EQUATIONS FOR GL INVARIANT FAMILIES OF POLYNOMIALS

PAUL BREIDING, CHRISTIAN IKENMEYER, MATEUSZ MICHAŁEK, AND REUVEN HODGES

Abstract. We provide an algorithm that takes as an input a given parametric family of homogeneous polynomials, which is invariant under the action of the general linear group, and an integer d. It outputs the ideal of that family intersected with the space of homogeneous polynomials of degree d. Our motivation comes from Problem 13 in [26], which asks to find equations for the variety of quartic symmetroids. The algorithm heavily uses a database of specific Young tableaux and highest weight polynomials. We provide the database and the implementation of the database construction algorithm. Moreover, we provide a julia implementation to run the algorithm using the database, so that more varieties of homogeneous polynomials can easily be treated in the future.

In addition, we conduct a numerical experiment seeking to determine the degree of the variety of quartic symmetroids.

1. Introduction

Many mathematical models are defined by nonlinear maps $f : V \to W$ between vector spaces. The space V is called *parameter space* and W is called the *state space* of the model. For instance, such models are common in statistics and physics. The setting allows to generate possible outcomes of the model, by evaluating f. This is called the *forward problem.* On the other hand, the *inverse problem* is to decide if a point $w \in W$ belongs to the image of f, and if so, to determine its preimage.

In this article we focus on the case when f is a polynomial map. Under this assumption the forward problem consists in evaluating a system of polynomials, and the inverse problem is to solve a system of polynomial equations.

Our main aim is to describe the closure of the image of f, when V, W are complex vector spaces. The goal is to describe the polynomial equations that vanish on the image of f. Having such equations at hand decouples the inverse problem: for the decision problem, whether or not w is in the image of f, one can evaluate the polynomials at w instead of solving a system of equations. The former is much simpler than the latter.

The classical method to find equations relies on the computation of a lexicographic Gröbner basis [12, 21] to perform elimination of variables. It is a symbolic method, that in practice may be used only on small examples. Thus, the motivation for us is to describe an alternative algorithm that can go beyond these small cases. In general, this task is too ambitious. But, if we assume that the problem has some underlying symmetries, we can use the power of *representation theory* to reduce

P. Breiding has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 787840). C. Ikenmeyer was supported by the DFG grant IK 116/2-1.

complexity. In this paper, we make the following assumption for f: we require it to be mapping into a vector space of polynomials and we assume that the image of f is GL-invariant.

Assumption 1.1. We assume that $W = S^c(\mathbb{C}^n)$ is the space of homogeneous polynomials of degree c in n variables. Furthermore, we assume that the image of f is invariant under $\mathrm{GL}(\mathbb{C}^n)$, which acts by variable substitution.

Our motivation comes from Problem 13 in [26], which asks to find equations for the variety of quartic symmetroids. This is a subvariety of the vector space of homogeneous polynomials in $n = 4$ variables of degree $c = 4$. It is GL(4)-invariant. We address this problem in Section 5.

We note that the GL action both gives us many advantages and is very natural. Our ambient space $S^c(\mathbb{C}^n)$ of polynomials may be regarded as a space of varieties. Following Felix Klein Erlangen program geometric quantities should be group-invariant. Thus, very often when studying sets of polynomials, we would like those sets not to depend on the choice of the basis. This is precisely the GL invariance. Further, the space of polynomials vanishing is often huge, but the GL action reduces the complexity and allows us to describe it using just a few generators.

2. Contributions

We present an algorithm to study the image of f under Assumption 1.1. This algorithm produces the following: let

$$X := \overline{\mathrm{im}(f)}$$

be the closure of the image of f and let I be the ideal of polynomials that vanish on X. Given f and any d we return the minimal set of polynomials that under the GL action span I_d. This algorithm is exact, ie. does not rely on any approximations. However, instead of a purely symbolic algorithm that works with the parametrized variety X directly, a much more efficient implementation just samples from X (without approximations) and uses only the sampled points as input, which reduces the finding problem to a linear algebra problem. The details are given in Section 4. This means that due to unlucky sampling in principle the algorithm could output equations that are not actually equations. In practice the probability of this is extermely low and can be further reduced to an inverse exponentially small probability by running the algorithm several times. One of the algorithm's central ingredients is a *database* which contains basis of highest weight spaces for different plethysms.

