reduction

Input: L, G finite subsets of \mathcal{R}

Output: a finite subset of \mathcal{R}

1. $M \leftarrow$ Macaulay matrix of L
2. $M \leftarrow$ Gaussian Elimination of M (Linear algebra)
3. $F \leftarrow$ polynomials from rows of M
4. Return F

Macaulay matrix

columns $\hat{=}$ monomials (sorted by monomial order $<$)
rows $\hat{=}$ coeffs of polynomials in L

Example: Cyclic-4

$\mathcal{R} = \mathbb{Q}[a, b, c, d]$, $<$ denotes DRL and we use the normal selection strategy for **Select**(P). $I = \langle f_1, \dots, f_4 \rangle$, where

$$f_1 = abcd - 1,$$

$$f_2 = abc + abd + acd + bcd,$$

$$f_3 = ab + bc + ad + cd,$$

$$f_4 = a + b + c + d.$$

Example: Cyclic-4

$\mathcal{R} = \mathbb{Q}[a, b, c, d]$, $<$ denotes DRL and we use the normal selection strategy for **Select**(P). $I = \langle f_1, \dots, f_4 \rangle$, where

$$f_1 = abcd - 1,$$

$$f_2 = abc + abd + acd + bcd,$$

$$f_3 = ab + bc + ad + cd,$$

$$f_4 = a + b + c + d.$$

We start with $G = \{f_4\}$ and $P_1 = \{(f_3, bf_4)\}$, thus $L_1 = \{f_3, bf_4\}$.

Example: Cyclic-4

$\mathcal{R} = \mathbb{Q}[a, b, c, d]$, $<$ denotes DRL and we use the normal selection strategy for **Select**(P). $I = \langle f_1, \dots, f_4 \rangle$, where

$$f_1 = abcd - 1,$$

$$f_2 = abc + abd + acd + bcd,$$

$$f_3 = ab + bc + ad + cd,$$

$$f_4 = a + b + c + d.$$

We start with $G = \{f_4\}$ and $P_1 = \{(f_3, bf_4)\}$, thus $L_1 = \{f_3, bf_4\}$.
Let us do **symbolic preprocessing**:

$$\begin{aligned} T(L_1) &= \{\textcolor{blue}{ab}, b^2, bc, ad, bd, cd\} \\ L_1 &= \{f_3, bf_4\} \end{aligned}$$

Example: Cyclic-4

$\mathcal{R} = \mathbb{Q}[a, b, c, d]$, $<$ denotes DRL and we use the normal selection strategy for **Select**(P). $I = \langle f_1, \dots, f_4 \rangle$, where

$$\begin{aligned}f_1 &= abcd - 1, \\f_2 &= abc + abd + acd + bcd, \\f_3 &= ab + bc + ad + cd, \\f_4 &= a + b + c + d.\end{aligned}$$

We start with $G = \{f_4\}$ and $P_1 = \{(f_3, bf_4)\}$, thus $L_1 = \{f_3, bf_4\}$.
Let us do **symbolic preprocessing**:

$$\begin{aligned}T(L_1) &= \{ab, b^2, bc, ad, bd, cd\} \\L_1 &= \{f_3, bf_4\}\end{aligned}$$

$$b^2 \notin L(G),$$

Example: Cyclic-4

$\mathcal{R} = \mathbb{Q}[a, b, c, d]$, $<$ denotes DRL and we use the normal selection strategy for **Select**(P). $I = \langle f_1, \dots, f_4 \rangle$, where

$$\begin{aligned}f_1 &= abcd - 1, \\f_2 &= abc + abd + acd + bcd, \\f_3 &= ab + bc + ad + cd, \\f_4 &= a + b + c + d.\end{aligned}$$

We start with $G = \{f_4\}$ and $P_1 = \{(f_3, bf_4)\}$, thus $L_1 = \{f_3, bf_4\}$.
Let us do **symbolic preprocessing**:

$$\begin{aligned}T(L_1) &= \{ab, b^2, bc, ad, bd, cd\} \\L_1 &= \{f_3, bf_4\}\end{aligned}$$

$$b^2 \notin L(G), bc \notin L(G),$$

Example: Cyclic-4

$\mathcal{R} = \mathbb{Q}[a, b, c, d]$, $<$ denotes DRL and we use the normal selection strategy for **Select**(P). $I = \langle f_1, \dots, f_4 \rangle$, where

$$f_1 = abcd - 1,$$

$$f_2 = abc + abd + acd + bcd,$$

$$f_3 = ab + bc + ad + cd,$$

$$f_4 = a + b + c + d.$$

We start with $G = \{f_4\}$ and $P_1 = \{(f_3, bf_4)\}$, thus $L_1 = \{f_3, bf_4\}$.
Let us do **symbolic preprocessing**:

$$\begin{aligned} T(L_1) &= \{ab, b^2, bc, ad, bd, cd, d^2\} \\ L_1 &= \{f_3, bf_4, df_4\} \end{aligned}$$

$b^2 \notin L(G)$, $bc \notin L(G)$, $d \text{lt}(f_4) = ad$,

Example: Cyclic-4

$\mathcal{R} = \mathbb{Q}[a, b, c, d]$, $<$ denotes DRL and we use the normal selection strategy for **Select**(P). $I = \langle f_1, \dots, f_4 \rangle$, where

$$f_1 = abcd - 1,$$

$$f_2 = abc + abd + acd + bcd,$$

$$f_3 = ab + bc + ad + cd,$$

$$f_4 = a + b + c + d.$$

We start with $G = \{f_4\}$ and $P_1 = \{(f_3, bf_4)\}$, thus $L_1 = \{f_3, bf_4\}$.
Let us do **symbolic preprocessing**:

$$\begin{aligned} T(L_1) &= \{ab, b^2, bc, ad, bd, cd, d^2\} \\ L_1 &= \{f_3, bf_4, df_4\} \end{aligned}$$

