SIGNATURE-BASED GRÖBNER BASIS ALGORITHMS IN SINGULAR

CHRISTIAN EDER

ABSTRACT. In 2001 Faugère published his **F5** algorithm for computing Gröbner bases. This algorithm is known to compute no zero reductions for regular input sequences and it has broken several previously untractable problems, for example in cryptanalysis. Using signatures **F5** can detect useless computations in advance. Over the years many variants of this algorithm have been published with slight optimizations. Version $4-0-0$ of SINGULAR provides the first open-source implementation of signature-based Gröbner basis algorithms including all variants known.

1. INTRODUCTION

Since 1965 [2] Gröbner bases are practically feasible. One of the main algorithmic improvements is the prediction of useless data during the computations. That means to use criteria to detect zero reductions in advance [3, 4]. Gebauer and Möller gave an optimal implementation of Buchberger's criteria [12], but showed that not all zero reductions are discarded. In 2002 Faugère presented the **F5** algorithm [9] which uses new criteria based on so-called "signatures". For regular input sequences **F5** does not compute any zero reduction at all. Thus a new class of so-called "signature-based" Gröbner basis algorithms emerged, see, for example, [1,6,7,11,13].

We present a basic version we call **SBA** with which we explain the ideas behind this kind of algorithms easily. Moreover, we present experimental results for an efficient implementation in the computer algebra system SINGULAR [5] that is open-source and includes all known variants.

2. NOTATIONS

Let $\mathscr{R}$ be a polynomial ring over a field $\mathscr{K}$. All polynomials $f \in \mathscr{R}$ can be uniquely written as a finite sum $f = \sum_{\kappa_\nu x^\nu \in \mathscr{M}} \kappa_\nu x^\nu$ where $\kappa_\nu \in \mathscr{K}$, $x^\nu := \prod_i x_i^{\nu_i}$ and $\mathscr{M}$ is minimal. The elements of $\mathscr{M}$ are the *terms* of f. A *monomial* is a polynomial with exactly one term. A monomial with a coefficient of 1 is *monic*. $f \simeq g$ for $f, g \in \mathscr{R}$ if there exists a non-zero $\kappa \in \mathscr{K}$ such that $f = \kappa g$.

Let $\mathscr{R}^m$ be a free $\mathscr{R}$-module and let $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_m$ be the canonical basis of unit vectors in $\mathscr{R}^m$. $\alpha \in \mathscr{R}^m$ can be uniquely written as a finite sum $\alpha = \sum_{a\boldsymbol{e}_i \in \mathscr{N}} a\boldsymbol{e}_i$ where the a are monomials and $\mathscr{N}$ is minimal. The elements of $\mathscr{N}$ are the *terms* of α. A *module monomial* is an element of $\mathscr{R}^m$ with exactly one term. A module monomial with a coefficient of 1 is *monic*. Let $\alpha \simeq \beta$ for $\alpha, \beta \in \mathscr{R}^m$ if $\alpha = \kappa\beta$ for some non-zero $\kappa \in \mathscr{K}$.

Consider a finite sequence of polynomials $f_1, \ldots, f_m \in \mathscr{R}$. We call $f_1, \ldots, f_m$ a regular sequence if f_i is a non-zero-divisor on $\mathscr{R}/\langle f_1, \ldots, f_{i-1}\rangle$ for $i = 2, \ldots, m$. We define the homomorphism $\alpha \mapsto \overline{\alpha}$ from $\mathscr{R}^m$ to $\mathscr{R}$ by $\overline{\alpha} := \sum_{i=1}^m \alpha_i f_i$. An element $\alpha \in \mathscr{R}^m$ with $\overline{\alpha} = 0$ is called a *syzygy*. The module of all syzygies of $f_1, \ldots, f_m$ is denoted by $\operatorname{syz}(f_1, \ldots, f_m)$.

Let $\leq$ denote two different orders – one for $\mathscr{R}$ and one for $\mathscr{R}^m$: The order for $\mathscr{R}$ is a monomial order, which means that it is a well-order on the set of monomials in $\mathscr{R}$ such that $a \leq b$ implies $ca \leq cb$ for all monomials $a, b, c \in \mathscr{R}$. The order for $\mathscr{R}^m$ is a module monomial order which means that it is a well-order on the set of module monomials in $\mathscr{R}^m$ such that $S \leq T$ implies

$cS \leq cT$ for all module monomials $S, T \in \mathscr{R}^m$ and monomials $c \in \mathscr{R}$. We require the two orders to be *compatible* in the sense that $a \leq b$ if and only if $a\boldsymbol{e}_i \leq b\boldsymbol{e}_i$ for all monomials $a, b \in \mathscr{R}$ and $i = 1, \ldots, m$. The following compatible orders are commonly used in signature-based Gröbner basis algorithms.