The variety of quartic symmetroids consists of polynomials that are determinants of symmetric four by four matrices with entries that are linear forms in four variables. To distinguish it from the general X we will use another symbol for it:

$$Q := \{\det(x_0A_0 + x_1A_1 + x_2A_2 + x_3A_3) \mid A_i \in \mathbb{C}^{4\times 4} \text{ and } A_i^T = A_i, 0 \le i \le 3\}.$$

It is a $\mathrm{GL}(\mathbb{C}^4)$ invariant subvariety of $S^4(\mathbb{C}^4)$ of codimension 10. We apply our algorithm to this variety, and gives us the following result.

Theorem 2.1. *There are no equations for Q in degrees up to (including) 8.*

Our second contribution is a numerical experiment seeking to determine the degree of the variety of quartic symmetroids. Based on the results from Section 5 we make the following conjecture.

Conjecture 2.2. The degree δ of Q is $850000 \leq \delta \leq 851000$.

We combine numerical and symbolic methods in our algorithms. Both the numerical and the symbolic algorithm appeared (explicitly or implicitly by using highest weight polynomials as images of symmetrizations over the wreath product) in particular examples before [6, 7, 8, 18, 2, 10, 11, 9, 14, 19, 1, 23, 22, 20, 3, 24, 13]. However, to our knowledge, this is the first general implementation and the first one with which it is possible to check for equations of degree 8 on $S^4(\mathbb{C}^4)$. This is made possible by the use of an idea that we call *equivariant hash functions*, see Section 4. We provide the source code of our implementation and an easy to use user interface for future researchers to build upon.

We remark that new algorithms for evaluating highest weight polynomials have been developed very recently in [4]. No open source implementation of these algorithms is available, but in a special case (see [14]) the running time improvements seem to be of practical importance.

3. Representation theory

Representation theory can be very beneficial for large computations. In one line, it allows to replace a possibly high dimensional irreducible representation, by a one dimensional subspace—the span of the highest weight vector. We briefly recall the relevant concepts for our setting. For more details the reader is referred to [17, 21].

Every irreducible, polynomial representation $V = V_\lambda$ of $\mathrm{GL}(\mathbb{C}^n)$ is associated to a Young diagram λ with at most n rows. Fixing the torus $T \subset \mathrm{GL}(\mathbb{C}^n)$ of diagonal matrices the representation V of T is decomposable $V = \oplus_{\chi \in \mathbb{Z}^n} V_\chi$, where $tv = \chi(t)v$ for $v \in V_\chi$ and $\mathbb{Z}^n$ is the lattice of characters of the torus T. The lexicographically largest χ, say $\chi_0 = (l_1, \dots, l_n)$ is called the highest weight of V. The Young diagram λ has l_i boxes in the i-th row. We have $\dim V_{\chi_0} = 1$. The unique up to scaling element of V_{χ_0} is called the *highest weight vector*.

Example 3.1. Let $V = S^d\mathbb{C}^n$ be the d-th symmetric power of $\mathbb{C}^n$. It is an irreducible representation. The characters of the torus χ appearing in the representation correspond to n-tuples of nonnegative integers summing up to d. The highest weight is $(d, 0, \dots, 0) \in \mathbb{Z}^n$. The highest weight vector is $e_1 \cdots e_1$. The associated Young diagram is a row with d boxes.

More generally, for any representation V of $\mathrm{GL}(\mathbb{C}^n)$ a vector $v \in V$ is called a *highest weight vector* if it is an image of a highest weight vector in some irreducible representation V_χ under a $\mathrm{GL}(V)$-equivariant map. If $V = \bigoplus_\lambda V_\lambda^{\oplus a_\lambda}$ is the decomposition of V then a_λ equals the dimension of the vector space of highest weight vectors in V of weight λ. Further, any highest weight vector of weight λ uniquely determines a subrepresentation $V_\lambda \subset V$. In other words, representation theory allows to replace a possibly large representation V by much smaller spaces of highest weight vectors.