$b^2 \notin L(G)$, $bc \notin L(G)$, $d \text{lt}(f_4) = ad$, all others also $\notin L(G)$,

Example: Cyclic-4

Now reduction:

Convert polynomial data L_1 to Macaulay Matrix M_1

$$\begin{array}{ccccccc} & ab & b^2 & bc & ad & bd & cd & d^2 \\ df_4 & \left(\begin{array}{ccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{array} \right) \\ f_3 & \\ bf_4 & \end{array}$$

Example: Cyclic-4

Now reduction:

Convert polynomial data L_1 to Macaulay Matrix M_1

$$\begin{array}{ccccccc} & ab & b^2 & bc & ad & bd & cd & d^2 \\ df_4 & \left(\begin{array}{ccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array} \right) \\ f_3 & \left(\begin{array}{ccccccc} 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{array} \right) \\ bf_4 & \left(\begin{array}{ccccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{array} \right) \end{array}$$

Gaussian Elimination of M_1 :

$$\begin{array}{ccccccc} & ab & b^2 & bc & ad & bd & cd & d^2 \\ df_4 & \left(\begin{array}{ccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array} \right) \\ f_3 & \left(\begin{array}{ccccccc} 1 & 0 & 1 & 0 & -1 & 0 & -1 \end{array} \right) \\ bf_4 & \left(\begin{array}{ccccccc} 0 & 1 & 0 & 0 & 2 & 0 & 1 \end{array} \right) \end{array}$$

Example: Cyclic-4

Convert matrix data back to polynomial structure F_1 :

$$\begin{array}{c} ab \quad b^2 \quad bc \quad ad \quad bd \quad cd \quad d^2 \\ df_4 \left(\begin{array}{ccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ f_3 & 1 & 0 & 1 & 0 & -1 & 0 & -1 \\ bf_4 & 0 & 1 & 0 & 0 & 2 & 0 & 1 \end{array} \right) \end{array}$$

$$F_1 = \left\{ \underbrace{ad + bd + cd + d^2}_{f_5}, \underbrace{ab + bc - bd - d^2}_{f_6}, \underbrace{b^2 + 2bd + d^2}_{f_7} \right\}$$

Example: Cyclic-4

Convert matrix data back to polynomial structure F_1 :

$$\begin{array}{c} ab \quad b^2 \quad bc \quad ad \quad bd \quad cd \quad d^2 \\ df_4 \left(\begin{array}{ccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ f_3 & 1 & 0 & 1 & 0 & -1 & 0 & -1 \\ bf_4 & 0 & 1 & 0 & 0 & 2 & 0 & 1 \end{array} \right) \end{array}$$

$$F_1 = \left\{ \underbrace{ad + bd + cd + d^2}_{f_5}, \underbrace{ab + bc - bd - d^2}_{f_6}, \underbrace{b^2 + 2bd + d^2}_{f_7} \right\}$$

$\text{lt}(f_5), \text{lt}(f_6) \in L(G)$, so

$$\mathbf{G} \leftarrow \mathbf{G} \cup \{f_7\}.$$

Example: Cyclic-4

Next round:

$$G = \{t_4, t_7\}, P_2 = \{(t_2, bcf_4)\}, L_2 = \{t_2, bcf_4\}.$$

Example: Cyclic-4

Next round:

$$G = \{f_4, f_7\}, P_2 = \{(f_2, bcf_4)\}, L_2 = \{f_2, bcf_4\}.$$

We can simplify the computations:

$$\text{lt}(bcf_4) = abc = \text{lt}(cf_6).$$

f_6 possibly better reduced than f_4 . (f_6 is not in G !)

$$\implies L_2 = \{f_2, cf_6\}$$

Example: Cyclic-4

Next round:

$$G = \{f_4, f_7\}, P_2 = \{(f_2, bcf_4)\}, L_2 = \{f_2, bcf_4\}.$$

We can simplify the computations:

$$\text{lt}(bcf_4) = abc = \text{lt}(cf_6).$$

f_6 possibly better reduced than f_4 . (f_6 is not in G !)

$$\implies L_2 = \{f_2, cf_6\}$$

Symbolic preprocessing:

$$\begin{aligned} T(L_2) &= \{\textcolor{blue}{abc}, bc^2, abd, acd, bcd, cd^2\} \\ L_2 &= \{f_2, cf_6, \quad \} \end{aligned}$$

Example: Cyclic-4

Next round:

$$G = \{f_4, f_7\}, P_2 = \{(f_2, bcf_4)\}, L_2 = \{f_2, bcf_4\}.$$

We can simplify the computations:

$$\text{lt}(bcf_4) = abc = \text{lt}(cf_6).$$

f_6 possibly better reduced than f_4 . (f_6 is not in G !)

$$\implies L_2 = \{f_2, cf_6\}$$

Symbolic preprocessing:

$$\begin{aligned} T(L_2) &= \{\mathbf{abc}, \mathbf{bc}^2, abd, acd, bcd, cd^2\} \\ L_2 &= \{f_2, cf_6, \quad\} \end{aligned}$$

$$bc^2 \notin L(G),$$

Example: Cyclic-4

Next round:

$$G = \{f_4, f_7\}, P_2 = \{(f_2, bcf_4)\}, L_2 = \{f_2, bcf_4\}.$$

We can simplify the computations:

$$\text{lt}(bcf_4) = abc = \text{lt}(cf_6).$$

f_6 possibly better reduced than f_4 . (f_6 is not in G !)

$$\implies L_2 = \{f_2, cf_6\}$$

Symbolic preprocessing:

$$\begin{aligned} T(L_2) &= \{abc, bc^2, abd, acd, bcd, cd^2\} \\ L_2 &= \{f_2, cf_6, \quad\} \end{aligned}$$

$bc^2 \notin L(G)$, $abd = \text{lt}(bdf_4)$, but also $abd = \text{lt}(bf_5)$!

