Efficient Gröbner Bases computations

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

Oberseminar am Lehrstuhl Johannes Buchmann
TU Darmstadt

February 19, 2015

Technische Universität
Kaiserslautern

Conventions

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

$$:\Longleftrightarrow$$

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

Why?

1. A lot of crypto systems boil down to find a solution (a finite number of solutions) of a system of polynomial equations.

Why?

1. A lot of crypto systems boil down to find a solution (a finite number of solutions) of a system of polynomial equations.

2. For example, multivariate crypto systems like (Multi-)HFE(+-), UOV or Rainbow

3. Minrank (n, k, r) problem: Given matrices $M_0, \dots, M_k \in \mathscr{M}_{n\times n}(\mathscr{K})$, find (if possible) $(\lambda_1, \dots, \lambda_k) \in \mathscr{K}^k$ such that

$$\operatorname{rank}\left(\sum_{i=1}^{k} \lambda_i M_i - M_0\right) \leq r.$$

Why?

1. A lot of crypto systems boil down to find a solution (a finite number of solutions) of a system of polynomial equations.

2. For example, multivariate crypto systems like (Multi-)HFE(+-), UOV or Rainbow

3. Minrank (n, k, r) problem: Given matrices $M_0, \ldots, M_k \in \mathscr{M}_{n \times n}(\mathscr{K})$, find (if possible) $(\lambda_1, \ldots, \lambda_k) \in \mathscr{K}^k$ such that

$$\operatorname{rank}\left(\sum_{i=1}^{k} \lambda_i M_i - M_0\right) \leq r.$$

Solving polynomial equations is important
Gröbner Bases are cool!

S-polynomials

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

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

S-polynomials

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

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

Buchberger's criterion [1]

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

Buchberger's algorithm

Input: Ideal $I = \langle f_1, \ldots, f_m \rangle$
Output: Gröbner basis G for I

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

Buchberger's algorithm

Input: Ideal $I = \langle f_1, \ldots, f_m \rangle$
Output: Gröbner basis G for I

1. $G \leftarrow \emptyset$
2. $G \leftarrow G \cup \{f_i\}$ for all $i \in \{1, \ldots, m\}$
3. Set $P \leftarrow \{\mathrm{spol}(f_i, f_j) \mid f_i, f_j \in G, i > j\}$
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 $P \leftarrow \{\mathrm{spol}(f_i, f_j) \mid f_i, f_j \in G, i > j\}$
4. Choose $p \in P$, $P \leftarrow P \setminus \{p\}$
   (a) If $p \xrightarrow{G} 0$ ▶▶ **no new information**
   Go on with the next element in P.
   (b) If $p \xrightarrow{G} q \neq 0$ ▶▶ **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 $\mathbb{Q} \to$ several $\mathbb{Z}_{p_i}$ computations and CRT
- Predict zero reductions fast checks $\to$ fewer useless reductions
- Sort pair set selection of pairs, degree drops, mutants, etc.
- Homogenization d-Gröbner bases, sugar degree
- Change of order transformation to different monomial order
- Linear Algebra (specialized) Gaussian Elimination
- Sparse Gröbner Bases exploitation of sparsity, Newton polygons
- ...

How to improve computations?

- Predict zero reductions fast checks $\rightarrow$ fewer useless reductions

- Linear Algebra (specialized) Gaussian Elimination

- **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

$$\boxed{\mathbf{g_1 = xy - z^2}, \quad \mathbf{g_2 = y^2 - z^2}}$$

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

$$\boxed{\mathbf{g_1 = xy - z^2}, \quad \mathbf{g_2 = y^2 - z^2}}$$

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

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

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

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

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

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

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

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

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

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

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

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

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?

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

How to detect zero reductions in advance?

