

# SAGBI COMBINATORICS OF MAXIMAL MINORS AND A SAGBI ALGORITHM

WINFRIED BRUNS AND ALDO CONCA

**ABSTRACT.** The maximal minors of a matrix of indeterminates are a universal Gröbner basis by a theorem of Bernstein, Sturmfels and Zelevinsky. On the other hand it is known that they are not always a universal SAGBI basis. By an experimental approach we discuss their behavior under varying monomial orders and their extensions to SAGBI bases. These experiments motivated a new implementation of the SAGBI algorithm which is organized in a Singular script and falls back on Normaliz for the combinatorial computations. In comparison to packages in the current standard distributions of Macaulay 2, version 1.21, and Singular, version 4.2.1 and a package intended for CoCoA 5.4.2, it extends the range of computability by at least one order of magnitude.

## 1. INTRODUCTION

Let  $X$  be an  $m \times n$  matrix of indeterminates over a field  $K$ . The minors of  $X$ , i.e., the determinants of the submatrices of  $X$  generate subalgebras and ideals of the polynomial ring  $R = K[X_{ij} : i = 1, \dots, m, j = 1, \dots, n]$  that are important not only in algebraic geometry and commutative algebra, but also in representation theory and combinatorics. For a recent survey of the manifold approaches to the theory of ideals and subalgebras generated by minors we refer the reader to our recent volume [9] with Raicu and Varbaro. One of these approaches is via Gröbner and SAGBI bases. We trust that the reader is familiar with the notion of Gröbner basis. A SAGBI basis (terminology of Robbiano and Sweedler [31]) of a subalgebra  $A$  for a given monomial (or term) order on the ambient polynomial ring is a system of generators  $\mathcal{F}$  such that the set  $\text{in}(\mathcal{F})$  of initial monomials generates the initial algebra  $\text{in}(A)$ .

Let us assume  $m \leq n$ . By a theorem of Sturmfels [33], the  $t$ -minors, i.e., the determinants of the  $t \times t$ -submatrices, are a Gröbner basis of the ideal they generate under any diagonal monomial order—a diagonal monomial order chooses the product of the diagonal elements as the initial monomial of a  $t$ -minor. For the maximal minors, for which  $t = m$ , much more is true: They are a universal Gröbner basis, namely a Gröbner basis for an arbitrary monomial order. See Conca, De Negri and Gorla [15] for a simple proof of this theorem of Bernstein, Sturmfels and Zelevinsky [3, 35].

The subalgebra  $G(m, n)$  generated by the maximal minors is the homogeneous coordinate ring of the Grassmann variety of  $m$ -dimensional subspaces of  $K^n$ . For general  $t$  the  $t$ -minors are not a SAGBI basis of the algebra they generate. But for maximal minors this is true for diagonal orders, and the toric deformation of  $G(m, n)$  that it offers gives

---

2010 *Mathematics Subject Classification.* 13F50, 13F65, 13P10, 13P99, 14M25.

*Key words and phrases.* Grassmannian, maximal minors, SAGBI basis, Singular, Normaliz.

AC is supported by PRIN 2020355B8Y and by INdAM-GNSAGA.

comfortable access to many homological and enumerative invariants of  $G(m, n)$ . Other term orders for which maximal minors are a SAGBI bases of  $G(m, n)$  have been recently discovered, see for example [18] and [25].

In view of the Bernstein–Sturmfels–Zelevinsky theorem the question of universality arises also here. But it was disproved by Speyer and Sturmfels [32]: for a  $3 \times 6$  matrix there exists a lexicographical order under which the 3-minors are not a SAGBI basis of  $G(3, 6)$ .

The starting point of the project on which we report in this paper was the question of what can be said about reverse lexicographical orders in this respect. As we will see, revlex universality also fails, but needs a  $3 \times 8$  matrix for failure. In view of the special properties of normal monomial algebras it is also interesting to ask whether initial algebras of  $G(m, n)$  are always normal. This is surprisingly often true, and finding a counterexample needs more patience. By our experimental methods whose results are reported in Section 3, we cannot answer the question whether there is a finite universal SAGBI basis of  $G(m, n)$ . But our findings support the conjecture that this is true.

Our investigation of SAGBI bases of Grassmannians motivated the development of an algorithm for the computation of SAGBI bases in general. It is almost impossible to devise a really new algorithm, but the efficiency of the implementation is extremely important for this complicated task. Our algorithm is implemented in the Singular [19] script `sagbiNormaliz.lib`. While Singular is used for the polynomial arithmetic, newly developed functions of Normaliz [13] are called for the combinatorial tasks. The variants of our algorithms are explained in Section 4. One of them is based on control by Hilbert series.

In the last section we compare our algorithm to packages that come with the standard distributions of Macaulay2 [22], version 1.21, and Singular [19], version 4.2.1. We also compare it with a preliminary version of a CoCoA5 [2] package which will be in the standard distribution from version 5.4.2 on. Applications to different types of subalgebras show that our implementation extends the range of computability by more than one order of magnitude.

Normaliz 3.10.0 together with the Singular library `sagbiNormaliz.lib` and a new version of `normaliz.lib`, has been published in January 2023. On request we will provide the software and all data of this project to interested readers.

## 2. BASICS OF SAGBI BASES

The reader finds a compact discussion of SAGBI bases in [9, Sect. 1.3]. We use the notation developed there. Kreuzer and Robbiano [29, Sect. 6.6] give a more extensive introduction; also see Ene and Herzog [21] and Sturmfels [34]. SAGBI bases were introduced independently by Robbiano and Sweedler [31] and Kapur and Madlener [28]. The acronym SAGBI stands for “subalgebra analog to Gröbner bases of ideals” [31]. Some authors have adopted recently a new terminology, Khovanskii bases, for a notion that generalize that SAGBI bases but in this paper we keep the traditional name.

Let  $A \subset R = K[X_1, \dots, X_n]$  be a  $K$ -subalgebra and  $\mathcal{F}$  a (not necessarily finite) family of polynomials belonging to  $A$ . We assume that  $R$  is endowed with a monomial (or term) order  $<$ . In the following we will often simply speak of an *order*. One calls  $\mathcal{F}$  a SAGBI basis of  $A$  if the initial monomials  $\text{in}(f)$ ,  $f \in \mathcal{F}$ , generate the initial algebra  $\text{in}(A)$ . A

SAGBI basis is automatically a system of generators of  $A$ . If  $\mathcal{F}$  is finite, then  $A$  and  $K[\text{in}(\mathcal{F})]$  are connected by a flat deformation (Conca, Herzog and Valla [17]), and this allows the transfer homological and enumerative data from the toric algebra  $K[\text{in}(\mathcal{F})]$  to  $A$ . Chapter 6 of [9] exploits this approach for the investigation of algebras generated by minors.

SAGBI bases need not be finite, but can always be chosen countable. Therefore one must allow that  $\mathcal{F} = (f_u)_{u \in N}$  with  $N = \{1, \dots, p\}$  or  $N = \mathbb{N}$ . We will always assume that the members of  $\mathcal{F}$  are monic. This is evidently no essential restriction of generality as long as the base ring  $K$  is a field.