Example: Cyclic-4

Next round:

$$G = \{f_4, f_7\}, P_2 = \{(f_2, bcf_4)\}, L_2 = \{f_2, bcf_4\}.$$

We can simplify the computations:

$$\text{lt}(bcf_4) = abc = \text{lt}(cf_6).$$

f_6 possibly better reduced than f_4 . (f_6 is not in G !)

$$\implies L_2 = \{f_2, cf_6\}$$

Symbolic preprocessing:

$$\begin{aligned} T(L_2) &= \{\mathbf{abc}, \mathbf{bc}^2, \mathbf{abd}, acd, bcd, cd^2\} \\ L_2 &= \{f_2, cf_6, \} \end{aligned}$$

$bc^2 \notin L(G)$, $abd = \text{lt}(bdf_4)$, but also $abd = \text{lt}(bf_5)$!

Let us investigate this in more detail.

Interlude – Simplify

Idea

Try to replace $u \cdot f$ by a product $(wv) \cdot g$ where vg corresponds to an already computed row in the Gauss. Elim. of a previous matrix M_i .
⇒ Reuse rows that are reduced but not “in” G .

Interlude – Simplify

Idea

Try to replace $u \cdot f$ by a product $(wv) \cdot g$ where vg corresponds to an already computed row in the Gauss. Elim. of a previous matrix M_i .

⇒ Reuse rows that are reduced but not “in” G .

Input: monomial u , polynomial f , list \mathcal{F} of old F_i (from M_i after Gauss. Elim.)

Output: product $v \cdot g$ replacing $u \cdot f$

Interlude – Simplify

Idea

Try to replace $u \cdot f$ by a product $(wv) \cdot g$ where vg corresponds to an already computed row in the Gauss. Elim. of a previous matrix M_i .

⇒ Reuse rows that are reduced but not “in” G .

Input: monomial u , polynomial f , list \mathcal{F} of old F_i (from M_i after Gauss. Elim.)

Output: product $v \cdot g$ replacing $u \cdot f$

1. $d \leftarrow$ current index in the F4 algorithm
2. $D(u) \leftarrow \{\text{list of divisors of } u\}$
3. for $w \in D(u)$
 - (a) if $\exists j \in \{1, \dots, d-1\}$ such that $w \cdot f$ corresponds to row in M_j
 - ▷ $\exists_1 g \in F_j$ such that $\text{lt}(g) = \text{lt}(w \cdot f)$
 - ▷ if $w \neq u$: Return **Simplify** $(\frac{u}{w}, g, \mathcal{F})$ (recursive call)
 - ▷ else: Return $1 \cdot g$
4. else: Return $u \cdot f$

Note

- ▶ Tries to reuse all rows from old matrices.
⇒ We need to keep them in memory.
- ▶ We also simplify generators of S-pairs, as we have done in our example: $(f_2, bcf_4) \implies (f_2, cf_6)$.
- ▶ One can also choose “better” reducers by other properties, not only “last reduced one”.
- ▶ Without **Simplify** the F4 algorithm is rather slow.

Note

- ▶ Tries to reuse all rows from old matrices.
⇒ We need to keep them in memory.
- ▶ We also simplify generators of S-pairs, as we have done in our example: $(f_2, bcf_4) \implies (f_2, cf_6)$.
- ▶ One can also choose “better” reducers by other properties, not only “last reduced one”.
- ▶ Without **Simplify** the F4 algorithm is rather slow.

In our example:
Choose bf_5 as reducer, not bdf_4 .

Example: Cyclic-4

Symbolic preprocessing - now with **simplify**:

$$\begin{aligned} T(L_2) &= \{abc, bc^2, abd, acd, bcd, cd^2\} \\ L_2 &= \{f_2, cf_6\} \end{aligned}$$

$$bc^2 \notin L(G),$$

Example: Cyclic-4

Symbolic preprocessing - now with **simplify**:

$$\begin{aligned} T(L_2) &= \{\mathbf{abc}, \mathbf{bc}^2, \mathbf{abd}, \mathbf{acd}, \mathbf{bcd}, \mathbf{cd}^2\} \\ L_2 &= \{f_2, cf_6\} \end{aligned}$$

$bc^2 \notin L(G)$, $abd = \text{lt}(bf_5)$,

Example: Cyclic-4

Symbolic preprocessing - now with **simplify**:

$$\begin{aligned} T(L_2) &= \{\mathbf{abc}, \mathbf{bc}^2, \mathbf{abd}, \mathbf{acd}, \mathbf{bcd}, \mathbf{cd}^2, \mathbf{b}^2\mathbf{d}, \mathbf{c}^2\mathbf{d} \quad \} \\ L_2 &= \{f_2, cf_6, \mathbf{bf}_5 \quad \} \end{aligned}$$

$$bc^2 \notin L(G), abd = \text{lt}(bf_5),$$

Example: Cyclic-4

Symbolic preprocessing - now with **simplify**:

$$\begin{aligned} T(L_2) &= \{abc, bc^2, abd, acd, bcd, cd^2, b^2d, c^2d, \dots\} \\ L_2 &= \{f_2, cf_6, bf_5, cf_5, df_7\} \end{aligned}$$

$bc^2 \notin L(G)$, $abd = \text{lt}(bf_5)$, and so on.

Example: Cyclic-4

Symbolic preprocessing - now with **simplify**:

$$\begin{aligned}T(L_2) &= \{abc, bc^2, abd, acd, bcd, cd^2, b^2d, c^2d, \dots\} \\L_2 &= \{f_2, cf_6, bf_5, cf_5, df_7\}\end{aligned}$$

$bc^2 \notin L(G)$, $abd = \text{lt}(bf_5)$, and so on.