The main observation is that if $X \subset S^c(\mathbb{C}^n)$ is $\mathrm{GL}(\mathbb{C}^n)$ invariant then I_d is a representation of $\mathrm{GL}(\mathbb{C}^n)$, which is a subrepresentation of $S^d(S^c((\mathbb{C}^n)^*))$. The representation $S^d(S^c(\mathbb{C}^n)^*)$ is known as a plethysm. In general the formulas for its decomposition into irreducible representation are not known, and determining a combinatorial description for the multiplicities of irreducibles is Problem 9 in Stanley's list of open problems in algebraic combinatorics [25]. However, they are known up to $d \leq 5$ and for fixed d and c there are algorithms to find such decompositions. For general d and $c = 3$ even the task of deciding positivity of a_λ is NP-hard, see [16].

If $S^d(S^c((\mathbb{C}^n)^*)) = \bigoplus_{\lambda \vdash db}(S^\lambda)^{\oplus a_\lambda}$ is the decomposition, then we seek to find such subrepresentations $(S^\lambda)^{\oplus b_\lambda} \subset (S^\lambda)^{\oplus a_\lambda}$ that $I_d = \bigoplus_{\lambda \vdash db}(S^\lambda)^{\oplus b_\lambda}$. This is equivalent to finding a b_λ-dimensional linear subspace in the space of highest weight vectors in $(S^\lambda)^{\oplus a_\lambda}$. We provide a database of polynomials in

$$S^d(S^c) := S^d(S^c((\mathbb{C}^n)^*))$$

that for each λ provides a basis of the highest weight space of $(S^\lambda)^{\oplus a_\lambda}$. Finally, we apply exact linear algebra methods to find which combinations of those vectors vanish on X. This is done by finding exact random points of X giving linear conditions on highest weight spaces.

To generate a basis of the highest weight vectors in $S^d(S^c)$ one may first generate a basis of highest weight vectors of weight λ in $(S^c)^{\otimes d}$. This is obtained by applying the Pieri rule. As writing this basis in terms of tensors is quite memory and time consuming, it is much better to simply remember it in terms of semistandard Young tableaux. The symmetrizing operator $(S^c)^{\otimes d} \to S^d(S^c)$ maps this basis to a generating set. Out of that set one chooses a basis, using linear algebra. There are many choices to pick a basis out of a generating set. We choose a random initial element in the generating set and add it to our basis. Then we choose another random element in the generating set, and check if it is linearly independent to the current basis. If it is we add this new element to the basis. We repeat this process until the number of basis elements equals the multiplicity of S^λ in $S^d(S^c)$. To check linear independence it is enough to be able to evaluate a polynomial corresponding to a given Young tableaux at many points. We apply a method that allows fast evaluation, without the necessity to expand the whole highest weight vector.

4. Algorithm

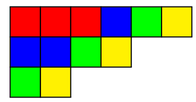
We describe here how to convert a Young tableau into a highest weight polynomial over the monomial basis. Evaluation at a point in X is then straightforward. In this way, if we can sample efficiently from X, we can evaluate the basis of a_λ many highest weight polynomials at a_λ sampled points, obtain a square matrix A of evaluations, and use linear algebra to compute $b_\lambda = \dim \ker A$.

We are given two natural numbers $d, c \in \mathbb{N}$. Moreover, we are given a so-called *isobaric Young tableau*. This is an top-left justified arrangement of dc many boxes with entries from $\{1, \ldots, d\}$ such that every entry appears exactly c times, see this example with $d = 4, c = 3$:

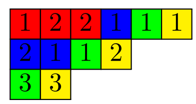
1	1	1	2	3	4
2	2	3	4		
3	4				

In fact, we may assume that the tableau is *semistandard*, which means that the entries are increasing within each column from top to bottom and they are nondecreasing within each row from left to right. The example above is semistandard. It is an open question whether or not it is possible to use semistandardness to get a computational advantage.

We color the boxes in the same color iff the have the same number, and then we remove the numbers:



Let μ_i denote the number of boxes in column i. Let $\mathfrak{S}_k$ denote the symmetric group on k letters. A *column permutation assignment* is defined as an assignment of numbers to the boxes such that in each column i each number from $\{1, \ldots, \mu_i\}$ appears exactly once. For example, this is a column permutation assignment:



Each column in a column permutation assignment specifies a permutation, so we can define the *sign* of a column permutation assignment to be the product of the signs of the permutations that correspond to the columns. The example above has sign $1 \cdot (-1) \cdot (-1) \cdot 1 \cdot 1 \cdot 1 = 1$.