For us the following simple lemma is an important tool in the computation of SAGBI bases. For a  $\mathbb{N}$ -graded vector  $K$ -vector space  $V$  one defines the *Hilbert function* of  $V$  by

$$H(V, k) = \dim_K V_k, \quad k \in \mathbb{N}, \quad (2.1)$$

where  $V_k$  is the subspace of degree  $k$  elements of  $V$ .

**Lemma 1.** *Let  $R$  be a polynomial ring, endowed with a monomial order, and  $A$  a finitely generated graded subalgebra. Furthermore let  $\mathcal{F}$  be a family of polynomials in  $A$  and  $B = K[\text{in}(\mathcal{F})]$  the subalgebra generated by the monomials  $\text{in}(f)$ ,  $f \in \mathcal{F}$ . Then the following hold:*

- (1)  $H(B, k) \leq H(A, k)$  for all  $k$ .
- (2)  $\mathcal{F}$  is a SAGBI basis of  $A$  if and only if  $H(B, k) = H(A, k)$  for all  $k \in \mathbb{N}$ .

*Proof.* For a graded subspace  $V$  of the polynomial ring one has  $H(V, k) = H(\text{in}(V), k)$  for all  $k$  [9, 1.4.3]. Furthermore there are inclusions

$$B_k \subset \text{in}(A)_k, \quad k \in \mathbb{N},$$

and equality holds if and only if  $H(B_k, k) = H(\text{in}(A)_k, k)$  for all  $k$ . Together with Equation (2.1) this proves the lemma.  $\square$

To present  $A$  as a residue class ring of a polynomial ring, we choose  $P = K[Y_u : u \in N]$  and define a surjection

$$\phi : P \rightarrow A, \quad \phi(Y_u) = f_u, \quad u \in N. \quad (2.2)$$

The  $K$ -algebra  $K[\text{in}(\mathcal{F})]$  is a homomorphic image of  $P$ , as well, namely by the surjection

$$\psi : P \rightarrow K[\text{in}(\mathcal{F})], \quad \psi(Y_u) = \text{in}(f_u), \quad u \in N.$$

The kernel of  $\psi$  is generated by a set of binomials. In the terminology of [31] a binomial in  $\text{Ker } \psi$  is called a *tête-à-tête*.

A monomial in  $P$  is given by an exponent vector  $e = (e_u)_{u \in N}$  of natural numbers  $e_u$  of which all but finitely many are 0. We set  $Y^e = \prod_{u \in N} Y_u^{e_u}$ . Let  $F \in P$  be a polynomial, given as a  $K$ -linear combination of monomials,  $F = \sum_i a_i Y^{e_i}$  where the  $e_i$  are exponent vectors, and  $a_i \neq 0$  for all indices involved. We set

$$\begin{aligned} \text{in}_\phi(F) &= \max_i \text{in}(\phi(Y^{e_i})), \\ \text{init}_\phi(F) &= \sum_{\text{in}(\phi(Y^{e_i})) = \text{in}_\phi(F)} a_i Y^{e_i}. \end{aligned}$$

Note that in the definition of  $\text{in}_\phi(F)$  the maximum is taken over the initial monomials with respect to the monomial order on  $R$  so that it is a monomial in  $R$ . In contrast,  $\text{init}_\phi(F)$  is a polynomial in  $P$ . Since  $\phi(\text{init}_\phi(F))$  can be 0, in general  $\text{in}_\phi(F) \neq \text{in}(\phi(F))$ , and this cancellation of initials is the crucial aspect of SAGBI computation. One says that a polynomial  $F \in \text{Ker } \phi$  *lifts* a polynomial  $H \in \text{Ker } \psi$  if  $\text{init}_\phi(F) = \text{init}_\phi(H)$ .

We can now formulate the SAGBI *criterion* (see [9, 1.3.14]).

**Theorem 2.** *With the notation introduced, let  $\mathcal{B}$  be a set of binomials generating  $\text{Ker } \psi$ . Then the following are equivalent:*

- (1)  $\mathcal{F}$  is a SAGBI basis of  $A$ ;
- (2) every binomial  $b \in \mathcal{B}$  can be lifted to a polynomial  $F \in \text{Ker } \phi$ .

The Buchberger algorithm for a Gröbner basis of an ideal  $I$  starts from a system of generators  $G$  of  $I$ . Then one applies two steps, namely (i) the computation of the  $S$ -polynomials  $S(g_1, g_2)$ ,  $g_1, g_2 \in G$ , and (ii) their reductions modulo  $G$ . The nonzero reductions are then added to  $G$ , and the next round of  $S$ -polynomials of the augmented  $G$  and their reductions is run. This produces an increasing sequence of initial ideals  $\text{in}(G)$ . Because of Noetherianity the process stops after finitely many rounds with a Gröbner basis of  $I$ . The reduction of an  $S$ -polynomial  $S(g_1, g_2)$  to 0 is equivalent to the liftability of the “divided Koszul syzygy” of  $\text{in}(g_1)$  and  $\text{in}(g_2)$  to a syzygy of the polynomials  $g \in G$ .

The computation of SAGBI bases follows the same pattern. There are however two main differences: an analog of the divided Koszul syzygies does not exist, and ascending chains of monomial subalgebras of  $R$  need not stabilize. For SAGBI bases one must therefore compute a binomial system of generators of  $\text{Ker } \Psi$ , and one cannot expect the algorithm to stop. The analog of reduction is called subduction (we again follow [31]).

**Definition 3.** Let  $g \in R$ . Then  $r \in R$  is a *subduction of  $g$  modulo  $\mathcal{F}$*  if there exist monomials  $\mathcal{F}^{e_1}, \dots, \mathcal{F}^{e_m}$  and non-zero coefficients  $a_i \in K$  such that the following hold:

- (1)  $g = a_1 \mathcal{F}^{e_1} + \dots + a_m \mathcal{F}^{e_m} + r$ ;
- (2)  $\text{in}(\mathcal{F}^{e_i}) \leq \text{in}(g)$  for  $i = 1, \dots, m$ ;
- (3) no monomial  $\mu \in \text{supp}(r)$  is of type  $\text{in}(\mathcal{F}^e)$ .

The process that computes a subduction of  $f$  modulo  $\mathcal{F}$  is also called subduction. Here  $\text{supp}(r)$  is the set of monomials of  $R$  appearing in  $r$  with a nonzero coefficient. In the computation of SAGBI bases one can replace (3) by the weaker condition

- (3')  $\text{in}(r)$  is not of type  $\text{in}(\mathcal{F}^e)$ .

There is an obvious algorithm that produces a *subduction remainder*  $r$  in (3'): if  $\text{in}(f) = \text{in}(\mathcal{F}^e)$ , we replace  $f$  by  $f - a\phi(\mathcal{F}^e)$  where  $a$  is the leading coefficient of  $f$ , and iterate this *subduction step* as long as possible. The algorithm stops since the sequence of initial monomials is descending and descending sequences in a monomial order are finite. Once (3') is reached, one applies subduction steps iteratively to the remaining monomials to achieve the ‘tail subduction’ asked for by (3).

The algorithm (SAGBI) starts from the finite family  $\mathcal{F}_0$  generating the subalgebra  $A \subset R$ . Then one proceeds as follows:

- (1) Set  $i = 0$ .