Now try to exploit the special structure of the Macaulay matrices.

Improve Gaussian Elimination

Use **Linear Algebra** for reduction steps in GB computations.

Improve Gaussian Elimination

Use **Linear Algebra** for reduction steps in GB computations.

$$\begin{matrix} 1 & 3 & 0 & 0 & 7 & 1 & 0 \\ 1 & 0 & 4 & 1 & 0 & 0 & 5 \\ 0 & 1 & 6 & 0 & 8 & 0 & 1 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 3 & 1 \end{matrix}$$

Improve Gaussian Elimination

Use **Linear Algebra** for reduction steps in GB computations.

$$\begin{matrix} 1 & 3 & 0 & 0 & 7 & 1 & 0 \\ 1 & 0 & 4 & 1 & 0 & 0 & 5 \\ 0 & 1 & 6 & 0 & 8 & 0 & 1 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 3 & 1 \end{matrix}$$

Knowledge of underlying GB structure

Improve Gaussian Elimination

Use **Linear Algebra** for reduction steps in GB computations.

$$\begin{array}{ll} \text{S-pair} & \left\{ \begin{array}{ccccccc} 1 & 3 & 0 & 0 & 7 & 1 & 0 \\ 1 & 0 & 4 & 1 & 0 & 0 & 5 \end{array} \right. \\ \text{S-pair} & \left\{ \begin{array}{ccccccc} 0 & 1 & 6 & 0 & 8 & 0 & 1 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \end{array} \right. \\ \text{reducer} & \leftarrow \begin{array}{ccccccc} 0 & 0 & 0 & 0 & 1 & 3 & 1 \end{array} \end{array}$$

Knowledge of underlying GB structure

Improve Gaussian Elimination

Use **Linear Algebra** for reduction steps in GB computations.

$$\begin{array}{l} \text{S-pair} \\ \left\{ \begin{array}{ccccccc} 1 & 3 & 0 & 0 & 7 & 1 & 0 \\ 1 & 0 & 4 & 1 & 0 & 0 & 5 \end{array} \right. \\ \text{S-pair} \\ \left\{ \begin{array}{ccccccc} 0 & 1 & 6 & 0 & 8 & 0 & 1 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \end{array} \right. \\ \text{reducer} \quad \leftarrow \quad \begin{array}{ccccccc} 0 & 0 & 0 & 0 & 1 & 3 & 1 \end{array} \end{array}$$

Knowledge of underlying GB structure

Improve Gaussian Elimination

Use **Linear Algebra** for reduction steps in GB computations.

$$\begin{array}{c} \text{S-pair} \\ \left\{ \begin{array}{ccccccc} 1 & 3 & 0 & 0 & 7 & 1 & 0 \\ 1 & 0 & 4 & 1 & 0 & 0 & 5 \end{array} \right. \\ \text{S-pair} \\ \left\{ \begin{array}{ccccccc} 0 & 1 & 6 & 0 & 8 & 0 & 1 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \end{array} \right. \\ \text{reducer} \quad \leftarrow \quad \begin{array}{ccccccc} 0 & 0 & 0 & 0 & 1 & 3 & 1 \end{array} \end{array}$$

Knowledge of underlying GB structure

Idea

Do a static **reordering before** the Gaussian Elimination to achieve a better initial shape. **Reorder afterwards.**

