

Exploiting algebraic structures to solve polynomial systems of equations

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

Séminaire de théorie des codes et cryptographie à Neuchâtel (et à Zürich)

May 18, 2015



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

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

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 $M_0, \dots, M_k \in \mathcal{M}_{n \times n}(\mathcal{K})$, find (if possible) $(\lambda_1, \dots, \lambda_k) \in \mathcal{K}^k$ such that

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

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 $M_0, \dots, M_k \in \mathcal{M}_{n \times n}(\mathcal{K})$, find (if possible) $(\lambda_1, \dots, \lambda_k) \in \mathcal{K}^k$ such that

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

Solving polynomial equations is important
Gröbner Bases are cool!

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 [1]

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 $\mathbb{Q} \rightarrow$ several \mathbb{Z}_{p_i} computations and CRT
- ▶ Predict zero reductions fast checks \rightarrow 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 → 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

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

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 [2]

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 [2]

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 [2]

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 [2]

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 [2]

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 $\text{spol}(g_3, g_2)$:

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

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

Buchberger's criteria

Chain criterion [3]

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 [3]

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 [3]

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!

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

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}

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

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	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 $I = \langle f_1, \dots, f_m \rangle$

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 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 \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 $\lambda \text{It}(g) = m$
 ▷ $F \leftarrow F \cup \{\lambda g\}$
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 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 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 $\hat{=}$ coefficients 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**:

$$T(L_1) = \{\textcolor{blue}{ab}, b^2, bc, ad, bd, cd\}$$

$$L_1 = \{f_3, bf_4\}$$

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**:

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

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

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

$$f_1 = abcd - 1,$$

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

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

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

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**:

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

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

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

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

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**:

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

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

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

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

Gaussian Elimination of M_1 :

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

Example: Cyclic-4

Convert matrix data back to polynomial structure F_1 :

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

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

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:

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

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

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:

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

$bc^2 \notin L(G)$, $abd = \text{lt}(bdf_4)$, 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:

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

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

Let us investigate this in more detail.

Interlude – Simplify

Idea

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

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.

Interlude – Simplify

Idea

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

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**:

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

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

Example: Cyclic-4

Symbolic preprocessing - now with **simplify**:

$$\begin{aligned} T(L_2) &= \{ \mathbf{abc}, \mathbf{bc}^2, \mathbf{abd}, \mathbf{acd}, \mathbf{bcd}, \mathbf{cd}^2, \mathbf{b}^2\mathbf{d}, \mathbf{c}^2\mathbf{d} \quad \} \\ L_2 &= \{ f_2, \mathbf{cf}_6, \mathbf{bf}_5 \quad \} \end{aligned}$$

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

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.

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

Knowledge of underlying GB structure

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

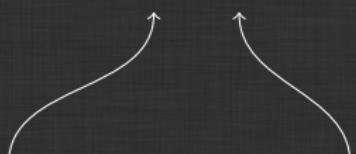


Faugère-Lachartre Idea

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

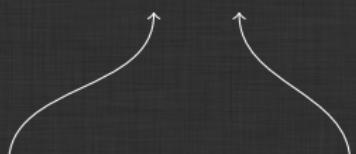


Faugère-Lachartre Idea

1st step: Sort pivot and non-pivot columns

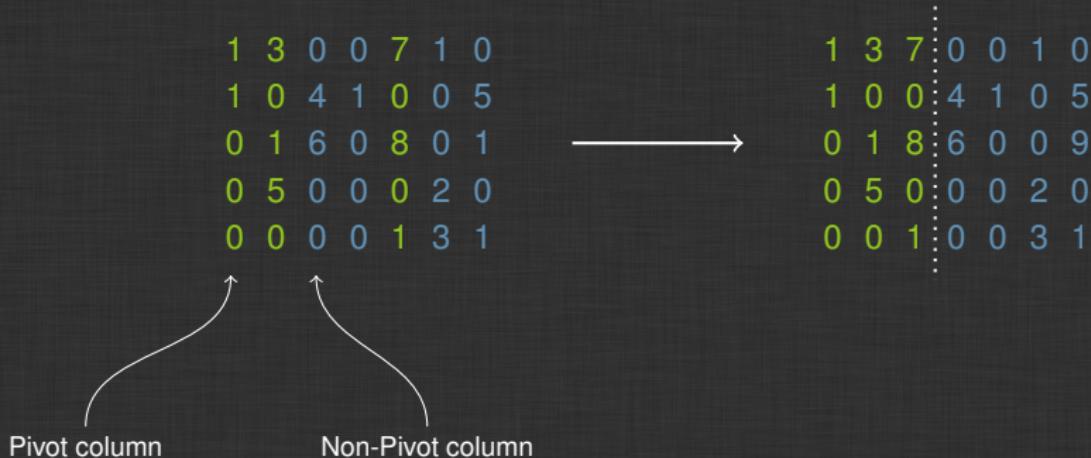
1	3	0	0	7	1	0
1	0	4	1	0	0	5
0	1	6	0	8	0	1
0	5	0	0	0	2	0
0	0	0	0	1	3	1