- (2) Set  $\mathcal{F}' = \emptyset$  and compute a binomial system of generators  $\mathcal{B}_i$  of the kernel of  $\psi_i : P_i \rightarrow K[\mathcal{F}_i]$ ,  $P_i = K[Y_F : F \in \mathcal{F}_i]$ ,  $\psi_i(Y_F) = \text{in}(F)$ .
- (3) For all  $\beta \in \mathcal{B}_i$  compute the subduction  $r$  of  $\phi_i(\beta)$  modulo  $\mathcal{F}_i$ ,  $\phi_i$  given by the substitution  $Y_F \mapsto F$ ,  $F \in \mathcal{F}_i$ . If  $r \neq 0$ , make  $r$  monic and add it to  $\mathcal{F}'$ .
- (4) If  $\mathcal{F}' = \emptyset$ , set  $\mathcal{F}_j = \mathcal{F}_i$ ,  $P_j = P_i$ ,  $\mathcal{B}_j = \mathcal{B}_i$  for all  $j \geq i$  and stop.
- (5) Otherwise set  $\mathcal{F}_{i+1} = \mathcal{F}_i \cup \mathcal{F}'$ ,  $i = i + 1$  and go to (2).

It is not hard to see that  $\mathcal{F} = \bigcup_{i=0}^{\infty} \mathcal{F}_i$  is a SAGBI basis of  $A$ , and that the algorithm stops after finitely many steps if  $A$  has a finite SAGBI basis.

**Remark 4.** The computation of SAGBI bases is in general a very complex operation. However, in some cases it can offer a fast solution to problems that seem much simpler at first sight, but then turn out to be very hard. We discuss two cases.

(a) In our work preparing the article [8] with Varbaro, we did not succeed to compute the defining ideal of the algebra  $A$  generated by the  $2 \times 2$  minors of a  $4 \times 4$  matrix of indeterminates over a field  $K$  in “nonexceptional” characteristics, for example, characteristic 0. Singular, CoCoA and Macaulay did not stop in computing the defining ideal in reasonable time, not even up to degree 4.

From [7] it was however known that  $A$  has a finite SAGBI basis, and meanwhile more is known about it by work of Varbaro; see [9, 6.4.10]. In Section 5 we come back to the computation. It takes only 40 sec; see Table 1. With the right bookkeeping one can explicitly lift the final tête-a-tête to a defining ideal of the algebra. Since one wants the defining ideal in terms of the original system of generators, further processing is necessary and a minimization must follow. Nevertheless this is a feasible approach.

By the work of Huang et al. [27] it is now known that the defining ideal for algebras of 2-minors is generated in degree 2 and 3.

(b) Suppose that one wants to compute the Hilbert series of a Grassmannian explicitly by a computer algebra system. It is of course possible to use representation theoretic methods or classical approaches going back to Hodge [26]; see Braun [6] for explicit formulas. But these need preparations, and the same is true if one wants to exploit the Plücker relations. Computing a Gröbner basis by elimination is therefore tempting, but already for rather small cases it takes surprisingly long—already  $3 \times 9$  takes days. In contrast, the computation from a SAGBI basis given by the generating minors in a diagonal monomial order is almost instantaneous. Also the explicit computation of the SAGBI basis with respect to a diagonal monomial order is very fast. See Remark 12(f) for computational data.

### 3. SAGBI COMBINATORICS OF MAXIMAL MINORS

Let  $K$  be a field and  $X$  an  $m \times n$  matrix of indeterminates with  $m \leq n$ . By  $\mathcal{M}$  we denote the set of  $m$ -minors of  $X$ , i.e., the determinants of the submatrices

$$(X_{iu_j} : i = 1, \dots, m, j = 1, \dots, m), \quad u_1 < \dots < u_m.$$

The subalgebra  $G(m, n) = K[\mathcal{M}]$  of  $R = K[X_{ij} : i = 1, \dots, m, j = 1, \dots, n]$  is an object of classical algebraic geometry, namely the homogeneous coordinate ring of the Grassmannian of  $m$ -dimensional vector subspaces of  $K^n$  in its Plücker embedding. A “natural”

monomial order on  $R$  is lexicographic (or degree reverse lexicographic) for the order

$$X_{11} > \cdots > X_{1n} > X_{21} > \cdots > X_{2n} > \cdots > X_{m1} > \cdots > X_{mn}. \quad (3.1)$$

It is *diagonal*: the product of the indeterminates in the diagonal is the initial monomial of each minor in  $\mathcal{M}$ . The standard bitableaux are a  $K$ -basis of  $G(m, n)$  (see [9, Chap 3]), and this implies that  $\mathcal{M}$  is a SAGBI basis of  $G(m, n)$  for every diagonal monomial order on  $R$ . To the best of our knowledge, this was first observed by Sturmfels [34]. This toric deformation gives a comfortable access to the cohomological and enumerative properties of  $G(m, n)$ . For example see [9, Sect. 6.2].

In view of Lemma 1 it is crucial for our experimental approach to compute the Hilbert series

$$H_{G(m, n)}(t) = \sum_{k=0}^{\infty} (\dim_K G(m, n)_k) t^k.$$

Usually we work with the *normalized* degree on  $G(m, n)$  in which the  $m$ -minors have degree 1. So  $G(m, n)_k$  is the degree  $km$  homogeneous component of  $G(m, n)$  in the standard grading of  $R$ . Since the Hilbert series of  $G(m, n)$  and its initial algebra coincide and the initial algebra is normal, the computation of the Hilbert series by Normaliz is almost instantaneous.

It takes some work to show that  $\mathcal{M}$  is a Gröbner basis of the ideal  $I_m = I_m(X)$  of maximal minors of  $X$  with respect to a diagonal monomial order. But much more is true: by a theorem of Bernstein–Sturmfels–Zelevinsky [3, 35],  $\mathcal{M}$  is a *universal* Gröbner basis of  $I_m$ , a Gröbner basis with respect to every monomial order on  $R$ . See [15] for the simple proof of Conca, De Negri and Gorla and [16] for further developments. This raises the question whether  $\mathcal{M}$  is also a universal SAGBI basis for  $G(m, n)$ . While this is true for  $m = 2$  [9, 5.3.6], it fails already for  $m = 3$ ,  $n = 6$ , as observed by Speyer and Sturmfels [32]: there exists a lexicographic order on  $R$  for which  $\mathcal{M}$  is *not* a SAGBI basis of  $G(m, n)$ . But is  $\mathcal{M}$  *universally revlex*, namely a SAGBI basis for all degree reverse lexicographic orders on  $R$ ?

Before we answer this question, let us outline two strategies for the investigation of it and related questions. The first strategy is very simple: after fixing the matrix format, choose an order of the indeterminates, extend it to a (reverse) lexicographic order and check whether  $\mathcal{M}$  is a SAGBI basis for this order by comparing the Hilbert functions. (Instead of varying the order, we use a random permutation of the matrix entries.) This strategy is realized in a Singular script and the Hilbert series are computed by Normaliz. In the cases in which  $\mathcal{M}$  is not a SAGBI basis, one can furthermore try to extend it to a full SAGBI basis by the algorithm documented in Section 4.