To each column permutation assignment T we assign the word $w(T)$ that is obtained by reading from T first all entries from one color, then from the next, and so on. The order of colors and the order in which we read entries from the same color does not matter, because we define to words of length cd to be *equivalent* if they arise from each other by permuting symbols within the block $\{1, \ldots, c\}$ or within $\{c+1, \ldots, 2c\}$, and so on, or if they arise by permuting the d many blocks (in other words, they are equivalent iff they lie in the same orbit under the action of the wreath product $\mathfrak{S}_c \wr \mathfrak{S}_d$). The equivalence class of words $w(T)$ in the example above is $\{\{1,2,2\},\{1,1,2\},\{1,1,3\},\{1,2,3\}\}$. To every column permutation assignment T, let $\kappa(T)$ denote the equivalence class of $w(T)$. Consider the vector space spanned by all possible $\kappa(T)$, where we interpret distinct $\kappa(T)$ to be linearly independent unit vectors.

Up to a simple rescaling of the basis, the highest weight polynomial in monomial presentation is then

$$\sum_{\text{column permutation assignment } T} \operatorname{sgn}(T)\kappa(T).$$

A bottleneck in the computation of this polynomial is the number of column permutation assignments. For example, if $d = 8$, $c = 4$, then for the Young diagram with row lengths $\lambda = (8,8,8,8)$ we have $110\,075\,314\,176$ many column permutation assignments. Therefore it is imperative to perform as few operations as possible for each summand. Here are a few points which accelerate the computation:

(1) We use a Gray code to iterate through the sum so that the sign alternates for every summand. Therefore we never have to compute the sign of a permutation. The Gray code is a product of hardcoded Gray codes for small symmetric groups.
(2) We do not compute $\kappa(T)$, because it would require sorting. Instead we use an *equivariant hash function*: We can efficiently compute the hash function value for a column permutation assignment and the equivariance of the hash function guarantees that words that equivalent under the wreath product action are mapped to the same hash value. Since we know all possible

images of the hash function in advance, we can choose the parameters to make it collision-free in a precomputation step.

(3) To crucially speed up to computation the hash function value is not computed for each summand, but the hash function value is just *adjusted* at each step. This is possible, because the hash function is chosen as follows. Let $T_{i,j}$ be the jth entry in the ith colored block of T. Then, the hash function h is

$$h(T) := \sum_{i=1}^{d} \big(\sum_{j=1}^{c} \iota(T_{i,j}) \big)^k \bmod p$$

for a suitable $k \in \mathbb{N}$ and prime p, where $\iota(i)$ is the i-th entry in a fixed array of random numbers from $\{0, \ldots, p-1\}$. Raising to the k-th power is done by repeated squaring. The Gray code ensures that only two blocks are changed and only one entry in each block, which makes updating the hash value very efficient. To give a rough idea of the performance, after the precomputation of the hash function the summation over the 110 075 314 176 entries for $\lambda = (8, 8, 8, 8)$ takes only a few hours on a laptop.

Those ideas are incorporated into our implementation.

5. Numerical methods

The algorithm that we have described in the last section is symbolic. It is based on exact computations, thus yielding exact results. As we have demonstrated, it can go beyond the cases that the classical method relying on Gröbner basis [12, 21] can cope with.

Nevertheless, there are still limits to our algorithm with the current technology that numerical methods can surpass. For instance, our main theorem (Theorem 2.1) shows that no equations of degree at most 8 vanish no the variety of quartic symmetroids Q. But we could not find the minimal degree d, for which there are equations; i.e., such that $I_d \neq \emptyset$. Numerical methods, although not exact, can help to make an educated guess for those numbers. In this last section we want to explain this.

We first explain an approach on how to compute the degree of X. Applying this to the special case of quartic symmetroids leads to Conjecture 2.2. Thereafter, we will discuss that one can in principle extract the minimal d, such that $I_d \neq \emptyset$, from this computation. This poses new numerical challenges, however.

5.A. **Experiment: the degree of the variety of quartic symmetroids.** We make a numerical computation to determine the degree of Q. The approach described in this section can easily be generalized to the general situation involving X, but for simplicity we will stick to the special situation with Q.