Pivot column Non-Pivot column



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

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

Faugère-Lachartre Idea

4th step: Reduce lower right part

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

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

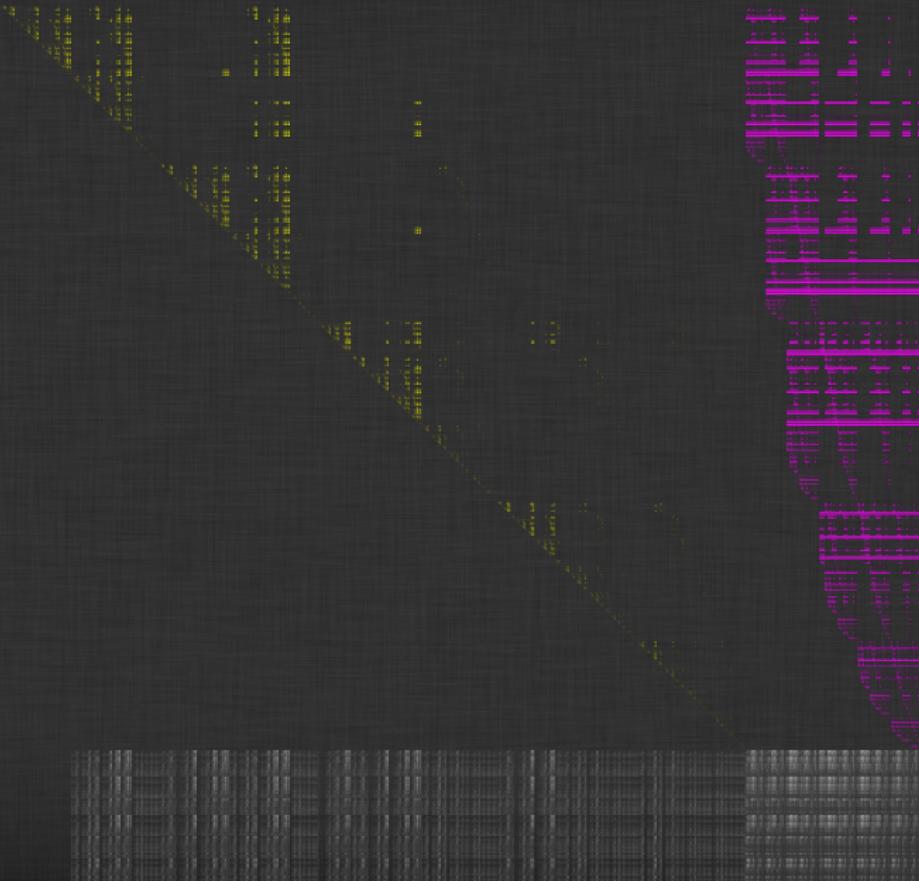
5th step: Remap columns of lower right part

How our matrices look like (1)

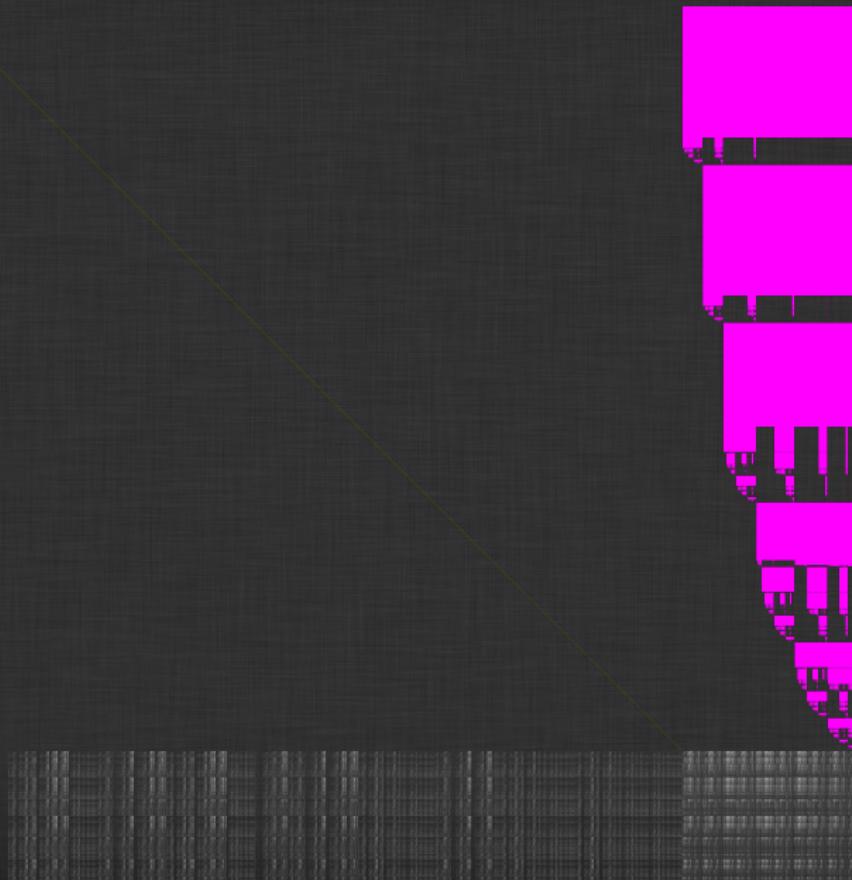
How our matrices look like (2)