Definition 1. *Let $<$ be a monomial order on $\mathscr{R}$ and let ae_i, be_j be two module monomials in $\mathscr{R}^m$. $ae_i <_{pot} be_j$ iff $i < j$ or $i = j$ and $a < b$. This can be combined by either a weighted degree or a weighted leading monomial:*

(a) $ae_i <_{d\text{-}pot} be_j$ *iff* $\deg(\overline{ae_i}) < \deg\left(\overline{be_j}\right)$ *or* $\deg(\overline{ae_i}) = \deg\left(\overline{be_j}\right)$ *and* $a\boldsymbol{e}_i <_{pot} b\boldsymbol{e}_j$.

(b) $ae_i <_{lt\text{-}pot} be_j$ *iff* $\mathrm{lt}(\overline{ae_i}) < \mathrm{lt}\left(\overline{be_j}\right)$ *or* $\mathrm{lt}(\overline{ae_i}) = \mathrm{lt}\left(\overline{be_j}\right)$ *and* $a\boldsymbol{e}_i <_{pot} b\boldsymbol{e}_j$.

Definition 2.

(a) The lead term *resp.* signature $\mathfrak{s}(\alpha)$ *of* $\alpha \in \mathscr{R}^m \setminus \{0\}$ *denotes the* $\leq$*-maximal term of* α. *If* $a\boldsymbol{e}_i = \mathfrak{s}(\alpha)$ *then we call* $\mathrm{ind}(\alpha) := i$ *the* index *of* α. *For* $\overline{\alpha} \in \mathscr{R} \setminus \{0\}$ *the* lead term $\mathrm{lt}(\overline{\alpha})$ *is the* $\leq$*-maximal term of* f. *The* lead coefficient $\mathrm{lc}(\overline{\alpha})$ *is the coefficient of* $\mathrm{lt}(\overline{\alpha})$.

(b) For $\alpha \in \mathscr{R}^m$ *the* sig-poly pair *of* α *is* $(\mathfrak{s}(\alpha), \overline{\alpha}) \in \mathscr{R}^m \times \mathscr{R}$. $\alpha, \beta \in \mathscr{R}^m$ *are* equal up to sig-poly pairs *if* $\mathfrak{s}(\alpha) = \mathfrak{s}(\kappa\beta)$ *and* $\overline{\alpha} = \overline{\kappa\beta}$ *for some non-zero* $\kappa \in \mathscr{K}$. *Correspondingly,* α, β *are said to be* equal up to sig-lead pairs *if* $\mathfrak{s}(\alpha) = \mathfrak{s}(\kappa\beta)$ *and* $\mathrm{lt}(\overline{\alpha}) = \mathrm{lt}\left(\overline{\kappa\beta}\right)$ *for some non-zero* $\kappa \in \mathscr{K}$.

3. Signature Gröbner bases

Every non-syzygy module element $\alpha \in \mathscr{R}^m$ has two main associated characteristics – the signature $\mathfrak{s}(\alpha) \in \mathscr{R}^m$ and the polynomial lead term $\mathrm{lt}(\overline{\alpha}) \in \mathscr{R}$. Lead terms and signatures include a coefficient for mathematical convenience, an implementation of a signature-based Gröbner basis algorithm over fields need not store the signature coefficients. To keep track of the signatures we get a classic polynomial reduction together with a further condition.

Definition 3. *Let* $\alpha \in \mathscr{R}^m$ *and let* t *be a term of* $\overline{\alpha}$. *Then we can* $\mathfrak{s}$-reduce t *by* $\beta \in \mathscr{R}^m$ *if there exists a monomial* b *such that* $\mathrm{lt}\left(\overline{b\beta}\right) = t$ *and* $\mathfrak{s}(b\beta) \leq \mathfrak{s}(\alpha)$.