We use affine coordinates by setting $A_0 = \mathbf{1}_4$ equal to the 4×4 identity matrix. Then, we have the following situation:

$$\begin{aligned} f : V &\to W, \\ a = (A_1, A_2, A_3) &\mapsto \text{coefficients of } \det(x_0 \mathbf{1}_4 + x_1 A_1 + x_2 A_2 + x_3 A_3) \end{aligned}$$

and $\dim V = 31$ and $\dim W = 35$. To determine the dimension of Q we compute the rank of the Jacobian matrix of f at a random point. We get $\dim Q = 25$. This implies that the dimension of the fibers of f for a general point $h \in Q$ is

$\dim f^{-1}(h) = 6$. The degree of Q is thus the number of isolated complex solutions of the following system of 31 polynomial equations in the 31 variables a:

$$\mathcal{B} \cdot f(a) = \beta \text{ and } \mathcal{C} \cdot a = \gamma, \tag{5.1}$$

where $\mathcal{B} \in \mathbb{C}^{25\times 35}$, $\beta \in \mathbb{C}^{25}$, $\mathcal{C} \in \mathbb{C}^{6\times 31}$ and $\gamma \in \mathbb{C}^6$ are chosen randomly. Due to the fact that we can easily generate one solution to this system, we can exploit monodromy by varying $\mathcal{B}$ and β in loops and numerically tracking the solutions along those loops. This produces new solutions for (5.1). The details of this technique are, for instance, explained in [15]. We used the implementation in `HomotopyContinuation.jl` [5] for our case of quartic symmetroids. After three months the algorithm had found 849998 solutions for (5.1). At this point the computation was aborted manually, because it hadn't found any more solution in a week. This led us to state Conjecture 2.2.

5.B. **Further directions.** Here, we explain an approach for answering the following question: Given a homogeneous polynomial map $f : \mathbb{C}^a \to \mathbb{C}^b$ what is the dimension of the vector space I_d of polynomials of degree d that vanish on the image?

The basic idea is this: suppose that we have run the algorithm from the previous section. Then, we have found a linear space L in $W = S^c(\mathbb{C}^n)$ and points $w_1, \ldots, w_\delta \in X \cap L$, such that δ is the degree of X. Any equation that vanishes on X also vanishes on the w_i. We now discuss when the reverse is true. If this holds, we can check numerically by solving a system of linear equations, whether or not there are equations of a fixed degree d vanishing on the X. Note that this does *not* yield equations for X. Furthermore, we can use coordinates for L for doing the linear algebra. This kind of *dimensionality reduction* can provide a significant reduction in computational complexity.

Let us write $b := \dim(S^c(\mathbb{C}^n))$ and the image of f is invariant under the action of $\mathrm{GL}(\mathbb{C}^n)$. We ask for the dimension of I_d, that is the degree d part of I. It should be emphasized that this is naturally a problem in linear algebra, as I_d is a vector space. Each point $x \in X$ determines a linear condition on the space $S^d(\mathbb{C}^b)$, giving rise to a hyperplane containing I_d. In fact, I_d is the intersection of all such hyperplanes. For dimensional reasons, it would be enough to pick consecutively random $x \in X$ and intersect the hyperplanes in $S^d(\mathbb{C}^b)$, until the intersection stabilizes. This is indeed sometimes done in practice, but the main problem is the large dimension $\binom{d+b-1}{d}$ of the space $S^d(\mathbb{C}^b)$. The method we describe is particularly useful if:

(1) the codimension of X is small,
(2) the degree of X is small.

From now on we work in the projective space $\mathbb{P}(\mathbb{C}^b)$ and consider X as a projective variety. Let $e := \operatorname{codim} X$.

We pick a random subspace $L = \mathbb{P}^e \subset \mathbb{P}(\mathbb{C}^b)$. By Bertini's theorem $\mathbb{P}^e$ intersects X in $\delta = \deg X$ many smooth points

$$S = L \cap X.$$

A random linear form h_1 is not a zero divisor in the ring $\mathbb{C}[y_1, \ldots, y_b]/I$, hence we have an exact sequence:

$$0 \to \mathbb{C}[y_1, \ldots, y_b]/I \to \mathbb{C}[y_1, \ldots, y_b]/I \to \mathbb{C}[y_1, \ldots, y_b]/(I + (h_1)),$$

where the first map is multiplication by h_1. Hence, the Hilbert series of $I + (h_1)$ equals $(1 - t)$ times the Hilbert series of I. In particular, the numerators of the Hilbert series are the same. The number of linear $h_1, \dots, h_l$ such that h_{i+1} is not a zero divisor modulo $I + (h_1, \dots, h_i)$ for every $0 \le i < l$ is governed by the depth of the (localization of the) ring $\mathbb{C}[y_1, \dots, y_b]/I$. Depth is always at most equal to the dimension and the cases when equality holds are called Cohen-Macaulay.

