On

Signature-based Gröbner Bases

over

Euclidean Rings

The following is joint work by

Christian Eder, Gerhard Pfister, Adi Popescu.

We are from the

University of Kaiserslautern.

Idea of signatures:

Try to detect zero reductions in advance.

Let $\mathrm{I} = \langle g_1, g_2 \rangle \in \mathcal{R} := \mathbb{Q}[x, y, z]$ and let $<$ denote DRL where

$$g_1 = xy - z^2,$$
$$g_2 = y^2 - z^2.$$

Apply signatures in $\mathcal{R}^2$:

$\mathbf{sig}(g_1) = \mathbf{e}_1,$
$\mathbf{sig}(g_2) = \mathbf{e}_2.$

Order signatures by POT (e.g. $\mathbf{e}_2 > x^{1000}\mathbf{e}_1$).

In general:
$\mathbf{sig}$(polynomial) $=$ $\mathbf{lt}$(module representation)

Main idea: Try to **keep signatures minimal**.

Generate first S-pair:

$$
\begin{aligned}
g_3 &:= sp(g_1, g_2) = yg_1 - xg_2 \\
&= y\left(xy - z^2\right) - x\left(y^2 - z^2\right) \\
&= xz^2 - yz^2.
\end{aligned}
$$

$$
\operatorname{sig}(g_3) = \operatorname{lt}(ye_1 - xe_2) = -xe_2.
$$

$\mathrm{sp}(g_3, g_2)$ reduces to zero:

$\mathrm{lt}(g_2) = y^2$ coprime to $\mathrm{lt}(g_3) = xz^2$

(Buchberger’s Product Criterion)

$\mathbf{sp}(\mathbf{g}_3, \mathbf{g}_2)$ reduces to zero (**signature edition**):

$$\begin{aligned} \mathbf{sig}(\mathbf{sp}(\mathbf{g}_3, \mathbf{g}_2)) &= \mathbf{lt}(y^2(ye_1 - xe_2) - xz^2e_2) \\ &= -xy^2e_2. \end{aligned}$$

Use syzygy $\mathbf{g}_1\mathbf{e}_2 - \mathbf{g}_2\mathbf{e}_1$ with lead term $\mathbf{xye}_2$

▷ Reduce module representation

▷ Lower signature for $\mathbf{sp}(\mathbf{g}_3, \mathbf{g}_2)$

(**Syzygy Criterion**)

What's about $\mathsf{sp}(g_3, g_1)$?

Buchberger's Product Criterion? **NO**

Buchberger's Chain Criterion? **NO**

But $\mathbf{sp}(g_3, g_1)$ reduces to zero:

$$\begin{aligned} \mathrm{sig}(\mathrm{sp}(g_3, g_1)) &= \mathrm{lt}(y(ye_1 - xe_2) - z^2 e_1) \\ &= -xye_2. \end{aligned}$$

Again: Syzygy $\mathbf{g_1 e_2 - g_2 e_1}$ with lead term $\mathbf{xye_2}$

▷ Reduce module representation

▷ Lower signature for $\mathbf{sp}(g_3, g_1)$

(**Syzygy Criterion**)

Precondition for this talk:
Next chosen S-pair from pair set has minimal possible signature.

Note: Over fields this limitation is not required, but makes life easier.

General rule

For each signature handle exactly **one** element.

Sketch of proof

Take pairs by increasing signature.
Think in the module: α, β module elements.

If $\mathbf{sig}(\alpha) = \mathbf{sig}(\beta)$ reduce in module.
(We are not cancelling the polynomial lts!)

▷ $\mathbf{sig}(\alpha - \beta) < \mathbf{sig}(\alpha), \mathbf{sig}(\beta)$.

▷ Algorithm has handled these relations already.

Let’s compute with signatures over **Euclidean rings**.

Problem not mentioned until now: **sig-reductions**

When reducing polynomials (**over fields**) we are not allowed to change the signature:

Increasing signature?
Well, we want small signatures.

Decreasing signature?
We can throw away the element, criteria apply.

Over fields $\Rightarrow$ **computation by increasing signature**.

Over Euclidean rings stuff gets more difficult.

We want **strong Gröbner Bases**:

For all $f \in I \setminus \{0\}$ there exists $g \in G$ s.t. $\mathrm{lt}(g) \mid \mathrm{lt}(f)$.

($G \subset I$ and $L(G) = L(I)$ is not enough anymore.)

Have to take care of the **coefficients**, too:

If

- ▷ $\mathrm{lm}(g) \mid \mathrm{lm}(f)$ &
- ▷ $\exists\, a, b$ coefficients s.t.
  $\mathrm{lc}(f) = a\,\mathrm{lc}(g) + b$, $a \neq 0$ and $b < \mathrm{lc}(f)$

---

then compute $f - a\,\frac{\mathrm{lm}(f)}{\mathrm{lm}(g)}\,g$.

(Either **smaller lm** or **smaller lc**!)

Same process generalizes concept of S-pairs:

We need to consider **GCD-pairs**.

For **efficiency** we need to **relax signature handling**:

Allow signature changes on the **coefficient** level.

Two main **problems** arise from this.

#1

Over Euclidean rings we **can no longer guarantee** that the computation of signature-based Gröbner Bases is done by **increasing signatures**.

#2

We need to **restrict signature-based criteria** to remove useless elements:

We can only remove elements for a given signature S if there exists a syzygy π s.t. $\mathrm{lt}(\pi) \mid S$

Still, **signature drops** may appear.

Idea

- Stop computation at this point.
- Interreduce intermediate basis without considering signatures.
- Apply new signatures / module representations and restart.

Optimization 1: Exploit GCD-pairs

Replace $f \in G$ by $gp(f, g)$ if there exists $g \in G$ s.t.

$$\begin{aligned} gp(f, g) &= (\pm 1)f + ctg \quad \& \\ sig(gp(f, g)) &= (\pm 1)sig(f) . \end{aligned}$$

▷ Trying to keep coefficient growth at a low.

Optimization 2: Optimistic sig-reductions

- sig-reduce an element f w.r.t. G.
- If there exists $g \in G$ s.t. $ct\, lt(g) = lt(f)$ and $sig(f - ctg) < sig(f)$ for some c and t then start **usual reduction process** (no longer taking care of signatures)
- If f reduces to zero we can go on.
- Otherwise we have to restart the computation.

Restarting is a huge bottleneck in general.

But often the intermediate computed elements are quite useful for further computations.

Optimization 3: Hybrid algorithm

- Start with signature-based algorithm.
- If the signature drops, restart for a (small) number of times the signature-based algorithm.
- Take intermediate basis and start **non-signature-based** Gröbner basis computation.

Examples	STD	HBA	STD/HBA
1	10.43	0.37	28.19
2	24.91	0.10	249.10
3	87.27	0.39	223.77
4	83.51	0.20	417.55
5	23,200.05	5,873.21	3.95
6	134.29	0.61	220.15
7	1,004.56	1,128.07	0.89
8	554.02	337.55	1.641

Up next

Throw some machine learning on it.

And what’s about finite rings?

Thank you for your attention.

Questions? Remarks?