The outcome of the $\mathfrak{s}$-reduction step is then $\alpha - b\beta$ and β is called the *$\mathfrak{s}$-reducer*. When β $\mathfrak{s}$-reduces t we also say for convenience that $b\beta$ $\mathfrak{s}$-reduces α. That way b is introduced implicitly instead of repeating the equation $\mathrm{lt}\left(\overline{b\beta}\right) = t$. Just as for classic polynomial reduction, if $\mathrm{lt}\left(\overline{b\beta}\right) \simeq \mathrm{lt}(\overline{\alpha})$ then the $\mathfrak{s}$-reduction step is a *top $\mathfrak{s}$-reduction step* and otherwise it is a *tail $\mathfrak{s}$-reduction step*. Analogously we define the distinction for signatures: If $\mathfrak{s}(b\beta) \simeq \mathfrak{s}(\alpha)$ then the reduction step is a *singular $\mathfrak{s}$-reduction step* and otherwise it is a *regular $\mathfrak{s}$-reduction step*. The result of an $\mathfrak{s}$-reduction of $\alpha \in \mathscr{R}^m$ is $\gamma \in \mathscr{R}^m$ that has been calculated from α through a sequence of $\mathfrak{s}$-reduction steps such that γ cannot be further $\mathfrak{s}$-reduced. The reduction is a *tail $\mathfrak{s}$-reduction* if only tail $\mathfrak{s}$-reduction steps are allowed and it is a *top $\mathfrak{s}$-reduction* if only top $\mathfrak{s}$-reduction steps are allowed. The reduction is a *regular $\mathfrak{s}$-reduction* if only regular $\mathfrak{s}$-reduction steps are allowed. $\alpha \in \mathscr{R}^m$ is *$\mathfrak{s}$-reducible* if it can be $\mathfrak{s}$-reduced. If α $\mathfrak{s}$-reduces to γ and γ is a syzygy then we say that α *$\mathfrak{s}$-reduces to zero* even if $\gamma \neq 0$. Note that analogously to the classic polynomial reduction $\mathfrak{s}$-reduction is always with respect to a finite *basis* $\mathscr{G} \subset \mathscr{R}^m$. The $\mathfrak{s}$-reducers in $\mathfrak{s}$-reduction are chosen from the basis $\mathscr{G}$.

Definition 4. *Let I be an ideal in $\mathscr{R}$. A finite subset $\mathscr{G} \subset \mathscr{R}^m$ is a* signature Gröbner basis in signature T *(for I) if all $\alpha \in \mathscr{R}^m$ with $\mathfrak{s}(\alpha) = T$ $\mathfrak{s}$-reduce to zero w.r.t. $\mathscr{G}$. $\mathscr{G}$ is a* signature Gröbner

basis up to signature T *(for I) if* $\mathcal{G}$ *is a signature Gröbner basis in all signatures* S *such that* $S < T$. $\mathcal{G}$ *is a* signature Gröbner basis *(for I) if it is a signature Gröbner basis for* I *in all signatures. We denote* $\overline{\mathcal{G}} := \{\overline{\alpha} \mid \alpha \in \mathcal{G}\} \subset \mathcal{R}$.

Lemma 5 ([13]). *If* $\mathcal{G}$ *is a signature Gröbner basis then* $\overline{\mathcal{G}}$ *is a Gröbner basis.*

Next we give an algorithmic description of signature Gröbner bases.

Definition 6. *Let* $\alpha, \beta \in \mathcal{R}^m$ *such that* $\overline{\alpha} \neq 0$, $\overline{\beta} \neq 0$ *and let the monic least common multiple of* $\mathrm{lt}(\overline{\alpha})$ *and* $\mathrm{lt}\left(\overline{\beta}\right)$ *be* $\lambda = \mathrm{lcm}\left(\mathrm{lt}(\overline{\alpha}), \mathrm{lt}\left(\overline{\beta}\right)\right)$. *The* S-pair *between* α *and* β *is given by* $\mathrm{spair}(\alpha,\beta) := \frac{\lambda}{\mathrm{lt}(\overline{\alpha})}\alpha - \frac{\lambda}{\mathrm{lt}\left(\overline{\beta}\right)}\beta$. $\mathrm{spair}(\alpha,\beta)$ *is* singular *if* $\mathfrak{s}\left(\frac{\lambda}{\mathrm{lt}(\overline{\alpha})}\alpha\right) \simeq \mathfrak{s}\left(\frac{\lambda}{\mathrm{lt}\left(\overline{\beta}\right)}\beta\right)$. *Otherwise it is* regular.

Theorem 7 ([14]). *Let* T *be a module monomial of* $\mathcal{R}^m$ *and let* $\mathcal{G} \subset \mathcal{R}^m$ *be a finite basis. Assume that all regular S-pairs* $\mathrm{spair}(\alpha,\beta)$ *with* $\alpha, \beta \in \mathcal{G}$ *and* $\mathfrak{s}(\mathrm{spair}(\alpha,\beta)) < T$ $\mathfrak{s}$*-reduce to zero and all* $\boldsymbol{e}_i$ *with* $\boldsymbol{e}_i < T$ $\mathfrak{s}$*-reduce to zero. Then* $\mathcal{G}$ *is a signature Gröbner basis up to signature* T.