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

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

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

$$\begin{aligned}&\Longrightarrow \mathrm{lt}\left(g_2\right) \mathrm{lot}\left(g_3\right) - \mathbf{g_2} \mathrm{lot}\left(\mathbf{g_3}\right) - \mathrm{lt}\left(g_3\right) \mathrm{lot}\left(g_2\right)\\ &= -\mathrm{lot}\left(g_2\right) \mathrm{lot}\left(g_3\right) - \mathrm{lt}\left(g_3\right) \mathrm{lot}\left(g_2\right)\\ &= -g_3 \mathrm{lot}\left(g_2\right).\end{aligned}$$

How to detect zero reductions in advance?

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

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

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

$$\begin{aligned}&\Longrightarrow \operatorname{lt}(g_2)\operatorname{lot}(g_3) - \mathbf{g_2}\operatorname{lot}(\mathbf{g_3}) - \operatorname{lt}(g_3)\operatorname{lot}(g_2)\\ &= -\operatorname{lot}(g_2)\operatorname{lot}(g_3) - \operatorname{lt}(g_3)\operatorname{lot}(g_2)\\ &= -g_3\operatorname{lot}(g_2).\end{aligned}$$

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

Buchberger's criteria

Product criterion [2]

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

Product criterion [2]

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

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

Product criterion [2]

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

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

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

Buchberger's criteria

Product criterion [2]

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

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

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

$\Longrightarrow$ We can rewrite $\operatorname{spol}(g_3, g_2)$:

$$\operatorname{spol}(g_3, g_2) = y \underbrace{\operatorname{spol}(g_3, g_1)}_{\xrightarrow{G} 0} - z^2 \underbrace{\operatorname{spol}(g_2, g_1)}_{\xrightarrow{G} -g_3} = y\left(yg_3 - z^2 g_1\right) - z^2\left(xg_2 - yg_1\right)$$

Buchberger's criteria

Product criterion [2]

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

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

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

$\Longrightarrow$ We can rewrite $\operatorname{spol}(g_3, g_2)$:

$$\operatorname{spol}(g_3,g_2) = y\underbrace{\operatorname{spol}(g_3,g_1)}_{\xrightarrow{G} 0} - z^2\underbrace{\operatorname{spol}(g_2,g_1)}_{\xrightarrow{G} -g_3} = y\left(yg_3 - z^2g_1\right) - z^2\left(xg_2 - yg_1\right)$$

Once we have reduced $\operatorname{spol}(g_2, g_1)$ and $\operatorname{spol}(g_3, g_1)$ we do not need to reduce $\operatorname{spol}(g_3, g_2)$.

Chain criterion [3]

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

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

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

Buchberger's criteria

Chain criterion [3]

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

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

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

Note

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

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

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

Buchberger's criteria

Chain criterion [3]

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

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

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

Note

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

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

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

Combine both criteria using Gebauer-Möller's installation [8].

Can we do even better?

In our example we still need to consider

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

How to get rid of this useless computation?

Can we do even better?

In our example we still need to consider

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

How to get rid of this useless computation?

Use more structure of $I \Longrightarrow$ **Signatures**

Signatures

Let $I = \langle f_1, \ldots, f_m \rangle \subset \mathscr{R}$.

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

Signatures

Let $I = \langle f_1, \ldots, f_m \rangle \subset \mathscr{R}$.
Idea: Give each $f \in I$ a bit more structure:

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

Signatures

Let $I = \langle f_1, \ldots, f_m \rangle \subset \mathscr{R}$.
Idea: Give each $f \in I$ a bit more structure:

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

2. Let $\alpha \mapsto \overline{\alpha} : \mathscr{R}^m \to \mathscr{R}$ such that $\overline{e}_i = f_i$ for all i.

Signatures

Let $I = \langle f_1, \ldots, f_m \rangle \subset \mathscr{R}$.