While the random search suggests reasonable working hypotheses, it cannot prove statements about universality for which we must exhaust all relevant monomial orders. Even if one tries to use all the symmetries of  $G(m, n)$ , it is impossible to scan all (reverse) lexicographic orders. Instead we start from the *matching fields* in the terminology of Sturmfels–Zelevinsky [35]: a matching field assigns each minor a monomial in its Laplace expansion. It is called *coherent* if it always selects the initial monomial with respect to a monomial order. To reduce the number of matching fields, we use a result of Sturmfels and Zelevinsky: given a coherent matching field, there are exactly  $m(m-1)$

indeterminates that do not appear in any monomial (we call them the “missing” indeterminates). Sturmfels and Zelevinsky also explain that the location of these missing indeterminates satisfies certain restrictions, for example in each row one finds exactly  $(m - 1)$  of them. Hence the configuration of the missing indeterminates subdivides the set of coherent matching fields. For  $m = 3$  and  $n \geq 6$ , up to row and column permutations, there are exactly 4 types. See Figure 1 in which the entries 0 denote the missing indeterminates.

$$\begin{array}{ll}
 1: \begin{bmatrix} & 0 & 0 & \dots \\ 0 & & 0 & \dots \\ 0 & 0 & & \dots \end{bmatrix} & 2: \begin{bmatrix} & 0 & 0 & \dots \\ 0 & & & 0 & \dots \\ 0 & 0 & & & \dots \end{bmatrix} \\
 3: \begin{bmatrix} & 0 & 0 & & \dots \\ 0 & & & 0 & \dots \\ 0 & & & & 0 & \dots \end{bmatrix} & 4: \begin{bmatrix} & & & & 0 & 0 & \dots \\ & & & 0 & 0 & & \dots \\ 0 & 0 & & & & & \dots \end{bmatrix}
 \end{array}$$

FIGURE 1. Types of matching fields for  $m = 3$

We call a matching field *lex (revlex) compatible* if there exists a (reverse) lexicographic order that produces the given matching field as the sequence of the initial monomials of  $\mathcal{M}$ . The following observation saves computation time.

**Proposition 5.** *Type 4 is not lex compatible.*

*Proof.* Among the indeterminates one entry is largest with respect to the order, say  $X_{uv}$ . However, regardless of which it is, there always exist a minor involving 3 columns such that both monomials in its Laplace expansion that are divisible by  $X_{uv}$  are excluded since they both hit a 0 in one of the rows different from row  $u$ .  $\square$

The types are scanned individually. Even after these preparations it would take too long to create all matching fields for a given type and then check whether they are lex or revlex compatible, and if so, whether  $\mathcal{M}$  is a SAGBI basis for such an order. Usually there are several such orders; but it depends only on the matching field whether  $\mathcal{M}$  is a SAGBI basis. As soon as the matching field is fixed, this is only a question of whether the corresponding monomial algebra and  $G(m, n)$  have the same Hilbert series.

Therefore we choose an incremental approach that is realized in a C++ program. Let  $\mathcal{M} = \{\Delta_1, \dots, \Delta_N\}$ ,  $N = \binom{n}{m}$ . The matching fields under consideration are sequences of monomials  $\delta_1, \dots, \delta_N$  such that  $\delta_i$  appears in the Laplace expansion of  $\Delta_i$  and avoids the indeterminates that are excluded by the chosen type. The notion of lex or revlex compatibility extends naturally to initial subsequences of a matching field. For a given initial subsequence  $\gamma_1, \dots, \gamma_u$  let  $\Gamma(\gamma_1, \dots, \gamma_u)$  be the set of all compatible matching fields that extend  $\gamma_1, \dots, \gamma_u$  by  $\gamma_{u+1}, \dots, \gamma_N$ . We want to compute  $\Gamma(\emptyset)$  where  $\emptyset$  stands for the empty initial subsequence. This is done via the recursive relation

$$\Gamma(\gamma_1, \dots, \gamma_u) = \bigcup_{\gamma_{u+1}} \Gamma(\gamma_1, \dots, \gamma_u, \gamma_{u+1})$$

where  $\gamma_{u+1}$  satisfies the following conditions:

- (1)  $\gamma_{u+1}$  appears in the Laplace expansion of  $\Delta_{u+1}$ ,
- (2)  $\gamma_{u+1}$  is not excluded by the given type, and
- (3)  $\gamma_1, \dots, \gamma_u, \gamma_{u+1}$  is compatible.

Less formally, we extend a compatible initial sequence  $\gamma_1, \dots, \gamma_u$  in all possible ways. Condition (3) is the crucial test: if  $\gamma_1, \dots, \gamma_u$  is not extensible by at least one  $\gamma_{u+1}$ , the recursion stops and we backtrack to  $\gamma_1, \dots, \gamma_{u-1}$  and try the next choice of  $\gamma_u$ .

Let us state the results of the two experimental approaches:

**Theorem 6.** *Let  $m = 3$ .*

- (1)  $\mathcal{M}$  is a universal SAGBI basis for  $n \leq 5$ , but not for  $n \geq 6$ .
- (2)  $\mathcal{M}$  is a universally revlex SAGBI basis for  $n \leq 7$ .
- (3) There exist a lexicographic order for  $n = 6$  and a reverse lexicographic order for  $n = 8$  such that  $\mathcal{M}$  is not a SAGBI basis for them.

Unexpectedly often we have observed that both  $K[\text{in}(M)]$  and the full initial algebra  $\text{in}(K[\mathcal{M}])$  are normal:

**Theorem 7.** *Let  $m = 3$ .*

- (1) For lex orders  $K[\text{in}(\mathcal{M})]$  is normal for  $n \leq 9$ .
- (2) For revlex orders  $K[\text{in}(\mathcal{M})]$  is normal for  $n \leq 8$ .
- (3) For  $n = 9$  there exists a revlex order such that  $\mathcal{M}$  is a SAGBI basis, but  $\text{in}(K[\mathcal{M}])$  is not normal.
- (4) For  $n = 10$  there exists a lex order such  $K[\text{in}(\mathcal{M})]$  is not normal.

As a concrete example let us give a matrix for Theorem 7(3) where  $[u \mid v]$  denotes  $X_{uv}$ :

$$\begin{pmatrix} [1 \mid 4] & [3 \mid 1] & [1 \mid 5] & [2 \mid 5] & [3 \mid 4] & [2 \mid 8] & [2 \mid 9] & [2 \mid 6] & [2 \mid 1] \\ [1 \mid 9] & [1 \mid 3] & [2 \mid 2] & [2 \mid 7] & [1 \mid 6] & [3 \mid 5] & [3 \mid 2] & [1 \mid 1] & [2 \mid 3] \\ [1 \mid 8] & [2 \mid 4] & [3 \mid 7] & [3 \mid 9] & [3 \mid 8] & [1 \mid 2] & [3 \mid 3] & [1 \mid 7] & [3 \mid 6] \end{pmatrix}.$$

The indeterminates are ordered as in (3.1) and the monomial order is the degrevlex extension; as mentioned, we have permuted the entries of the matrix instead of changing the order of the indeterminates.