Note the similarity of Theorem 7 and Buchberger's criterion [2]. Theorem 7 suggests to consider only regular S-pairs for the computation of signature Gröbner bases. Thus in the following "S-pair" always refers to "regular S-pair".

4. A SIGNATURE-BASED GRÖBNER BASIS ALGORITHM

SBA is a generic signature-based algorithm in the vein of Buchberger's algorithm. Its efficiency depends on Line 7 that implements the Rewritten criterion (Lemma 8) to predict zero reductions.

The main differences to Buchberger's algorithm are the regular $\mathfrak{s}$-reduction in Line 8 and the set $\mathcal{H}$ which consists of found syzygies during the computation. Those syzygies are then used to remove useless elements in **Rewritable**. Moreover, **UpdateSyz** describes a generic subalgorithm that updates $\mathcal{H}$ and tries to find more syzygies. Clearly, its implementation depends on many factors and is out of scope of this extended abstract. Why is it important to know the syzygies?

Algorithm 1 SBA (Signature-based Gröbner Basis Algorithm)

Require: Ideal $I = \langle f_1, \ldots, f_m \rangle \subset \mathcal{R}$, monomial order $\leq$ on $\mathcal{R}$, extended to $\mathcal{R}^m$, a rewrite order $\trianglelefteq$ on $\mathcal{G} \cup \mathcal{H}$
Ensure: Signature Gröbner basis $\mathcal{G}$ for I, Gröbner basis $\mathcal{H}$ for $\mathrm{syz}(f_1, \ldots, f_m)$
1: $\mathcal{G} \leftarrow \emptyset$, $\mathcal{H} \leftarrow \emptyset$
2: $\mathcal{P} \leftarrow \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_m\}$
3: $\mathcal{H} \leftarrow \left\{f_i\boldsymbol{e}_j - f_j\boldsymbol{e}_i \mid 1 \leq i < j \leq m\right\} \subset \mathcal{R}^m$
4: **while** $\mathcal{P} \neq \emptyset$ **do**
5: $\quad \gamma = a\alpha - b\beta \leftarrow$ element of minimal signature w.r.t. $\leq$ from $\mathcal{P}$
6: $\quad \mathcal{P} \leftarrow \mathcal{P} \setminus \{\gamma\}$
7: $\quad$ **if** ($a\alpha$ is not rewritable) **and** ($b\beta$ is not rewritable) **then** // Rewritten criterion (Lemma 8)
8: $\quad\quad \gamma \leftarrow$ result of regular $\mathfrak{s}$-reducing β
9: $\quad\quad$ **if** $\overline{\gamma} = 0$ **then**
10: $\quad\quad\quad \mathcal{H} \leftarrow \mathcal{H} + \{\gamma\}$
11: $\quad\quad$ **else**
12: $\quad\quad\quad \mathcal{P} \leftarrow \mathcal{P} \cup \{\mathrm{spair}(\alpha,\gamma) \mid \alpha \in \mathcal{G}, \mathrm{spair}(\alpha,\gamma)$ regular$\}$
13: $\quad\quad\quad \mathcal{G} \leftarrow \mathcal{G} \cup \{\gamma\}$
14: $\quad\quad\quad$ **UpdateSyz**$(\mathcal{G}, \mathcal{H})$
15: **return** $(\mathcal{G}, \mathcal{H})$

Lemma 8 (Rewritten criterion). *For signature T* **SBA** *needs to handle exactly one $a\alpha \in \mathscr{R}^m$ from the set $\mathscr{C}_T = \{a\alpha \mid \alpha \in \mathscr{G} \cup \mathscr{H}, a \in \mathscr{M} \text{ and } \mathfrak{s}(a\alpha) = T\}$.*

Usually the Rewritten criterion as stated above is only defined for $\alpha \in \mathscr{G}$. The so-called **F5** *criterion* or *Syzygy criterion* is also included in Lemma 8: $\alpha \in \mathscr{H}$ means that α is a syzygy and its signature is equal to the lead term in $\mathscr{R}^m$. Clearly, whenever in a situation such that $\alpha \in \mathscr{H}$ and $a\alpha \in \mathscr{C}_T$ we choose $\alpha \in \mathscr{H}$ in $\mathscr{C}_T$, since then we do not need to do any computation in signature $T = \mathfrak{s}(a\alpha)$. The choice in Lemma 8 depends on a rewrite order $\trianglelefteq$:

Definition 9. *A* rewrite order $\trianglelefteq$ *is a total order on $\mathscr{G}$ such that $\mathfrak{s}(\alpha) \mid \mathfrak{s}(\beta) \Rightarrow \alpha \trianglelefteq \beta$.*

Thus it is well-defined to choose $\max_{\trianglelefteq} \mathscr{C}_T$ and all other corresponding S-pairs in signature T can be removed. Defining good and efficient rewrite orders is a field of active research.