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

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

Signatures

Let $I = \langle f_1, \ldots, f_m \rangle \subset \mathscr{R}$.
Idea: Give each $f \in I$ a bit more structure:

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

2. Let $\alpha \mapsto \overline{\alpha} : \mathscr{R}^m \to \mathscr{R}$ such that $\overline{e}_i = f_i$ for all i.

3. Each $f \in I$ can be represented via some $\alpha \in \mathscr{R}^m$: $f = \overline{\alpha}$

4. A **signature** of f is given by $\mathfrak{s}(f) = \mathrm{lt}_{\prec}(\alpha)$ where $f = \overline{\alpha}$.

Signatures

Let $I = \langle f_1, \ldots, f_m \rangle \subset \mathscr{R}$.
Idea: Give each $f \in I$ a bit more structure:

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

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

$$g_1 = xy - z^2,\ \mathfrak{s}(g_1) = e_1,$$
$$g_2 = y^2 - z^2,\ \mathfrak{s}(g_2) = e_2.$$

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

$$g_1 = xy - z^2, \ \mathfrak{s}(g_1) = e_1,$$
$$g_2 = y^2 - z^2, \ \mathfrak{s}(g_2) = e_2.$$

$$g_3 = \operatorname{spol}(g_2, g_1) = xg_2 - yg_1$$
$$\Rightarrow \mathfrak{s}(g_3) = x\,\mathfrak{s}(g_2) = xe_2.$$

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

$$g_1 = xy - z^2,\ \mathfrak{s}(g_1) = e_1,$$
$$g_2 = y^2 - z^2,\ \mathfrak{s}(g_2) = e_2.$$

$$g_3 = \operatorname{spol}(g_2, g_1) = xg_2 - yg_1$$
$$\Rightarrow \mathfrak{s}(g_3) = x\,\mathfrak{s}(g_2) = xe_2.$$

$$\operatorname{spol}(g_3, g_1) = yg_3 - z^2 g_1$$
$$\Rightarrow \mathfrak{s}(\operatorname{spol}(g_3, g_1)) = y\,\mathfrak{s}(g_3) = xye_2.$$

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

$$g_1 = xy - z^2, \; \mathfrak{s}(g_1) = e_1,$$
$$g_2 = y^2 - z^2, \; \mathfrak{s}(g_2) = e_2.$$

$$g_3 = \operatorname{spol}(g_2, g_1) = xg_2 - yg_1$$
$$\Rightarrow \mathfrak{s}(g_3) = x\,\mathfrak{s}(g_2) = xe_2.$$

$$\operatorname{spol}(g_3, g_1) = yg_3 - z^2 g_1$$
$$\Rightarrow \mathfrak{s}\left(\operatorname{spol}(g_3, g_1)\right) = y\,\mathfrak{s}(g_3) = xye_2.$$

Note that $\mathfrak{s}\left(\operatorname{spol}(g_3, g_1)\right) = xy\,e_2$ and $\operatorname{lm}(g_1) = xy$.

Think in the module

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

Think in the module

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

S-pairs/S-polynomials:

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

Think in the module

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

S-pairs/S-polynomials:

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

$\mathfrak{s}$-reductions:

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

Think in the module

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

S-pairs/S-polynomials:

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

$\mathfrak{s}$-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

$\mathfrak{s}(\alpha) = \mathfrak{s}(\beta) \implies$ 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 relatons $\prec \mathfrak{s}(\alpha)$ are known:

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

$\square$

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

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

What are all possible configurations to reach signature T?

Signature-based criteria

S-pairs in signature T

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

What are all possible configurations to reach signature T?

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

Special cases

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

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 $\trianglelefteq$.

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 $\Longrightarrow$ Go on to next signature.

Special cases

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

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

Revisiting our example with $\prec_{\text{pot}}$

$$\mathfrak{s}\left(\operatorname{spol}(g_3, g_1)\right) = xye_2$$

$$\left.\begin{array}{l} g_1 = xy - z^2 \\ g_2 = y^2 - z^2 \end{array}\right\} \Rightarrow \operatorname{psyz}(g_2, g_1) = g_1 e_2 - g_2 e_1 = xye_2 + \dots$$

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

Benchmark	STD	SBA $\prec_{\text{pot}}$	SBA $\prec_{\text{d-pot}}$	SBA $\prec_{\text{lt}}$
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 $\prec_{\text{pot}}$	SBA $\prec_{\text{d-pot}}$	SBA $\prec_{\text{lt}}$
cyclic-8	32.480	44.310	100.780	31.120
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