row	col	value
0	0	1
0	1	0
1	0	0
1	1	1
2	0	0
2	1	0
3	0	0
3	1	1
4	0	0
4	1	0
5	0	0
5	1	1
6	0	0
6	1	0
7	0	0
7	1	1
8	0	0
8	1	0
9	0	0
9	1	1
10	0	0
10	1	0
11	0	0
11	1	1
12	0	0
12	1	0
13	0	0
13	1	1
14	0	0
14	1	0
15	0	0
15	1	1
16	0	0
16	1	0
17	0	0
17	1	1
18	0	0
18	1	0
19	0	0
19	1	1
20	0	0
20	1	0
21	0	0
21	1	1
22	0	0
22	1	0
23	0	0
23	1	1
24	0	0
24	1	0
25	0	0
25	1	1
26	0	0
26	1	0
27	0	0
27	1	1
28	0	0
28	1	0
29	0	0
29	1	1
30	0	0
30	1	0
31	0	0
31	1	1
32	0	0
32	1	0
33	0	0
33	1	1
34	0	0
34	1	0
35	0	0
35	1	1
36	0	0
36	1	0
37	0	0
37	1	1
38	0	0
38	1	0
39	0	0
39	1	1
40	0	0
40	1	0
41	0	0
41	1	1
42	0	0
42	1	0
43	0	0
43	1	1
44	0	0
44	1	0
45	0	0
45	1	1
46	0	0
46	1	0
47	0	0
47	1	1
48	0	0
48	1	0
49	0	0
49	1	1
50	0	0
50	1	0
51	0	0
51	1	1
52	0	0
52	1	0
53	0	0
53	1	1
54	0	0
54	1	0
55	0	0
55	1	1
56	0	0
56	1	0
57	0	0
57	1	1
58	0	0
58	1	0
59	0	0
59	1	1
60	0	0
60	1	0
61	0	0
61	1	1
62	0	0
62	1	0
63	0	0
63	1	1
64	0	0
64	1	0
65	0	0
65	1	1
66	0	0
66	1	0
67	0	0
67	1	1
68	0	0
68	1	0
69	0	0
69	1	1
70	0	0
70	1	0
71	0	0
71	1	1
72	0	0
72	1	0
73	0	0
73	1	1
74	0	0
74	1	0
75	0	0
75	1	1
76	0	0
76	1	0
77	0	0
77	1	1
78	0	0
78	1	0
79	0	0
79	1	1
80	0	0
80	1	0
81	0	0
81	1	1
82	0	0
82	1	0
83	0	0
83	1	1
84	0	0
84	1	0
85	0	0
85	1	1
86	0	0
86	1	0
87	0	0
87	1	1
88	0	0
88	1	0
89	0	0
89	1	1
90	0	0
90	1	0
91	0	0
91	1	1
92	0	0
92	1	0
93	0	0
93	1	1
94	0	0
94	1	0
95	0	0
95	1	1
96	0	0
96	1	0
97	0	0
97	1	1
98	0	0
98	1	0
99	0	0
99	1	1
100	0	0
100	1	0
101	0	0
101	1	1
102	0	0
102	1	0
103	0	0
103	1	1
104	0	0
104	1	0
105	0	0
105	1	1
106	0	0
106	1	0
107	0	0
107	1	1
108	0	0
108	1	0
109	0	0
109	1	1
110	0	0
110	1	0
111	0	0
111	1	1
112	0	0
112	1	0
113	0	0
113	1	1
114	0	0
114	1	0
115	0	0
115	1	1
116	0	0
116	1	0
117	0	0
117	1	1
118	0	0
118	1	0
119	0	0
119	1	1
120	0	0
120	1	0
121	0	0
121	1	1
122	0	0
122	1	0
123	0	0
123	1	1
124	0	0
124	1	0
125	0	0
125	1	1
126	0	0
126	1	0
127	0	0
127	1	1
128	0	0
128	1	0
129	0	0
129	1	1
130	0	0
130	1	0
131	0	0
131	1	1
132	0	0
132	1	0
133	0	0
133	1	1
134	0	0