Even our recursive method for finding compatible matching fields does not reach  $n = 10$ . Already for  $n = 9$  it needs weeks of computation time, whereas it finishes in hours for  $n = 8$ . So one can say that we were rescued by the random search that produced counterexamples exactly one step beyond the limit of computability.

The experimental evidence that we have collected, makes us optimistic for the following

**Conjecture 8.**  *$G(m, n)$  has a finite universal SAGBI basis.*

The conjecture is supported by overwhelming experimental evidence for  $G(3, 6)$ ; in fact, we expect that the universal SAGBI basis has 15 elements of (normalized) degree 2 in addition to the 3-minors. We are confident that we can extend our experiments to checking the conjecture for  $m = 3$  and  $n \leq 8$ . It is already clear that  $n = 9$  is out of reach, not only because of the large number of cases, but also since the algorithm of Section 4 must often give up if the degrees of the polynomials in the SAGBI basis exceeds 7, and we have seen cases in which degree 11 is reached. Even if the combinatorial computations



should be still doable, the complexity of the polynomial computations and the available memory set a limit.

**Remark 9.** (a) While  $\mathcal{M}$  fails to be a SAGBI basis much earlier and much more often for lex orders than for revlex ones, we have observed that the missing polynomials usually have considerably lower degree in the lex case.

(b) Instead of lex and revlex orders one can experiment with arbitrary orders. At least for  $m = 3$  and  $n \leq 8$ , all matching fields found for arbitrary orders are lex or revlex compatible.

(c) The algebra  $G(m, n)$  is a retract of the Rees algebra  $\mathcal{R}(I_m)$  by degree selection, and therefore the equality  $K[\text{in}(\mathcal{M})] = \text{in}(K[\mathcal{M}])$  is a necessary condition for  $\mathcal{R}(I_m) = \text{in}(\mathcal{R}(I_m))$ . Not surprisingly, it is not a sufficient condition, as many counterexamples demonstrate. Note that  $\mathcal{R}(I_m) = \text{in}(\mathcal{R}(I_m))$  is equivalent to  $\text{in}(I_m)^k = \text{in}(I_m^k)$  for all  $k$ . Even if this does not hold in general, it often starts to fail for unexpectedly large  $k$ .

As a concrete example consider the matrix

$$\begin{pmatrix} [2 | 4] & [1 | 5] & [3 | 5] & [3 | 6] & [1 | 1] & [1 | 6] & [2 | 7] \\ [1 | 4] & [2 | 6] & [1 | 7] & [3 | 4] & [3 | 3] & [2 | 5] & [2 | 3] \\ [1 | 2] & [3 | 7] & [3 | 2] & [3 | 1] & [1 | 3] & [2 | 1] & [2 | 2] \end{pmatrix}$$

where we use the same notation and monomial order as in the example following Theorem 7. Despite of  $K[\text{in}(\mathcal{M})] = \text{in}(K[\mathcal{M}])$  one has  $\mathcal{R}(I_m) \neq \text{in}(\mathcal{R}(I_m))$ , as the comparison of Hilbert series shows. Since the elements of  $\mathcal{M}$  have constant degree,  $\mathcal{R}(I_m)$  and  $\text{in}(\mathcal{R}(I_m))$  have a standard grading. With CoCoA [2] we have analyzed the binomial relations of  $K[\text{in}(\mathcal{M})]$ . With respect to the normalized bigrading of the Rees algebra there are 245 quadrics of bidegree  $(1, 1)$  and  $(0, 2)$ . Those of bidegree  $(1, 1)$  can be lifted since the minors form a universal Gröbner basis. Those of bidegree  $(0, 2)$  have standard degree 2, but the first degree in which the Hilbert series differ is 5. So they are liftable as well. The obstruction to equality is a relation of bidegree  $(1, 4)$  and standard degree 5. This implies that  $\text{in}(I_3)^k = \text{in}(I_3^k)$  for  $k \leq 3$ , but  $\text{in}(I_3)^4 \neq \text{in}(I_3^4)$ .

The algorithm (Gen) of Section 4 completes the SAGBI basis by exactly one more degree 5 element in a few seconds and confirms the analysis above: the additional element has bidegree  $(1, 4)$ . Consequences: (i)  $\text{in}(I_3^4)$  and  $\text{in}(I_3^4)$  differ in degree 13 and (ii)  $\text{in}(I_3^k) = \text{in}(I_3^4) \text{in}(I_3^{k-4})$  for  $k \geq 4$ .

#### 4. AN IMPLEMENTATION OF THE SAGBI ALGORITHM

Our implementation is based on Singular [19] and Normaliz [13]. We have realized three variants of the algorithm that we will explain below. They are organized in the Singular library `sagbiNormaliz.lib`, which in its turn connects to Normaliz for the combinatorial tasks via an extended version of the Singular library `normaliz.lib`. Both libraries will be published together with Normaliz version 3.10.0 (released in January 2023).

The interface offered by `normaliz.lib` writes input files for Normaliz, calls it, and then reads the output files. The Macaulay2 [22] interface `normaliz.m2` [12] is file based as well. The transfer of the Singular implementation to Macaulay2, together with an update of `normaliz.M2`, is not hard for an experienced Macaulay2 user. The Normaliz team would

be very grateful for help! A realization via the C++ class library libnormaliz would of course be preferable and simplify the implementation.

Independently of the SAGBI computations, version 3.10.0 of Normaliz has been augmented by functions for arbitrary positive affine monoids. Such a monoid represents the combinatorial skeleton of a monomial algebra  $M$ , namely the monoid of the exponent vectors of the monomials of  $M$ . Whereas the functions are implemented additively on the combinatorial side, we describe their results in the multiplicative language of monomial algebras:

- (1) A minimal system of generators, which is uniquely determined. (See Bruns and Gubeladze [10, Chap. 2] for the theory of affine monoids.). In fact, it is the set of *irreducible* elements of  $M$ , i.e., those monomials  $x$  that cannot be written as a product of monomials  $y, z \neq x$ . In addition, one can ask for the *representations* of the reducible elements in a given system of generators as power products of the irreducible ones.
- (2) A defining binomial ideal of  $A$ , given by a minimal system of generators. Such an ideal is often called *toric* and the (minimal) system of generators is a (minimal) *Markov basis*; see De Loera, Hemmecke and Köppe [20]. Contrary to (1), this is a time critical task. Normaliz has implemented the project-and-lift algorithm of Hemmecke and Malkin [24]. The Normaliz implementation is not inferior to Hemmecke and Malkin's 4ti2 [1].
- (3) The Hilbert series of  $M$ , which is the ordinary generating function of the enumeration of monomials by degree. It uses the classical commutation of Hilbert series via initial (monomial) ideals.

One needs the defining binomial ideal for the tête-a-tête, as is clear by their definition. The irreducible elements must be known for a minimal SAGBI basis and the control of the algorithm. The representations are necessary for subduction. One of our variants uses the Hilbert series, as we will explain below.