- Predicting zero reductions

- Fast linear algebra for computing Gröbner bases

Faugère's F4 algorithm

Input: Ideal $I = \langle f_1, \ldots, f_m \rangle$
Output: Gröbner basis G for I

1. $G \leftarrow \emptyset$
2. $G \leftarrow G \cup \{f_i\}$ for all $i \in \{1, \ldots, m\}$
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 $I = \langle f_1, \ldots, f_m \rangle$
Output: Gröbner basis G for I

1. $G \leftarrow \emptyset$
2. $G \leftarrow G \cup \{f_i\}$ for all $i \in \{1, \ldots, m\}$
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) $P_d \leftarrow$ **Select** (P), $P \leftarrow P \setminus P_d$
   (c) $L_d \leftarrow \{af, bg \mid (af, bg) \in P_d\}$
   (d) $L_d \leftarrow$ **Symbolic Preprocessing**(L_d, G)
   (e) $F_d \leftarrow$ **Reduction**(L_d, G)
   (f) for $h \in F_d$:
   - If lt$(h) \notin L(G)$ (all other h are "useless"):
     - $P \leftarrow P \cup \{$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 $\mathscr{R}$ by a subset of $\mathscr{R}$

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 $\mathscr{R}$ by a subset of $\mathscr{R}$

If **Select** (P) selects only one 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 $\mathscr{R}$
Output: a finite subset of $\mathscr{R}$

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 $\lambda \in \mathscr{R}$ such that $\lambda\, \mathrm{lt}(g) = m$
       $\triangleright\ F \leftarrow F \cup \{\lambda g\}$
4. Return F

Symbolic preprocessing

Input: L, G finite subsets of $\mathscr{R}$
Output: a finite subset of $\mathscr{R}$

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 $\lambda \in \mathscr{R}$ such that $\lambda\, \mathrm{lt}(g) = m$
   $\triangleright F \leftarrow F \cup \{\lambda g\}$
4. Return F

We optimize this soon!

Reduction

Input: L, G finite subsets of $\mathscr{R}$
Output: a finite subset of $\mathscr{R}$

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 $\mathscr{R}$
Output: a finite subset of $\mathscr{R}$

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	$\hat{=}$	coefficients of polynomials in L

Example: Cyclic-4

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

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

Example: Cyclic-4

$\mathscr{R} = \mathbb{Q}[a,b,c,d]$, $<$ denotes DRL and we use the normal selection strategy for **Select** (P). $I = \langle f_1, \ldots, f_4 \rangle$, 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 $L_1 = \{f_3, bf_4\}$.

Example: Cyclic-4

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

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

Example: Cyclic-4

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

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

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

Example: Cyclic-4

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

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

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

Example: Cyclic-4

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

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

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

Example: Cyclic-4

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

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

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

Example: Cyclic-4

Now reduction:
Convert polynomial data L_1 to Macaulay Matrix M_1

$$\begin{array}{c c}
 & \begin{array}{ccccccc} ab & b^2 & bc & ad & bd & cd & d^2 \end{array} \\
\begin{array}{c} df_4 \\ f_3 \\ bf_4 \end{array} & \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}
\end{array}$$

Example: Cyclic-4

Now reduction:
Convert polynomial data L_1 to Macaulay Matrix M_1

$$\begin{array}{c} \\ df_4 \\ f_3 \\ bf_4 \end{array} \begin{array}{c} \begin{array}{ccccccc} ab & b^2 & bc & ad & bd & cd & d^2 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

Gaussian Elimination of M_1:

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

Example: Cyclic-4

Convert matrix data back to polynomial structure F_1:

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

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

Example: Cyclic-4

Convert matrix data back to polynomial structure F_1:

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

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

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

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

Example: Cyclic-4

Next round:

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

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:

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

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

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

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:

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

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

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

Symbolic preprocessing:

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

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:

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

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

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

Symbolic preprocessing:

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

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

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:

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

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

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

Symbolic preprocessing:

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

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

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:

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

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

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

Symbolic preprocessing:

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

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

Let us investigate this in more detail.