Faugère-Lachartre Idea

1st step: Sort pivot and non-pivot columns

1	3	0	0	7	1	0
1	0	4	1	0	0	5
0	1	6	0	8	0	1
0	5	0	0	0	2	0
0	0	0	0	1	3	1

Faugère-Lachartre Idea

1st step: Sort pivot and non-pivot columns

1	3	0	0	7	1	0
1	0	4	1	0	0	5
0	1	6	0	8	0	1
0	5	0	0	0	2	0
0	0	0	0	1	3	1



Faugère-Lachartre Idea

1st step: Sort pivot and non-pivot columns

1	3	0	0	7	1	0
1	0	4	1	0	0	5
0	1	6	0	8	0	1
0	5	0	0	0	2	0
0	0	0	0	1	3	1

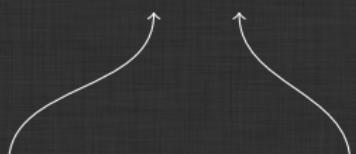


Faugère-Lachartre Idea

1st step: Sort pivot and non-pivot columns

1	3	0	0	7	1	0
1	0	4	1	0	0	5
0	1	6	0	8	0	1
0	5	0	0	0	2	0
0	0	0	0	1	3	1

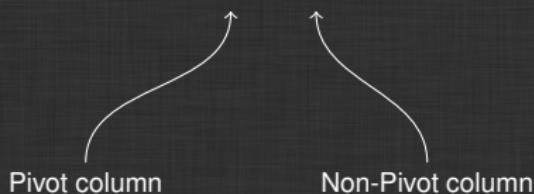
Pivot column Non-Pivot column



Faugère-Lachartre Idea

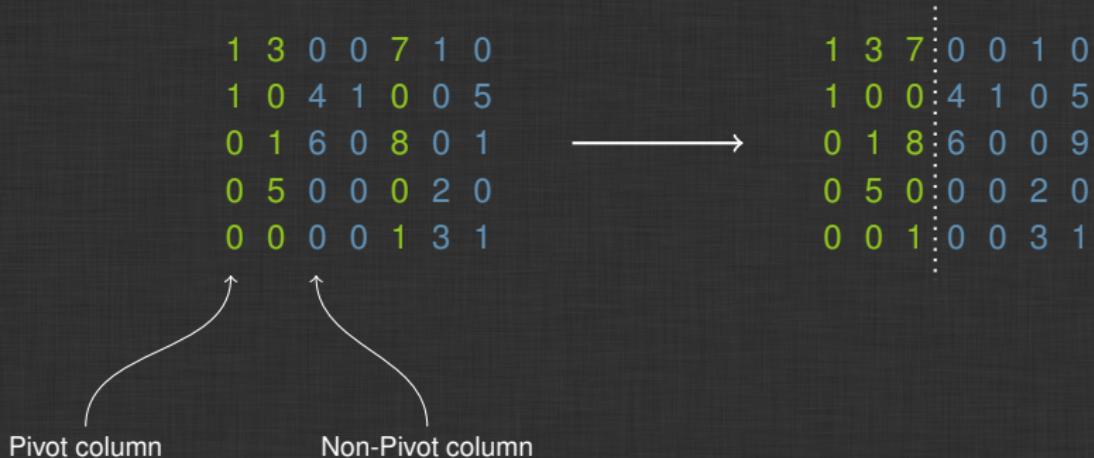
1st step: Sort pivot and non-pivot columns

1	3	0	0	7	1	0
1	0	4	1	0	0	5
0	1	6	0	8	0	1
0	5	0	0	0	2	0
0	0	0	0	1	3	1



Faugère-Lachartre Idea

1st step: Sort pivot and non-pivot columns



Faugère-Lachartre Idea

2nd step: Sort pivot and non-pivot rows

1	3	7	0	0	1	0
1	0	0	4	1	0	5
0	1	8	6	0	0	9
0	5	0	0	0	2	0
0	0	1	0	0	3	1

Faugère-Lachartre Idea

2nd step: Sort pivot and non-pivot rows

1	3	7	0	0	1	0
1	0	0	4	1	0	5
0	1	8	6	0	0	9
0	5	0	0	0	2	0
0	0	1	0	0	3	1

Pivot row

Faugère-Lachartre Idea

2nd step: Sort pivot and non-pivot rows

	1	3	7	0	0	1	0
	1	0	0	4	1	0	5
	0	1	8	6	0	0	9
	0	5	0	0	0	2	0
	0	0	1	0	0	3	1

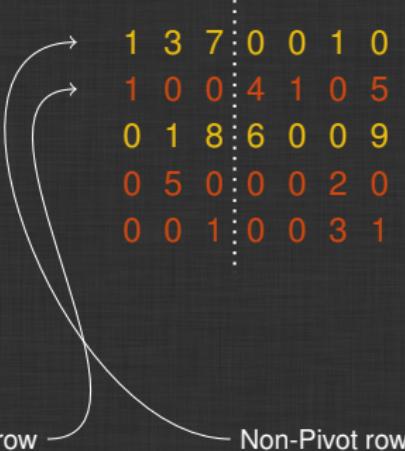
Pivot row Non-Pivot row

Faugère-Lachartre Idea

2nd step: Sort pivot and non-pivot rows

	1	3	7	0	0	1	0
	1	0	0	4	1	0	5
	0	1	8	6	0	0	9
	0	5	0	0	0	2	0
	0	0	1	0	0	3	1

Pivot row Non-Pivot row



Faugère-Lachartre Idea

2nd step: Sort pivot and non-pivot rows

$$\begin{array}{cccc|ccccc} 1 & 3 & 7 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 4 & 1 & 0 & 5 \\ 0 & 1 & 8 & 6 & 0 & 0 & 9 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 1 \end{array} \longrightarrow \begin{array}{cccc|ccccc} 1 & 0 & 0 & 4 & 1 & 0 & 5 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 1 \\ 1 & 3 & 7 & 0 & 0 & 1 & 0 \\ 0 & 1 & 8 & 6 & 0 & 0 & 9 \end{array}$$

Pivot row Non-Pivot row

Faugère-Lachartre Idea

3rd step: Reduce lower left part to zero

1	0	0	4	1	0	5
0	5	0	0	0	2	0
0	0	1	0	0	3	1
1	3	7	0	0	1	0
0	1	8	6	0	0	9

Faugère-Lachartre Idea

3rd step: Reduce lower left part to zero

$$\begin{array}{cc} \begin{matrix} 1 & 0 & 0 & 4 & 1 & 0 & 5 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 1 \\ 1 & 3 & 7 & 0 & 0 & 1 & 0 \\ 0 & 1 & 8 & 6 & 0 & 0 & 9 \end{matrix} & \xrightarrow{\hspace{10em}} & \begin{matrix} 1 & 0 & 0 & 4 & 1 & 0 & 5 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 7 & 10 & 3 & 10 \\ 0 & 0 & 0 & 6 & 0 & 2 & 1 \end{matrix} \end{array}$$

Faugère-Lachartre Idea

4th step: Reduce lower right part

1	0	0	4	1	0	5
0	5	0	0	0	2	0
0	0	1	0	0	3	1
0	0	0	7	10	3	10
0	0	0	6	0	2	1

Faugère-Lachartre Idea

4th step: Reduce lower right part

$$\begin{array}{cc|ccccc} 1 & 0 & 0 & 4 & 1 & 0 & 5 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 1 \\ \hline 0 & 0 & 0 & 7 & 10 & 3 & 10 \\ 0 & 0 & 0 & 6 & 0 & 2 & 1 \end{array} \longrightarrow \begin{array}{cc|ccccc} 1 & 0 & 0 & 4 & 1 & 0 & 5 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 1 \\ \hline 0 & 0 & 0 & 7 & 10 & 3 & 10 \\ 0 & 0 & 0 & 0 & 4 & 1 & 5 \end{array}$$