134	1	0
135	0	0
135	1	1
136	0	0
136	1	0
137	0	0
137	1	1
138	0	0
138	1	0
139	0	0
139	1	1
140	0	0
140	1	0
141	0	0
141	1	1
142	0	0
142	1	0
143	0	0
143	1	1
144	0	0
144	1	0
145	0	0
145	1	1
146	0	0
146	1	0
147	0	0
147	1	1
148	0	0
148	1	0
149	0	0
149	1	1
150	0	0
150	1	0
151	0	0
151	1	1
152	0	0
152	1	0
153	0	0
153	1	1
154	0	0
154	1	0
155	0	0
155	1	1
156	0	0
156	1	0
157	0	0
157	1	1
158	0	0
158	1	0
159	0	0
159	1	1
160	0	0
160	1	0
161	0	0
161	1	1
162	0	0
162	1	0
163	0	0
163	1	1
164	0	0
164	1	0
165	0	0
165	1	1
166	0	0
166	1	0
167	0	0
167	1	1
168	0	0
168	1	0
169	0	0
169	1	1
170	0	0
170	1	0
171	0	0
171	1	1
172	0	0
172	1	0
173	0	0
173	1	1
174	0	0
174	1	0
175	0	0
175	1	1
176	0	0
176	1	0
177	0	0
177	1	1
178	0	0
178	1	0
179	0	0
179	1	1
180	0	0
180	1	0
181	0	0
181	1	1
182	0	0
182	1	0
183	0	0
183	1	1
184	0	0
184	1	0
185	0	0
185	1	1
186	0	0
186	1	0
187	0	0
187	1	1
188	0	0
188	1	0
189	0	0
189	1	1
190	0	0
190	1	0
191	0	0
191	1	1
192	0	0
192	1	0
193	0	0
193	1	1
194	0	0
194	1	0
195	0	0
195	1	1
196	0	0
196	1	0
197	0	0
197	1	1
198	0	0
198	1	0
199	0	0
199	1	1
200	0	0
200	1	0
201	0	0
201	1	1
202	0	0
202	1	0
203	0	0
203	1	1
204	0	0
204	1	0
205	0	0
205	1	1
206	0	0
206	1	0
207	0	0
207	1	1
208	0	0
208	1	0
209	0	0
209	1	1
210	0	0
210	1	0
211	0	0
211	1	1
212	0	0
212	1	0
213	0	0
213	1	1
214	0	0
214	1	0
215	0	0
215	1	1
216	0	0
216	1	0
217	0	0
217	1	1
218	0	0
218	1	0
219	0	0
219	1	1
220	0	0
220	1	0
221	0	0
221	1	1
222	0	0
222	1	0
223	0	0
223	1	1
224	0	0
224	1	0
225	0	0
225	1	1
226	0	0
226	1	0
227	0	0
227	1	1
228	0	0
228	1	0
229	0	0
229	1	1
230	0	0
230	1	0
231	0	0
231	1	1
232	0	0
232	1	0
233	0	0
233	1	1
234	0	0
234	1	0
235	0	0
235	1	1
236	0	0
236	1	0
237	0	0
237	1	1
238	0	0
238	1	0
239	0	0
239	1	1
240	0	0
240	1	0
241	0	0
241	1	1
242	0	0
242	1	0
243	0	0
243	1	1
244	0	0
244	1	0
245	0	0
245	1	1
246	0	0
246	1	0
247	0	0
247	1	1
248	0	0
248	1	0
249	0	0
249	1	1
250	0	0
250	1	0
251	0	0
251	1	1
252	0	0
252	1	0
253	0	0
253	1	1
254	0	0
254	1	0
255	0	0
255	1	1
256		