Interlude – Simplify

Idea

Replace $u \cdot f$ by $(wv) \cdot g$ where $vg \in F_i$ for a previous reduction step.
$\Rightarrow$ Reuse rows that are reduced but not "in" G.

Interlude – Simplify

Idea

Replace $u \cdot f$ by $(wv) \cdot g$ where $vg \in F_i$ for a previous reduction step.
$\Rightarrow$ Reuse rows that are reduced but not “in” G.

Note

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

Interlude – Simplify

Idea

Replace $u \cdot f$ by $(wv) \cdot g$ where $vg \in F_i$ for a previous reduction step.
$\Rightarrow$ Reuse rows that are reduced but not "in" G.

Note

- Tries to reuse all rows from old matrices.
  $\Rightarrow$ We need to keep them in memory.
- We also simplify generators of S-pairs, as we have done in our example: $(f_2, bcf_4) \Longrightarrow (f_2, cf_6)$.
- 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:

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

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

Example: Cyclic-4

Symbolic preprocessing - now with simplify:

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

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

Example: Cyclic-4

Symbolic preprocessing - now with simplify:

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

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

Example: Cyclic-4

Symbolic preprocessing - now with simplify:

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

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

Example: Cyclic-4

Symbolic preprocessing - now with simplify:

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

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

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

Knowledge of underlying GB structure

Improve Gaussian Elimination

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

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

Knowledge of underlying GB structure

Improve Gaussian Elimination

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

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

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

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

Faugère-Lachartre Idea

1st step: Sort pivot and non-pivot columns

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

Pivot column

Faugère-Lachartre Idea

1st step: Sort pivot and non-pivot columns

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

Pivot column

Faugère-Lachartre Idea

1st step: Sort pivot and non-pivot columns

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

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

Pivot column

Non-Pivot column

Faugère-Lachartre Idea

1st step: Sort pivot and non-pivot columns

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

Pivot column

Non-Pivot column

Faugère-Lachartre Idea

2nd step: Sort pivot and non-pivot rows

$$\begin{array}{ccc:cccc}
1 & 3 & 7 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 4 & 1 & 0 & 5 \\
0 & 1 & 8 & 6 & 0 & 0 & 9 \\
0 & 5 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 1 & 0 & 0 & 3 & 1
\end{array}$$

Faugère-Lachartre Idea

2nd step: Sort pivot and non-pivot rows

$$
\begin{array}{ccc:cccc}
1 & 3 & 7 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 4 & 1 & 0 & 5 \\
0 & 1 & 8 & 6 & 0 & 0 & 9 \\
0 & 5 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 1 & 0 & 0 & 3 & 1
\end{array}
$$

Pivot row

Faugère-Lachartre Idea

2nd step: Sort pivot and non-pivot rows

$$\left(\begin{array}{ccc:cccc} 1 & 3 & 7 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 4 & 1 & 0 & 5 \\ 0 & 1 & 8 & 6 & 0 & 0 & 9 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 1 \end{array}\right)$$

Pivot row

Non-Pivot row

Faugère-Lachartre Idea

2nd step: Sort pivot and non-pivot rows

$$
\begin{array}{ccc:cccc}
1 & 3 & 7 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 4 & 1 & 0 & 5 \\
0 & 1 & 8 & 6 & 0 & 0 & 9 \\
0 & 5 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 1 & 0 & 0 & 3 & 1
\end{array}
$$

Pivot row

Non-Pivot row

Faugère-Lachartre Idea

2nd step: Sort pivot and non-pivot rows

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

Pivot row

Non-Pivot row

Faugère-Lachartre Idea

3rd step: Reduce lower left part to zero

$$
\left(\begin{array}{ccc:cccc}
1 & 0 & 0 & 4 & 1 & 0 & 5 \\
0 & 5 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 1 & 0 & 0 & 3 & 1 \\
\hdashline
1 & 3 & 7 & 0 & 0 & 1 & 0 \\
0 & 1 & 8 & 6 & 0 & 0 & 9
\end{array}\right)
$$