Faugère-Lachartre Idea

4th step: Reduce lower right part

$$\begin{array}{cc|ccccc} 1 & 0 & 0 & 4 & 1 & 0 & 5 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 1 \\ \hline 0 & 0 & 0 & 7 & 10 & 3 & 10 \\ 0 & 0 & 0 & 6 & 0 & 2 & 1 \end{array} \longrightarrow \begin{array}{cc|ccccc} 1 & 0 & 0 & 4 & 1 & 0 & 5 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 1 \\ \hline 0 & 0 & 0 & 7 & 10 & 3 & 10 \\ 0 & 0 & 0 & 0 & 4 & 1 & 5 \end{array}$$

5th step: Remap columns of lower right part

How our matrices look like

Some data about the matrix:

- ▶ F_4 computation of homogeneous KATSURA-12, degree 6 matrix

How our matrices look like

Some data about the matrix:

- ▶ F_4 computation of homogeneous KATSURA-12, degree 6 matrix
- ▶ Size 137MB

How our matrices look like

Some data about the matrix:

- ▶ F_4 computation of homogeneous KATSURA-12, degree 6 matrix
- ▶ Size 137MB
- ▶ 24,006,869 nonzero elements (density: 5%)

How our matrices look like

Some data about the matrix:

- ▶ F_4 computation of homogeneous KATSURA-12, degree 6 matrix
- ▶ Size 137MB
- ▶ 24,006,869 nonzero elements (density: 5%)
- ▶ Dimensions:

full matrix: 21,182 × 22,207

upper-left: 17,915 × 17,915

lower-left: 3,267 × 17,915

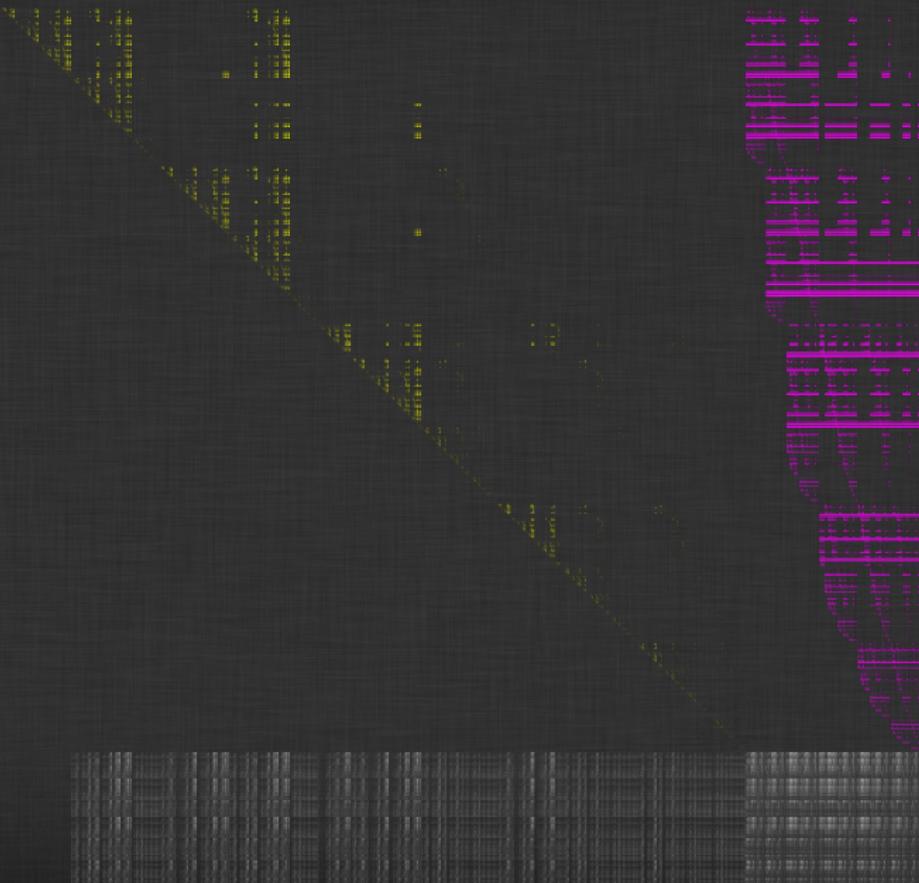
upper-right: 17,915 × 4,292

lower-right: 3,267 × 4,292

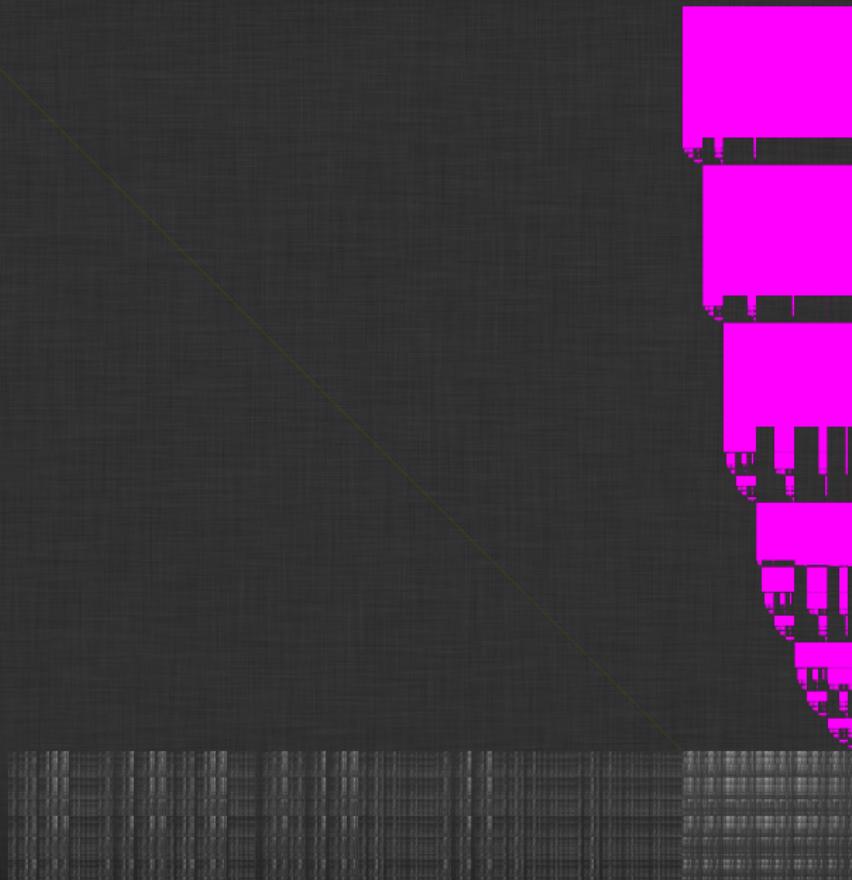
How our matrices look like (2)