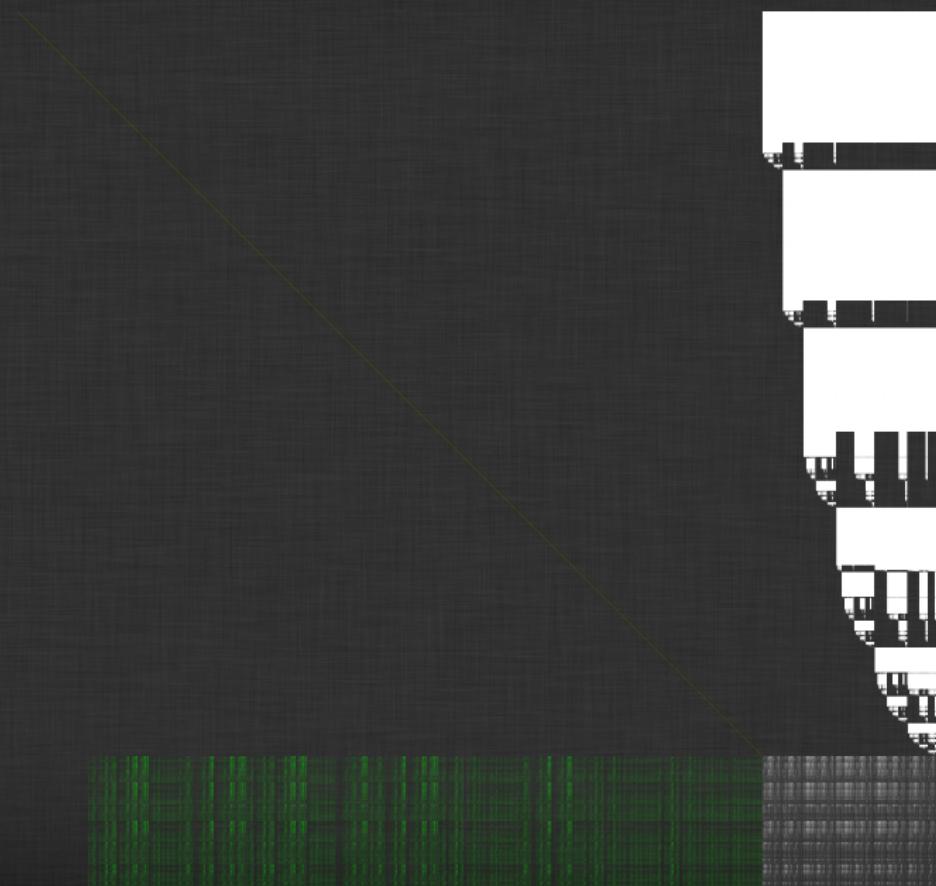
Hybrid Matrix Multiplication $A^{-1}B$



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



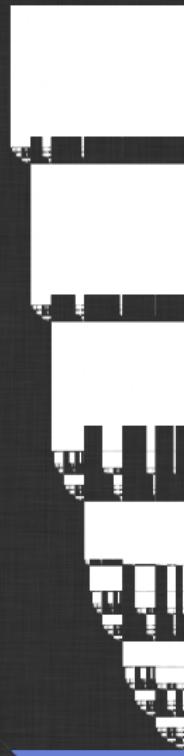
Reduce C to zero



Gaussian Elimination on D



New information



- ▶ New open-source, plain C library, specialized linear algebra for GB computations
- ▶ at the moment: dedicated to finite fields, $p \leq 65521 < 2^{32}$
- ▶ written together with Brice Boyer and Jean-Charles Faugère
- ▶ several strategies for splicing and reduction steps
- ▶ includes converter for our dedicated matrix format, e.g. from/to Magma
- ▶ comes with a huge matrix database, > 280 GB of data

- ▶ New open-source, plain C library, specialized linear algebra for GB computations
- ▶ at the moment: dedicated to finite fields, $p \leq 65521 < 2^{32}$
- ▶ written together with Brice Boyer and Jean-Charles Faugère
- ▶ several strategies for splicing and reduction steps
- ▶ includes converter for our dedicated matrix format, e.g. from/to Magma
- ▶ comes with a huge matrix database, > 280 GB of data

<http://hpac.imag.fr/gbla/>

Repository will soon be open for external contributions!

GBLA vs. Faugère-Lachartre

Implementation			FL 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

GBLA vs. Magma V2.20-10

Implementation	Magma	GBLA		
		1	16	32
Matrix/Threads:				
F_4	kat12 mat9	11.2	11.4	1.46
F_4	kat13 mat2	0.94	1.18	0.38
	mat3	9.33	11.0	1.70
	mat9	168	165	11.8
F_4	kat14 mat8	2747	2545	207
F_4	kat15 mat7	10,345	9,514	742
	mat8	13,936	12,547	961
	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, Universitiy 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*