After choosing $e = \operatorname{codim} X$ many linear forms, we arrive at the ring

$$\mathbb{C}[y_1, \dots, y_b]/(I + (h_1, \dots h_e))$$

which describes S as a *projective* scheme. In general, the ideal $(I + (h_1, \dots h_e))$ may have an embedded component at zero, however this again does not happen if X is arithmetically Cohen-Macaulay (which means that its coordinate ring is Cohen-Macaulay). In practice, we next choose an affine linear form and add it to the ideal to represent S as a finite subset of an affine space.

In particular, if our variety X is arithmetically Cohen-Macaulay then the Hilbert function of the finite set S in fact encodes the numerator of the Hilbert series of X. In any case a nonzero element in I_d gives rise to a nonzero element in $I(S)_d$. As long as $I(S)_d = 0$ we also have $I_d = 0$, hence we do not have to look for equations in those degrees. Further, in smallest degree d such that $I(S)_d \neq 0$ we have $\dim I_d = \dim I(S)_d$ in the Cohen-Macaulay case.

Example 5.1. In the following example we construct a toric ring of small depth. Consider the map:

$$(x_1, x_2, x_3, x_4) \mapsto (x_1x_2^4, x_1x_2^3x_3, x_1x_2x_3^3, x_1x_3^4, x_1x_2^4x_4, x_1x_2^3x_3x_4, x_1x_2x_3^3x_4, x_1x_3^4x_4).$$

The image is a toric variety of projective dimension two and degree eight. It is minimally generated by nine quadrics and twelve cubics. If we intersect the image with two affine linear forms we obtain eight points. These eight points do not contribute to new linear equations, however their ideal has thirteen minimal generators in degree two.

Thus, if we know S, we may estimate $\dim I_d$ using linear algebra approach described above, but now we deal with points in the ambient space of dimension $e = \operatorname{codim} X$. Hence, we have to solve $\deg X$ many *linear* equations in $\binom{d+e-1}{d}$ many variables.

Numerical methods help us both: to obtain S and to solve the linear equations. To generate $\mathbb{P}^e$ we take a span of $e + 1$ many random/general points of X. We obtain $\mathbb{P}^e$ together with $e + 1$ many points of S. To generate all of $S = \{w_1, \dots, w_\delta\}$ we apply the monodromy method from the previous subsection.

Now a new problem arises. As our points are just approximations of the points in S, if we ask for the rank of the matrix associated to the system of linear equations, symbolically it will always be nondegenerate. Further, the matrix we obtain depends on the choice of the basis of degree d polynomials we take. The idea is to look at the singular values of the associated matrix in the basis. This allows us to discover the rank of the approximated matrix.

We are confident that the approach, that we have just described, will be helpful in future computations involving GL invariant families of polynomials. For the particular case of quartic symmetroids Q, we were not able to apply it, yet. We first must settle Conjecture 2.2.

References

[1] Abdelmalek Abdesselam. Feynman diagrams in algebraic combinatorics. *Séminaire Lotharingien de Combinatoire [electronic only]*, 49:B49c, 45 p., electronic only–B49c, 45 p., electronic only, 2002.

[2] Abdelmalek Abdesselam, Christian Ikenmeyer, and Gordon Royle. 16,051 formulas for Ottaviani's invariant of cubic threefolds. *Journal of Algebra*, 447:649–663, 2016.

[3] Daniel J. Bates and Luke Oeding. Toward a salmon conjecture. *Experimental Mathematics*, 20(3):358–370, 2011.

[4] Markus Bläser, Julian Dörfler, and Christian Ikenmeyer. On the complexity of evaluating highest weight vectors. arXiv:2002.11594, 2020.

[5] Paul Breiding and Sascha Timme. HomotopyContinuation.jl: A Package for Homotopy Continuation in Julia. In *International Congress on Mathematical Software*, pages 458–465. Springer, 2018.