As far as a grading is involved, the algorithms assume the standard  $\mathbb{N}$ -grading of the ambient polynomial ring  $R$ . The extension to other positive gradings would be possible without much effort, not yet however the extension to multi-gradings because the current version of Normaliz does not allow them. For a (monomial) subalgebra  $A$  we work with the *normalized degree*. It is obtained as follows: first one computes the greatest common divisor of the degrees of the polynomials in a generating set, and then divides the standard degree by this “grading denominator”. This is compatible to Normaliz, which uses the normalized degree as well (unless one forbids the grading denominator).

Via the generating system  $\mathcal{F} = (f_u)_{u \in N}$  of the subalgebra  $A$ , the polynomial ring  $P = K[X_u : u \in N]$  is graded as well when we set  $\deg Y_u = \deg f_u$  for all  $u \in N$ . Under this grading all tête-a-tête are homogeneous.

All three variants proceed in “rounds” as described in the algorithm (SAGBI), but are modified in two variants:

- (Gen) The *general* variant is essentially (SAGBI). It stops when  $\mathcal{F}'$  in step (4) is empty. As this may never happen, the user can set a bound on the number of rounds. The general variant does not require a grading.

(Deg) After the computation of the tête-a-tête for a set  $\mathcal{F}_i$ , the *degree by degree* variant goes over the homogeneous components of the tête-a-tête and stops at degree  $d$  as soon as at least one of the subduction remainders is nonzero and therefore at least one new element  $f$  of the SAGBI basis has been found.

All subduction remainders then extend the SAGBI basis to degree  $d$ , and the “search degree” can be raised to  $d + 1$ . The stop criterion  $\mathcal{F}' = \emptyset$  is the same as for (Gen). It is reached in degree  $d$  when no homogeneous component of the tête-a-tête has a nonzero subduction remainder in degrees  $> d$ .

For (Deg) the user must set an upper bound for  $d$ . If it can be expected that the SAGBI basis is finite, the upper bound should be chosen very large.

(Hilb) The *Hilbert series controlled variant* refines (Deg). As input it not only needs a system of generators of the subalgebra  $A$ , but also the Hilbert series of  $A$ , given as a rational function by its numerator and denominator.

Let  $S$  be the SAGBI basis of  $A$  up to degree  $d$ . Then the Hilbert series of  $K[\text{in}(S)]$  is computed. If it agrees with the Hilbert series of  $A$ ,  $S$  is the complete SAGBI basis. Otherwise the “critical degree” is computed, i.e., the lowest degree in which the Hilbert functions of  $K[\text{in}(S)]$  and  $A$  differ, together with the difference.

This information not only tells us in which degree we must evaluate the tête-a-tête for  $K[\text{in}(S)]$  to find the next elements of the SAGBI basis, but also their number  $m$ . Therefore the subduction can stop as soon as  $m$  nonzero pairwise different remainders have been reached. They must be irreducible since they are not divisible by monomials of smaller degree and do not divide each other.

In contrast to (Gen) and (Deg), (Hilb) offers a perfect error control, which was extremely helpful during the development of the library.

(Gen) and (Deg) have been implemented in other packages (though not necessarily together) that will be named in Section 5, and only some details may vary. But we are not aware of an implementation of (Hilb).

All our variants return the (partial) SAGBI basis computed and an additional integer that takes the value 0, if the partial SAGBI basis is incomplete, the value 1 if completeness is unknown, and 2 if the SAGBI basis is complete.

**Remark 10.** (a) Both the evaluation of the tête-a-tête and a subduction step can be realized by the homomorphism  $\phi : P \rightarrow A$  of Equation (2.2). Both amount to mapping binomials from  $P$  to  $A$  via  $\phi$ . We use the Singular map functionality for it.

(b) For the subduction one has two choices: (i) to subduce the polynomials individually until the remainder has been found, or (ii) apply one subduction step to all polynomials simultaneously, and iterate this step as long as necessary. In our computations (ii) has proved to be the better choice.

**Remark 11.** The following comments will be illustrated by computational data in Section 5.

(a) If the given subalgebra  $A$  is graded, it is always advisable to use (Deg) or even (Hilb). (Gen) should be reserved for nongraded subalgebras: a stop after a certain number of rounds gives much less information on the SAGBI basis than the stop at a predefined degree. Even in graded cases in which the complete SAGBI basis is computed, (Deg) is usually better.

Subduction remainders of homogeneous polynomials of degree  $\leq d$  with respect to a family  $\mathcal{F}$  of homogeneous polynomials of degree  $\leq d$  remain untouched if  $\mathcal{F}$  is augmented by homogeneous polynomials of degree  $> d$ . In other words, the partial SAGBI basis computed by (Deg) or (Hilb) are increasing with respect to inclusion.

(b) The choice between (Deg) and (Hilb) is more difficult. (Hilb) knows exactly into which degree to look next and can often stop the subduction process much earlier than (Deg). On the other hand, the Hilbert series of  $A$  must be known, and in each round the Hilbert series of a monomial algebra must be computed.

The Hilbert series computation is very fast if the monomial algebra is normal since then the Normaliz algorithm based on triangulation and Stanley decomposition can be used. In the non-normal case it is a byproduct of the tête-a-tête computation that produces a Gröbner basis of the binomial ideal before shrinking it to a minimal system of generators, but the Hilbert series computation for the initial ideal of the tête-a-tête ideal does not come for free.

Another parameter is the unpredictable complexity of subduction, which in its turn depends crucially on the sparsity of the polynomials involved in it. So the decision between (Deg) and (Hilb) is a matter of the Hilbert series computation versus the subduction.

As for (Deg) the reader must set a degree bound for (Hilb), but can ask for a final check when it is reached.

(c) In some cases the time of the tête-a-tête computation depends significantly on the order of the polynomials entering it (see Remark 12(e)). Our explanation is that the Gröbner basis computation on which it is based depends on a monomial order and is *not* invariant under the permutation of coordinates. However, we have no suggestion how to choose an order for which the tête-a-tête computation is especially fast. As a step in this direction, the user can ask for a sorting of the polynomials at the beginning of each round. In the implementation it is fixed to be ascending in the degrevlex monomial order on the ambient polynomial ring  $R$ . Note, that both (Deg) and (Hilb) generate the elements of the SAGBI basis in ascending degree, but not necessarily in any more refined order.

## 5. COMPUTATIONAL DATA

In the following we give computation times for several examples and compare them to packages that are included in the computer algebra systems CoCoA5, Macaulay2 and Singular:

- (1) The computations with CoCoA 5 use a variation of the script developed by Anna Bigatti for [4] which is available at [5] and will be part of the standard distribution of CoCoA from version 5.4.2.
- (2) The Macaulay2 distribution, version 1.21 (December 2022) contains the package SubalgebraBases.m2 by Burr et al., version 1.3. It realizes the variant (Deg) and allows a degree bound.
- (3) The Singular distribution, version 4.2.1 (May 2021) contains the library sagbi.lib by Hackfeld, Pfister and Levandovskyy. It offers only the variant (Gen) with an optional bound on the number of rounds.

As far as we could complete the computations, all packages give the same results.

The original submission of this article and the preprint arXiv:2302.14345v1 were based on a pre-release version of Normaliz 3.10.0 and version 1.1 of SubalgebraBases.m2. For this revised version we have updated the computational data. The preprint [14] on SubalgebraBases.m2 was not yet available at the time of our original submission.