5. Experimental results

In the following we present the results of the implementation of efficient variants of **SBA** in Singular. All computations were done on an INTEL® XEON® X5460 @ 3.16GHz processor with 64 GB of RAM. **STD** denotes the standard Gröbner basis implementation in Singular based on [12]. Random$(r, 2, 2)$ denotes dense affine, random systems with r generators in a polynomial ring with r variables, all of degree 2. Similar HRandom denotes homogeneous systems.

We see in Figure 1 that all variants of **SBA** always compute less zero reductions than **STD**. Besides the affine `cyclic-8` example **SBA** w.r.t $<_{\text{lt-pot}}$ is faster than **STD**, often several times. Currently we are working on further improvements for **SBA**, especially in terms of using **F4**-style linear algebra for the $\mathfrak{s}$-reduction process.

Benchmark	STD	SBA $<_{\text{pot}}$	SBA $<_{\text{d-pot}}$	SBA $<_{\text{lt-pot}}$
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	353
katsura-11-h	3,933	0	0	353
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

(A) Number of zero reductions

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

(B) Time in seconds

Figure 1. Experimental results for Singular $4-0-0$

References

[1] Arri, A. and Perry, J. The F5 Criterion revised. *Journal of Symbolic Computation*, 46(2):1017–1029, June 2011. Preprint online at `arxiv.org/abs/1012.3664`.

[2] Buchberger, B. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, University of Innsbruck, 1965.

[3] Buchberger, B. A criterion for detecting unnecessary reductions in the construction of Gröbner bases. In *EUROSAM '79, An International Symposium on Symbolic and Algebraic Manipulation*, volume 72 of *Lecture Notes in Computer Science*, pages 3–21. Springer, 1979.

[4] Buchberger, B. Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory. pages 184–232, 1985.

[5] Decker, W., Greuel, G.-M., Pfister, G., and Schönemann, H. SINGULAR *4-0-0 — A computer algebra system for polynomial computations*, 2014. http://www.singular.uni-kl.de.
[6] Eder, C. and Perry, J. F5C: A Variant of Faugère's F5 Algorithm with reduced Gröbner bases. *Journal of Symbolic Computation, MEGA 2009 special issue*, 45(12):1442–1458, 2010. dx.doi.org/10.1016/j.jsc.2010.06.019.
[7] Eder, C. and Roune, B. H. Signature Rewriting in Gröbner Basis Computation. In *ISSAC 2013: Proceedings of the 2013 international symposium on Symbolic and algebraic computation*, pages 331–338, 2013.
[8] Faugère, J.-C. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139(1–3):61–88, June 1999. http://www-salsa.lip6.fr/~jcf/Papers/F99a.pdf.
[9] Faugère, J.-C. A new efficient algorithm for computing Gröbner bases without reduction to zero F5. In *ISSAC'02, Villeneuve d'Ascq, France*, pages 75–82, July 2002. Revised version from http://fgbrs.lip6.fr/jcf/Publications/index.html.
[10] Galkin, V. Termination of original F5. http://arxiv.org/abs/1203.2402, 2012.
[11] Gao, S., Volny IV, F., and Wang, D. A new algorithm for computing Groebner bases (rev. 2011). http://www.math.clemson.edu/~sgao/papers/gvw_R130704.pdf, 2013.
[12] Gebauer, R. and Möller, H. M. On an installation of Buchberger's algorithm. *Journal of Symbolic Computation*, 6(2-3):275–286, October/December 1988.
[13] Roune, B. H. and Stillman, M. Practical Gröbner Basis Computation. In *ISSAC 2012: Proceedings of the 2012 international symposium on Symbolic and algebraic computation*, 2012.
[14] Roune, B. H. and Stillman, M. Practical Gröbner Basis Computation. http://arxiv.org/abs/1206.6940, 2012.

University of Kaiserslautern
E-mail address: ederc @ mathematik.uni-kl.de