Faugère-Lachartre Idea

3rd step: Reduce lower left part to zero

$$\begin{array}{ccc|cccc} 1 & 0 & 0 & 4 & 1 & 0 & 5 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 1 \\ \hline 1 & 3 & 7 & 0 & 0 & 1 & 0 \\ 0 & 1 & 8 & 6 & 0 & 0 & 9 \end{array} \longrightarrow \begin{array}{ccc|cccc} 1 & 0 & 0 & 4 & 1 & 0 & 5 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 1 \\ \hline 0 & 0 & 0 & 7 & 10 & 3 & 10 \\ 0 & 0 & 0 & 6 & 0 & 2 & 1 \end{array}$$

Faugère-Lachartre Idea

4th step: Reduce lower right part

$$
\left(\begin{array}{ccc:cccc}
1 & 0 & 0 & 4 & 1 & 0 & 5 \\
0 & 5 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 1 & 0 & 0 & 3 & 1 \\
\hdashline
0 & 0 & 0 & 7 & 10 & 3 & 10 \\
0 & 0 & 0 & 6 & 0 & 2 & 1
\end{array}\right)
$$

4th step: Reduce lower right part

$$\left(\begin{array}{ccc:cccc} 1 & 0 & 0 & 4 & 1 & 0 & 5 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 1 \\ \hdashline 0 & 0 & 0 & 7 & 10 & 3 & 10 \\ 0 & 0 & 0 & 6 & 0 & 2 & 1 \end{array}\right) \longrightarrow \left(\begin{array}{ccc:cccc} 1 & 0 & 0 & 4 & 1 & 0 & 5 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 1 \\ \hdashline 0 & 0 & 0 & 7 & 10 & 3 & 10 \\ 0 & 0 & 0 & 0 & 4 & 1 & 5 \end{array}\right)$$

Faugère-Lachartre Idea

4th step: Reduce lower right part

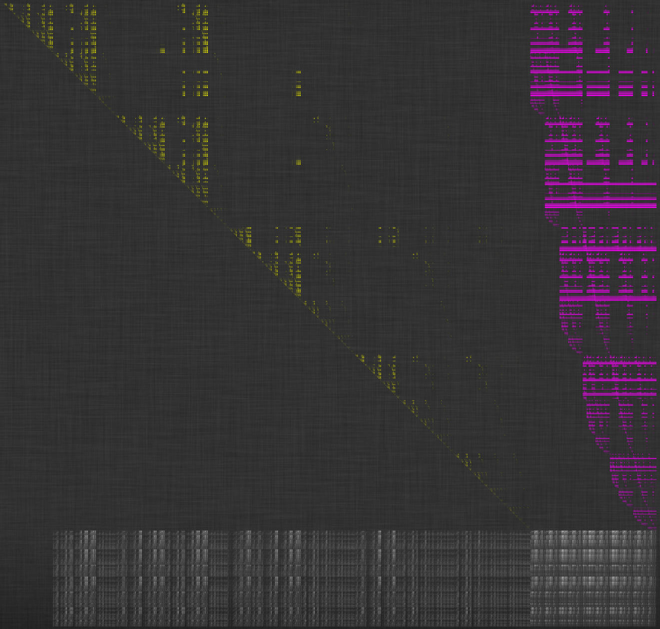
$$\left(\begin{array}{ccc:cccc} 1 & 0 & 0 & 4 & 1 & 0 & 5 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 1 \\ \hdashline 0 & 0 & 0 & 7 & 10 & 3 & 10 \\ 0 & 0 & 0 & 6 & 0 & 2 & 1 \end{array}\right) \longrightarrow \left(\begin{array}{ccc:cccc} 1 & 0 & 0 & 4 & 1 & 0 & 5 \\ 0 & 5 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 1 \\ \hdashline 0 & 0 & 0 & 7 & 10 & 3 & 10 \\ 0 & 0 & 0 & 0 & 4 & 1 & 5 \end{array}\right)$$