[6] Peter Bürgisser, Matthias Christandl, and Christian Ikenmeyer. Even partitions in plethysms. *Journal of Algebra*, 328(1):322 – 329, 2011.

[7] Peter Bürgisser and Christian Ikenmeyer. Geometric complexity theory and tensor rank. *Proceedings 43rd Annual ACM Symposium on Theory of Computing 2011*, pages 509–518, 2011.

[8] Peter Bürgisser and Christian Ikenmeyer. Explicit lower bounds via geometric complexity theory. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 141–150, New York, NY, USA, 2013. ACM.

[9] Peter Bürgisser, Christian Ikenmeyer, and Greta Panova. No occurrence obstructions in geometric complexity theory. *Journal of the American Mathematical Society*, 32:163–193, 2019. A conference version appeared in: Proceedings IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS 2016), 386–395.

[10] Man-Wai Cheung, Christian Ikenmeyer, and Sevak Mkrtchyan. Symmetrizing tableaux and the 5th case of the Foulkes conjecture. *Journal of Symbolic Computation*, 2016.

[11] Luca Chiantini, Jonathan D. Hauenstein, Christian Ikenmeyer, Joseph M. Landsberg, and Giorgio Ottaviani. Polynomials and the exponent of matrix multiplication. *Bulletin of the London Mathematical Society*, 50(3):369–389, 2018.

[12] David Cox, John Little, and Donal OShea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013.

[13] Noah Daleo, Jonathan Hauenstein, and Luke Oeding. Computations and equations for segre-grassmann hypersurfaces. *Portugaliae Mathematica*, 73, 08 2014.

[14] Julian Dörfler, Christian Ikenmeyer, and Greta Panova. On geometric complexity theory: Multiplicity obstructions are stronger than occurrence obstructions. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece.*, pages 51:1–51:14, 2019. journal version accepted for publication in SIAM J Appl Alg Geom (SIAGA).

[15] Timothy Duff, Cvetelina Hill, Anders Jensen, Kisun Lee, Anton Leykin, and Jeff Sommars. Solving polynomial systems via homotopy continuation and monodromy. *IMA Journal of Numerical Analysis*, 39(3):1421–1446, 04 2018.

[16] Nick Fischer and Christian Ikenmeyer. The computational complexity of plethysm coefficients. arXiv:2002.00788, 2020.

[17] William Fulton and Joe Harris. *Representation theory: a first course*, volume 129. Springer Science & Business Media, 2013.

[18] Jonathan Hauenstein, Christian Ikenmeyer, and Joseph M. Landsberg. Equations for lower bounds on border rank. *Experimental Mathematics*, 22(4):372–383, 2013.

[19] Christian Ikenmeyer and Umangathan Kandasamy. Implementing geometric complexity theory: On the separation of orbit closures via symmetries. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 713–726, New York, NY, USA, 2020. Association for Computing Machinery.

[20] Shrawan Kumar. A study of the representations supported by the orbit closure of the determinant. *Compositio Mathematica*, 151, 09 2011.

[21] Mateusz Michałek and Bernd Sturmfels. *Invitation to Nonlinear Algebra*. 2020.

[22] Luke Oeding and Claudiu Raicu. Tangential varieties of segre-veronese varieties. *Collectanea Mathematica*, 65, 11 2011.

[23] Giorgio Ottaviani. Five lectures on projective invariants, lecture notes for trento school, september 2012. arXiv:1305.2749, to appear in Rendiconti del Seminario Matematico, Torino, 2013.

[24] Claudiu Raicu. 3×3 minors of catalecticants. *Mathematical Research Letters*, 20, 07 2013.

[25] Richard P. Stanley. Positivity problems and conjectures in algebraic combinatorics. In *Mathematics: frontiers and perspectives*, pages 295–319. Amer. Math. Soc., Providence, RI, 2000.

[26] Bernd Sturmfels. Problems for the Einstein Program proposal. Available at https://math.berkeley.edu/~bernd/Einstein2014.pdf.

Institut für Mathematik, Technische Universität Berlin, Germany

University of Liverpool, United Kingdom

Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany and University of Konstanz

U. Illinois at Urbana-Champaign, Urbana, IL 61801, USA