We now list our test examples. With the exception of  $(HK_2)$  and  $(2x2_2)$ , the base field has characteristic 0.

- $(HK_0)$  the subalgebra of  $K[X, Y, Z]$  generated by the polynomials  $X^6, X^5Y, Y^5Z, XZ^5, Y^6 + Y^3Z^3$ . It is taken from Han and Kwak [23] where it serves as a simple counterexample to the Eisenbud–Goto conjecture. Order is degrevlex.
- $(HK_2)$  The same, but over a field of characteristic 2.
- $(Pow_l)$  The subalgebra of  $K[X, Y, Z]$  generated by the polynomials  $X^6 + Y^6 + Z^6, X^7 + Y^7 + Z^7, X^8 + Y^8 + Z^8$ . The monomial order is lex.
- $(Pow_r)$  The same as  $(Pow_l)$ , but order degrevlex.
- $(2x2_0)$  The subalgebra of  $K[X_{ij} : i, j = 1, \dots, 4]$  generated by the 2-minors. The monomial order is diagonal.
- $(2x2_2)$  The same, but over a field of characteristic 2.
- $(3x6)$   $G(3, 6)$ , a nondiagonal lex order.
- $(3x7)$   $G(3, 7)$ , a nondiagonal lex order.
- $(3x8)$   $G(3, 8)$ , a nondiagonal lex order.
- $(3x9_l)$   $G(3, 9)$ , a nondiagonal lex order.
- $(3x9_r)$   $G(3, 9)$ , a nondiagonal degrevlex order.

Table 1 contains data of the examples that are relevant for the SAGBI computation. In the table the first column after the name of the example is the degree bound for all

ecample	norm deg			#SAGBI	times in minutes			
	bound	SAGBI	(Deg)		(Deg)	(Hilb)	CoCoA5	M2
$(HK_0)$	16	—	—	80	1:13.67	1:19.56	3:29.00	T
$(HK_2)$	16	—	—	16	0:00.91	0:00.91	0:00.54	0:02.89
$(Pow_l)$	200	—	—	28	0:26.16	0:20.87	21:27.51	—
$(Pow_r)$	200	—	—	46	1:24.92	1:10.98	78:58.44	T
$(2x2_0)$	10	3	7	89	0:35:78	0:9.72	56:17.28	—
$(2x2_2)$	15	6	13	130	2:20.30	O	T	—
$(3x6)$	10	2	4	21	0:00.45	0:00.41	0:0 .40	0:01.10
$(3x7)$	10	2	4	37	0:01.48	0:00.82	0:08.95	1:44.20
$(3x8)$	10	3	6	67	0:08.15	0:03.07	0:44.84	T
$(3x9_l)$	10	3	6	101	0:37.27	0:12.94	14:36.46	—
$(3x9_r)$	10	7	8	90	M	0:22:69	—	—

TABLE 1. sagbiNormaliz.lib vs. CoCoA5 and M2

four computations, (Deg), (Hilb), CoCoA5 and M2. The next column gives the maximum degree of the SAGBI basis, provided we could compute a complete SAGBI basis. The

third column lists to what degree the computation had to be run for (Deg) to finish in these cases. Then we find the cardinality of the (partial) SAGBI basis, followed by the computation times. Some computations failed because of error conditions or the excess of the time bound:

- T Time bound of 90 minutes exceeded. Because the time bound was exceeded for  $(3 \times 8)$  by Macaulay2, we did not try larger formats.
- M Lack of memory (max 32 GB), see Remark 13(d).
- O An intermediate Hilbert series could not be transferred from Normaliz to Singular because of Singular's bound of 32 bit for the type `int`.

The times are given in the format min:sec with two decimals for the seconds. The times have been taken on a Dell xps17 laptop with an Intel i7-11800H at 2.3 GHz, on a Dell server r6525 for Macaulay 2, and for CoCoA5, on a MacBook Pro with an Intel Quad-Core i7 at 2,3 GHz. For comparison, the r6525 have been multiplied by 0.5, and the Macbook times have been multiplied by the factor 0.73 measured by running  $(3 \times 7)$  with Macaulay2 on the three machines.

**Remark 12.** We add comments on specific examples.

(a) The bulk of the computation time for  $(HK_0)$  goes into the tête-a-tête computation which reaches rather high degrees. In contrast, the polynomials are very sparse so that the subductions are fast. This explains why (Deg) is faster than (Hilb). We have added the characteristic 2 case since it shows that the SAGBI basis depends significantly on the base field. In this case it shrinks from characteristic 0 to characteristic 2 (and 3), but the opposite can happen as well.

We expect that  $(HK_0)$  and  $(HK_2)$  do not have finite SAGBI bases.

(b) In contrast to  $(HK_0)$ ,  $(Pow_r)$  uses more time for the subduction by rather dense polynomials. We expect that the SAGBI bases are infinite.

(c) We have mentioned the  $2 \times 2$  minors of a  $4 \times 4$  matrix of indeterminates already in Remark 4(a). That the defining ideal was not computable for us about 10 years ago, while the SAGBI basis takes only 25 sec with (Deg) and only 10 sec with (Hilb) is remarkable.

In characteristic 2 the algebraic structure is very different from that in characteristic 0, and this is also visible in the SAGBI basis that becomes considerably larger. Since (Deg) went through we could compute the Hilbert series. But then (Hilb) failed since an intermediate Hilbert series could not be transferred because of an overflow in Singular.

(d) The evaluation of the tête-a-tête on the partial SAGBI basis computed can of course fail for lack of memory, as we see in (Deg) for  $(3 \times 9_r)$ . For (Deg) it is impossible to know that the SAGBI basis is already complete (as is clear from (Hilb)), and degree 8 power products of 3-minors can already be very long polynomials. Even if the memory of a larger machine could suffice, computation time can set a limit at this point.

(e)  $(HK_0)$  has been run with sorting, all the others without. It is the only case in which we have seen a significant difference in running time. Without sorting the computation times grow by about 1 min, for (Deg) as well as for (Hilb), and the terminal output reveals that the time difference stems from the tête-a-tête computation.

(f) The computation of the Hilbert series of  $G(3, 9)$  via a SAGBI basis with respect to a diagonal monomial order (not in Table 1), order takes only 12 sec.

We add some computations with (Gen) to get a comparison to the Singular library sagbi.lib. For it the number of rounds of (SAGBI) can be limited. We have confined ourselves to the algebras related by minors since the degrees of the tête-a-tête for (HK) and (HK) become extremely large after one or two rounds, and (Gen) must evaluate them fully. We have added  $(3 \times 9_d)$ ,  $G(3, 9)$  with a diagonal monomial order. In Table 2 T indicates

example	rounds		#SAGBI	(Gen)	Singular
	bound	SAGBI			
$(2 \times 2_0)$	10	3	89	1:18.89	T
$(3 \times 6)$	10	2	21	0:00.58	0:00.22
$(3 \times 7)$	10	2	37	0:01.87	F
$(3 \times 8)$	10	3	65	0:16.47	—
$(3 \times 9_l)$	10	3	101	1:45.72	—
$(3 \times 9_d)$	10	1	84	0:11.31	T

TABLE 2. sagbiNormaliz.lib vs. Singular sagbi.lib

that the computation was stopped after 1 hour without output, and F indicates a failure because of a segmentation fault in Singular.