5th step: Remap columns of lower right part

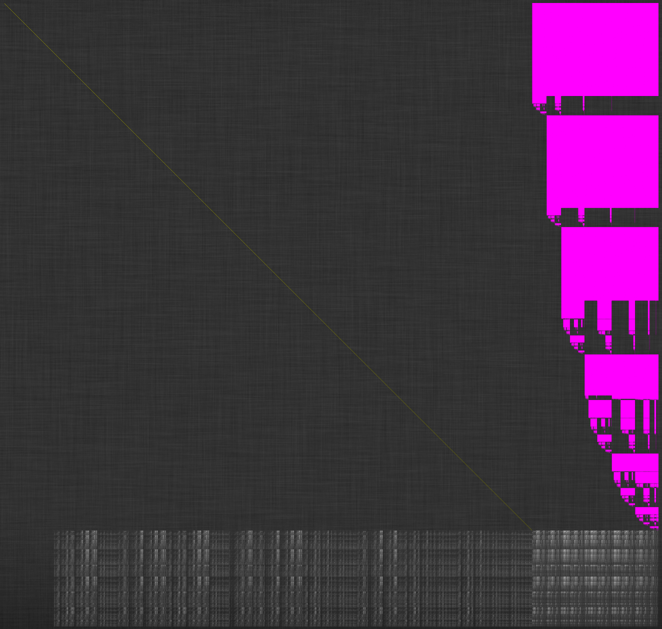
How our matrices look like (1)

How our matrices look like (2)