How our matrices look like (3)

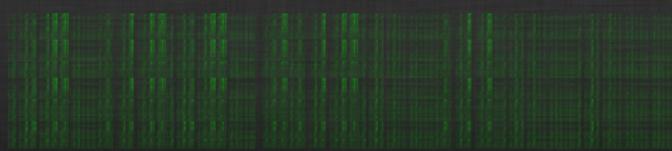
Hybrid Matrix Multiplication $A^{-1}B$



Hybrid Matrix Multiplication $A^{-1}B$



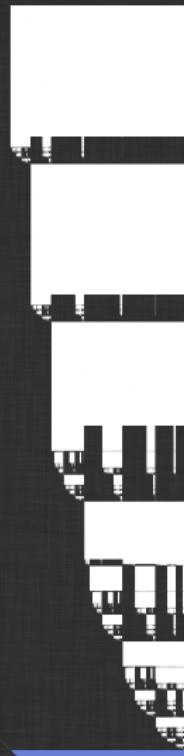
Reduce C to zero



Gaussian Elimination on D



New information



First attempts

2010 – UPMC Paris 6, INRIA PolSys Team
Jean-Charles Faugère & Sylvain Lachartre – **FL**

2011 – University of Kaiserslautern
Bradford Hovinen – **LELA**
<https://github.com/Singular/LELA>

2012 – UPMC Paris 6, INRIA PolSys Team
Fayssal Martani – **new implementation in LELA**
<https://github.com/martani/LELA>

2012-2013 – University of Kaiserslautern
Bjarke Hammersholt Roune – **MathicGB**
<https://github.com/broune/mathicgb>

2012-2014 – University of Passau
Severin Neumann – **parallelGBC**
<https://github.com/svrnm/parallelGBC>

References I

- [1] Albrecht, M. and Perry, J. F4/5. <http://arxiv.org/abs/1006.4933>, 2010.
- [2] Arri, A. and Perry, J. The F5 Criterion revised. *Journal of Symbolic Computation*, 46(2):1017–1029, June 2011. Preprint online at arxiv.org/abs/1012.3664.
- [3] Ars, G. *Applications des bases de Gröbner à la cryptographie*. PhD thesis, Université de Rennes I, 2005.
- [4] Ars, G. and Hashemi, A. Extended F5 Criteria. *Journal of Symbolic Computation, MEGA 2009 special issue*, 45(12):1330–1340, 2010.
- [5] Buchberger, B. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, University of Innsbruck, 1965.
- [6] Buchberger, B. Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems. *Aequ. Math.*, 4(3):374–383, 1970.
- [7] Buchberger, B. A criterion for detecting unnecessary reductions in the construction of Gröbner bases. In *EUROSAM '79, An International Symposium on Symbolic and Algebraic Manipulation*, volume 72 of *Lecture Notes in Computer Science*, pages 3–21. Springer, 1979.
- [8] Buchberger, B. Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory. pages 184–232, 1985.

References II

- [9] Decker, W., Greuel, G.-M., Pfister, G., and Schönemann, H. **SINGULAR 4-0-0 — A computer algebra system for polynomial computations**, 2014.
<http://www.singular.uni-kl.de>.
- [10] Eder, C. Improving incremental signature-based Groebner bases algorithms. *ACM SIGSAM Communications in Computer Algebra*, 47(1):1–13, 2013.
<http://arxiv.org/abs/1201.6472>.
- [11] Eder, C. Predicting zero reductions in Gröbner basis computations. submitted to Journal of Symbolic Computation, preprint at <http://arxiv.org/abs/1404.0161>, 2014.
- [12] Eder, C. and Faugère, J.-C. **A survey on signature-based Groebner basis algorithms**, 2014.
<http://arxiv.org/abs/1404.1774>
- [13] Eder, C., Gash, J., and Perry, J. Modifying Faugère's F5 Algorithm to ensure termination. *ACM SIGSAM Communications in Computer Algebra*, 45(2):70–89, 2011.
<http://arxiv.org/abs/1006.0318>.
- [14] Eder, C. and Perry, J. F5C: A Variant of Faugère's F5 Algorithm with reduced Gröbner bases. *Journal of Symbolic Computation, MEGA 2009 special issue*, 45(12):1442–1458, 2010. [dx.doi.org/10.1016/j.jsc.2010.06.019](https://doi.org/10.1016/j.jsc.2010.06.019).
- [15] Eder, C. and Perry, J. Signature-based Algorithms to Compute Gröbner Bases. In *ISSAC 2011: Proceedings of the 2011 international symposium on Symbolic and algebraic computation*, pages 99–106, 2011.