**Remark 13.** As documented in Table 2, sagbi.lib computes  $(3 \times 6)$ . It failed for the others. In view of the failure for  $(3 \times 7)$  after  $> 40$  min we did not try  $(3 \times 8)$  or  $(3 \times 9_l)$ . It is certainly surprising that sagbi.lib cannot recognize that the 84 minors form already a SAGBI basis for  $(3 \times 9_d)$ .

## REFERENCES

- [1] 4ti2 team. 4ti2-A software package for algebraic, geometric and combinatorial problems on linear spaces. Available at <https://github.com/4ti2/4ti2>.
- [2] John Abbott, Anna Maria Bigatti, Lorenzo Robbiano. CoCoA: a system for doing Computations in Commutative Algebra. Available at <http://cocoa.dima.unige.it>
- [3] D. Bernstein and A. Zelevinsky. Combinatorics of maximal minors. J. Algebraic Combin. 2, 2 (1993), 111–121.
- [4] A. Bigatti and L. Robbiano. Saturations of subalgebras, SAGBI bases, and U-invariants. J. Symbolic Comput. 109 (2022), 259–282.
- [5] A. Bigatti, A CoCoA5 package for SAGBI bases, Available at <https://www.dima.unige.it/bigatti/data/ComputingSaturationsOfSubalgebras/>
- [6] L. Braun. Hilbert series of the Grassmannian and k-Narayana numbers. Commun. Math. 27 (2019), no. 1, 27–41.
- [7] W. Bruns and A. Conca. KRS and powers of determinantal ideals. Compositio Math. 111, 1 (1998), 111–122.
- [8] W. Bruns, A. Conca and M. Varbaro. Relations between the minors of a generic matrix. Adv. Math. 244 (2013), 17–206.
- [9] W. Bruns, A. Conca, C. Raicu and M. Varbaro. Determinants, Gröbner bases and cohomology. Springer Monographs in Mathematics, Springer, Cham 2022.
- [10] W. Bruns and J. Gubeladze. Polytopes, rings, and K-theory. Springer Monographs in Mathematics, Springer, Dordrecht, 2009.

- [11] W. Bruns and J. Herzog. Cohen-Macaulay rings. Rev. ed. Cambridge: Cambridge University Press, 1998.
- [12] W. Bruns and G. Kämpf. A Macaulay2 interface for Normaliz. *J. Softw. Algebra Geom.* 2 (2010), 15–19.
- [13] W. Bruns, B. Ichim, C. Söger and U. von der Ohe, Normaliz. Algorithms for rational cones and affine monoids. Available at <https://normaliz.uos.de>.
- [14] M. Burr, O. Clarke, T. Duff, J. Leaman, N. Nichols and E. Walker. SubalgebraBases in Macaulay2. Preprint <https://arxiv.org/abs/2302.12473>.
- [15] A. Conca, E. De Negri, and E. Gorla. Universal Gröbner bases for maximal minors. *Int. Math. Res. Not.* 11 (2015), 3245–3262.
- [16] A. Conca, E. De Negri, and E. Gorla. Universal Gröbner bases and Cartwright-Sturmfels ideals. *Int. Math. Res. Not.* 7 (2020), 1979–1991.
- [17] A. Conca, J. Herzog, and G. Valla. SAGBI bases with applications to blow-up algebras. *J. Reine Angew. Math.* 474 (1996), 113–138.
- [18] O. Clarke and F. Mohammadi, Toric degenerations of Grassmannians and Schubert varieties from matching field tableaux. *J. Algebra* 559 (2020), 646–678.
- [19] W. Decker, G.-M. Greuel, G. Pfister, and H. Schonemann. Singular 4-1-1 — A computer algebra system for polynomial computations. <http://www.singular.uni-kl.de>, 2018.
- [20] J. A. De Loera, R. Hemmecke and M. Köppe. Algebraic and geometric ideas in the theory of discrete optimization. MOS-SIAM Series on Optimization, 14. Society for Industrial and Applied Mathematics (SIAM), Philadelphia 2013.
- [21] V. Ene and J. Herzog. Gröbner bases in commutative algebra, vol. 130 of Graduate Studies in Mathematics. American Mathematical Society, Providence, RI, 2012.
- [22] D. R. Grayson and M. E. Stillman. Macaulay2, a software system for research in algebraic geometry. Available at <http://www.math.uiuc.edu/Macaulay2/>.
- [23] J. I. Han and S. Kwak. Projective surfaces in  $\mathbb{P}^4$  that are counterexamples to the Eisenbud-Goto regularity conjecture. Preprint <https://arxiv.org/abs/2210.07174>.
- [24] Hemmecke and P. N. Malkin. Computing generating sets of lattice ideals and Markov bases of lattices. *J. Symb. Comp.* 44, 1463–1476 (2009).
- [25] A. Higashitani, H. Ohsugi. Quadratic Gröbner bases of block diagonal matching field ideals and toric degenerations of Grassmannians. *J. Pure Appl. Algebra* 226 (2022), no. 2.
- [26] W.V.D.Hodge. Some enumerative results in the theory of forms. *Proc. Cambridge Phil. Soc.* 39, 24–26 (1943).
- [27] H. Huang, M. Perlman, C. Polini, C. Raicu, and A. Sammartano. Relations between the  $2 \times 2$  minors of a generic matrix. *Adv. Math.* 386 (2021).
- [28] D. Kapur and K. Madlener. A completion procedure for computing a canonical basis for a  $k$ -subalgebra. In *Computers and mathematics* (Cambridge, MA, 1989). Springer, New York, 1989, pp. 1–11.
- [29] M. Kreuzer and L. Robbiano. Computational commutative algebra II. Springer-Verlag, Berlin, 2005.
- [30] E. Miller and B. Sturmfels. Combinatorial commutative algebra, vol. 227 of Graduate Texts in Mathematics. Springer-Verlag, New York, 2005.
- [31] L. Robbiano and M. Sweedler. Subalgebra bases. In *Commutative algebra* (Salvador, 1988), vol. 1430 of Lecture Notes in Math. Springer, Berlin, 1990, pp. 61–87.
- [32] D. Speyer and B. Sturmfels. The tropical Grassmannian. *Adv. Geom.* 4, 3 (2004), 389–411.
- [33] B. Sturmfels. Gröbner bases and Stanley decompositions of determinantal rings. *Math. Z.* 205, 1 (1990), 13–144.
- [34] B. Sturmfels. Gröbner bases and convex polytopes, vol. 8 of University Lecture Series. American Mathematical Society, Providence, RI, 1996.
- [35] B. Sturmfels and A. Zelevinsky. Maximal minors and their leading terms. *Adv. Math.* 98, 1 (1993), 65–112.



UNIVERSITÄT OSNABRÜCK, INSTITUT FÜR MATHEMATIK, 49069 OSNABRÜCK, GERMANY

*Email address:* `wbruns@uos.de`

DIPARTIMENTO DI MATEMATICA, UNIVERSITÀ DI GENOVA, ITALY

*Email address:* `conca@dima.unige.it`