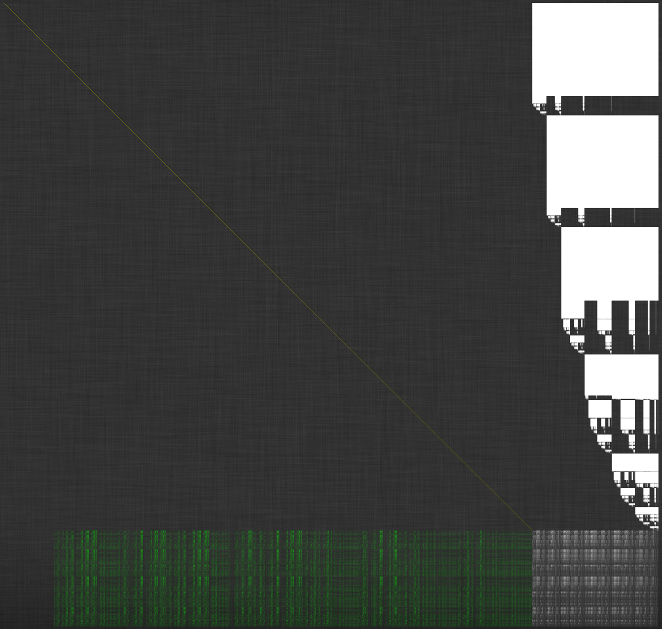
Hybrid Matrix Multiplication $A^{-1}B$



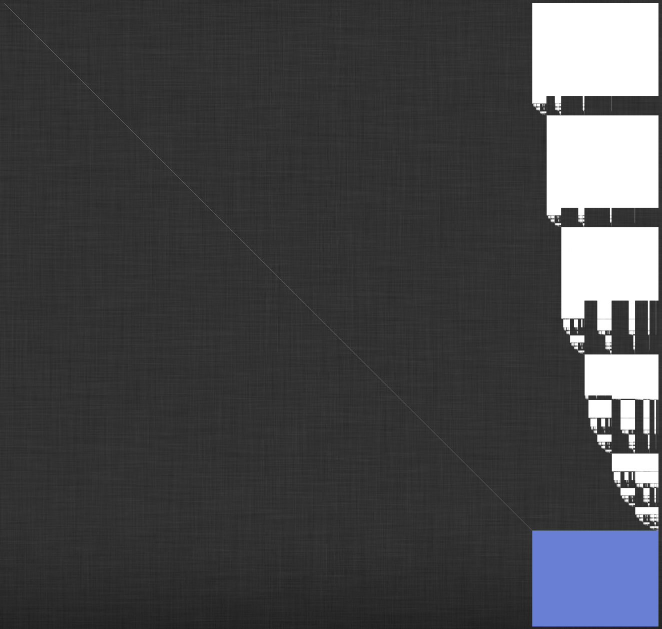
Hybrid Matrix Multiplication $A^{-1}B$



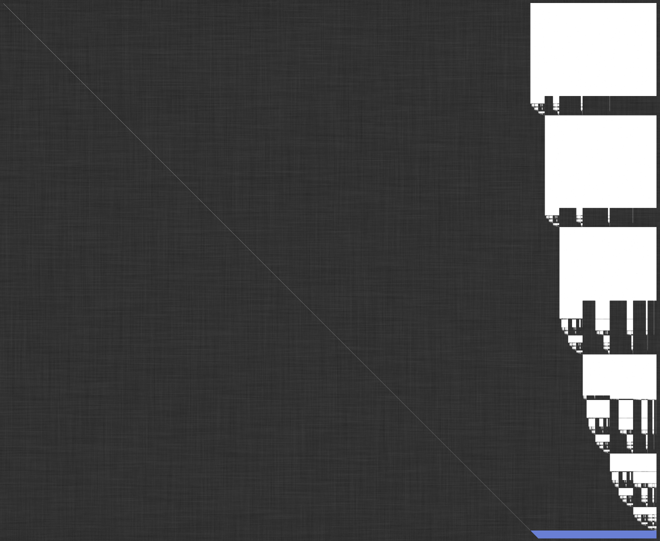
Reduce C to zero



Gaussian Elimination on D



New information



Experimental results – vs. Faugère-Lachartre

Implementation			FGL reduction			GBLA		
Matrix/Threads:			1	16	32	1	16	32
F_5	kat13	mat5	16.7	2.7	2.3	14.5	2.02	1.87
		mat6	27.3	4.15	4.0	23.9	3.08	2.65
F_5	kat14	mat7	139	17.4	16.6	142	13.4	10.6
		mat8	181	24.95	23.1	177	16.9	12.7
F_5	kat15	mat7	629	61.8	55.6	633	55.1	38.2
F_5	kat16	mat6	1,203	110	83.3	1,147	98.7	69.9
F_5	mr-9-10-7	mat3	591	70.8	71.3	733	57.3	37.9

Experimental results – vs. Magma V2.20-10

Implementation			Magma	GBLA		
Matrix/Threads:			1	1	16	32
F_4	kat12	mat9	11.2	11.4	1.46	1.60
F_4	kat13	mat2	0.94	1.18	0.38	0.61
		mat3	9.33	11.0	1.70	3.10
		mat9	168	165	16.0	11.8
F_4	kat14	mat8	2747	2545	207	165
F_4	kat15	mat7	10,345	9,514	742	537
		mat8	13,936	12,547	961	604
		mat9	24,393	22,247	1,709	1,256

References

[1] Buchberger, B. Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal, 1965. PhD thesis, Universtiy of Innsbruck, Austria

[2] Buchberger, B. A criterion for detecting unnecessary reductions in the construction of Gröbner bases, 1979. EUROSAM '79, An International Symposium on Symbolic and Algebraic Manipulation

[3] Buchberger, B. Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory, 1985. Multidimensional Systems Theory, D. Reidel Publication Company

[4] Eder, C. and Faugère, J.-C. A survey on signature-based Groebner basis algorithms, 2014. http://arxiv.org/abs/1404.1774

[5] Faugère, J.-C. A new efficient algorithm for computing Gröbner bases (F4), 1999. Journal of Pure and Applied Algebra

[6] Faugère, J.-C. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5), 2002. Proceedings of the 2002 international symposium on Symbolic and algebraic computation

[7] Faugère, J.-C. and Lachartre, S. Parallel Gaussian Elimination for Gröbner bases computations in finite fields, 2010. Proceedings of the 4th International Workshop on Parallel and Symbolic Computation

[8] Gebauer, R. and Möller, H. M. On an installation of Buchberger's algorithm, 1988. Journal of Symbolic Computation