References III

- [16] Eder, C. and Roune, B. H. Signature Rewriting in Gröbner Basis Computation. In *ISSAC 2013: Proceedings of the 2013 international symposium on Symbolic and algebraic computation*, pages 331–338, 2013.
- [17] Faugère, J.-C. A new efficient algorithm for computing Gröbner bases without reduction to zero F5. In *ISSAC'02, Villeneuve d'Ascq, France*, pages 75–82, July 2002. Revised version from <http://fgbtrs.lip6.fr/jcf/Publications/index.html>.
- [18] Faugère, J.-C. and Joux, A. Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases. *2729:44–60*, 2003.
- [19] Faugère, J.-C., Safey El Din, M., and Spaenlehauer, P.-J. Gröbner Bases of Bihomogeneous Ideals Generated by Polynomials of Bidegree (1,1): Algorithms and Complexity. *Journal of Symbolic Computation*, 46(4):406–437, 2011. Available online 4 November 2010.
- [20] Faugère, J.-C., Safey El Din, M., and Verron, T. On the complexity of Computing Gröbner Bases for Quasi-homogeneous Systems. In *Proceedings of the 38th international symposium on International symposium on symbolic and algebraic computation*, ISSAC '13, pages 189–196, New York, NY, USA, 2013. ACM.
- [21] Faugère, J.-C., Spaenlehauer, P.-J. and Svartz, J. Sparse Gröbner Bases: the unmixed Case. In *Proceedings of the 39th international symposium on International symposium on symbolic and algebraic computation*, ISSAC '14, Kobe, Japan, 2014.

References IV

- [22] Faugère, J.-C. and Svartz, J. Solving polynomial systems globally invariant under an action of the symmetric group and application to the equilibria of n vertices in the plane. In *Proceedings of the 37th international symposium on International symposium on symbolic and algebraic computation*, ISSAC '12, pages 170–178, New York, NY, USA, 2012. ACM.
- [23] Faugère, J.-C. and Svartz, J. Gröbner Bases of ideals invariant under a Commutative group : the Non-modular Case. In *Proceedings of the 38th international symposium on International symposium on symbolic and algebraic computation*, ISSAC '13, pages 347–354, New York, NY, USA, 2013. ACM.
- [24] Faugère, J.-C. and Rahmany, S. Solving systems of polynomial equations with symmetries using SAGBI-Gröbner bases. In *ISSAC '09: Proceedings of the 2009 international symposium on Symbolic and algebraic computation*, ISSAC '09, pages 151–158, New York, NY, USA, 2009. ACM.
- [25] Galkin, V. Simple signature-based Groebner basis algorithm.
<http://arxiv.org/abs/1205.6050>, 2012.
- [26] Gao, S., Guan, Y., and Volny IV, F. A new incremental algorithm for computing Gröbner bases. In *ISSAC '10: Proceedings of the 2010 international symposium on Symbolic and algebraic computation*, pages 13–19. ACM, 2010.
- [27] Gao, S., Volny IV, F., and Wang, D. A new algorithm for computing Groebner bases.
<http://eprint.iacr.org/2010/641>, 2010.

References V

- [28] Gao, S., Volny IV, F., and Wang, D. A new algorithm for computing Groebner bases (rev. 2011). <http://www.math.clemson.edu/~sgao/papers/gvw.pdf>, 2011.
- [29] Gao, S., Volny IV, F., and Wang, D. A new algorithm for computing Groebner bases (rev. 2013).
http://www.math.clemson.edu/~sgao/papers/gvw_R130704.pdf, 2013.
- [30] Gash, J. M. *On efficient computation of Gröbner bases*. PhD thesis, University of Indiana, Bloomington, IN, 2008.
- [31] Gash, J. M. A provably terminating and speed-competitive variant of F5 – F5t. *submitted to the Journal of Symbolic Computation*, 2009.
- [32] Gebauer, R. and Möller, H. M. On an installation of Buchberger's algorithm. *Journal of Symbolic Computation*, 6(2-3):275–286, October/December 1988.
- [33] Gerdt, V. P. and Hashemi, A. On the use of Buchberger criteria in G2V algorithm for calculating Gröbner bases. *Program. Comput. Softw.*, 39(2):81–90, March 2013.
- [34] Gerdt, V. P., Hashemi, A., and M.-Alizadeh, B. Involutive Bases Algorithm Incorporating F5 Criterion. *J. Symb. Comput.*, 59:1–20, 2013.
- [35] Huang, L. A new conception for computing Gröbner basis and its applications.
<http://arxiv.org/abs/1012.5425>, 2010.

References VI

- [36] Moreno-Socías, G. Degrevlex Gröbner bases of generic complete intersections. *Journal of Pure and Applied Algebra*, 180(3):263 – 283, 2003.
- [37] Pan, S., Hu, Y., and Wang, B. The Termination of Algorithms for Computing Gröbner Bases. <http://arxiv.org/abs/1202.3524>, 2012.
- [38] Pan, S., Hu, Y., and Wang, B. The Termination of the F5 Algorithm Revisited. In *ISSAC 2013: Proceedings of the 2013 international symposium on Symbolic and algebraic computation*, pages 291–298, 2013.
- [39] Roune, B. H. and Stillman, M. Practical Gröbner Basis Computation. In *ISSAC 2012: Proceedings of the 2012 international symposium on Symbolic and algebraic computation*, 2012.
- [40] Stegers, T. Faugère's F5 Algorithm revisited. Master's thesis, Technische Universität Darmstadt, revised version 2007.
- [41] Sun, Y. and Wang, D. K. A generalized criterion for signature related Gröbner basis algorithms. In *ISSAC 2011: Proceedings of the 2011 international symposium on Symbolic and algebraic computation*, pages 337–344, 2011.
- [42] Sun, Y., Wang, D. K., Ma, D. X., and Zhang, Y. A signature-based algorithm for computing Gröbner bases in solvable polynomial algebras. In *ISSAC 2012: Proceedings of the 2011 international symposium on Symbolic and algebraic computation*, pages 351–358, 2012.

References VII

- [43] Volny, F. *New algorithms for computing Gröbner bases*. PhD thesis, Clemson University, 2011.