AN ALGORITHMIC METHOD OF PARTIAL DERIVATIVES

CORNELIUS BRAND

Charles University

KEVIN PRATT

Carnegie Mellon University

Abstract. We study the following problem and its applications: given a homogeneous degree-d polynomial g as an arithmetic circuit, and a $d \times d$ matrix X whose entries are homogeneous linear polynomials, compute $g(\partial/\partial x_1, \ldots, \partial/\partial x_n) \det X$. By considering special cases of this problem we obtain faster parameterized algorithms for several problems, including the matroid k-parity and k-matroid intersection problems, faster *deterministic* algorithms for testing if a linear space of matrices contains an invertible matrix (Edmond's problem) and detecting k-internal outbranchings, and more. We also match the runtime of the fastest-known deterministic algorithm for detecting subgraphs of bounded pathwidth, while using a new approach.

Our approach raises questions in algebraic complexity related to Waring rank and the exponent of matrix multiplication ω. In particular, we study a new complexity measure on the space of homogeneous polynomials, namely the bilinear complexity of a polynomial's apolar algebra. Our algorithmic improvements are reflective of the fact that for the degree-n determinant polynomial this quantity is at most $O(n2^{\omega n})$, whereas all known upper bounds on the Waring rank of this polynomial exceed $n!$.

E-mail addresses: cbrand@iuuk.mff.cuni.cz, kpratt@andrew.cmu.edu.

1. Introduction

Let $\mathcal{S}_d^n \coloneqq \mathbb{R}[x_1, \ldots, x_n]_d$ denote the vector space of homogeneous polynomials of degree d in n variables with real coefficients. We define the *apolar inner product* $\langle \cdot, \cdot \rangle : \mathcal{S}_d^n \times \mathcal{S}_d^n \to \mathbb{R}$ via

$$\langle f, g \rangle \coloneqq f(\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n})g. \tag{1}$$

This inner product (alternativley known as the *Sylvester product*, the *Bombieri inner product*, or the *Fischer-Fock inner product*) originated in 19th century invariant theory [Syl52] and has become a source of interest in computer science due to algorithmic applications. In a typical application, one first identifies some easy-to-evaluate generating polynomial g whose coefficients encode solutions to a combinatorial problem. This information can then be recovered by computing $\langle f, g \rangle$ for a suitable choice of f. While this quantity is often $\#P$ hard to compute exactly (this follows from the coming example), in special cases it can be efficiently approximated. This approach has led to new algorithms for problems as disparate as approximating permanents and mixed discriminants [Gur05], sampling from determinantal point processes [AGR16], Nash social welfare maximization [AOGSS17], and approximately counting subgraphs of bounded treewidth [Pra19].

For example, given a matrix $A \in \mathbb{R}^{n \times n}$, define $P_A \coloneqq \prod_{i=1}^n \sum_{j=1}^n A_{i,j} x_j$. Then

$$\langle x_1 x_2 \cdots x_n, P_A \rangle = \sum_{\sigma \in \mathfrak{S}_n} \prod_{i=1}^n A_{i,\sigma(i)}$$

is the permanent of A.

As a second example, given a directed graph G with n vertices, let A_G be the matrix with entry (i, j) equal to the variable x_i if there is an edge from vertex v_i to vertex v_j, and zero otherwise. By the trace method,

$$\operatorname{tr}(A_G^d) = \sum_{\substack{\text{closed walks} \\ (v_{i_1}, v_{i_2}, \ldots, v_{i_d}) \in G}} x_{i_1} \cdots x_{i_d} \in \mathcal{S}_d^n.$$

Now let $A \in \mathbb{R}^{d \times n}$ be a matrix any d columns of which are linearly independent. Let $X = A \cdot \operatorname{diag}(x_1, \ldots, x_n) \cdot A^T$. By the Cauchy-Binet Theorem,

$$\det X = \sum_{S \in \binom{[n]}{d}} \det(A_S)^2 \prod_{i \in S} x_i.$$

(Here A_S refers to the $d \times d$ submatrix of A with columns indexed by the set S.) Since any d columns in A are linearly independent, $\det(A_S)^2 > 0$ for all $S \in \binom{[n]}{d}$. Then note that the result of differentiating $\operatorname{tr}(A_G^d)$ by $\det(A_S)^2 \prod_{i \in S} x_i$ is positive if there is a simple cycle on the vertices $\{v_i : i \in S\}$, and zero otherwise. It follows that $\langle \det X, \operatorname{tr}(A_G^d) \rangle > 0$ if and only if G contains a simple cycle of length d.

Motivated by such examples, we consider the algorithmic task of computing (1) when f is the determinant of a symbolic matrix (a matrix whose entries are homogeneous linear polynomials) and g is given as an arithmetic circuit. This has applications to parameterized algorithms, yielding faster algorithms for the matroid k-packing and k-parity problems, the first deterministic $\operatorname{poly}(n)$-time algorithm for testing if a subspace of matrices of dimension $O(\log n)$ contains an invertible matrix, faster deterministic algorithms for detecting k-internal outbranchings, among others. Starting from the observation of the above example, we also give a deterministic $\varphi^{2d} \operatorname{poly}(n) < 2.62^d \operatorname{poly}(n)$-time algorithm for detecting simple cycles of length d in an n vertex graph. Here $\varphi \coloneqq \frac{1+\sqrt{5}}{2}$ is the golden ratio. This brushes up against the fastest-known algorithm for this problem which has runtime $2.55^d \operatorname{poly}(n)$ [Tsu19]. Our algorithm also generalizes to detecting subgraphs of bounded pathwidth, unexpectedly matching the runtime of the fastest-known algorithm for this problem [FLPS16], while using a new, very mechanical, approach.

Our algorithms for computing special cases of (1) turn out to be equivalent to algorithms for performing arithmetic in a certain algebra $\mathcal{A}_f$ associated to f, namely the *apolar algebra* of f. Apolar algebras (also

known as Artinian Gorenstein algebras) have been studied extensively since the work of F.S. Macaulay in 1916 [Mac94] and are ubiquitous in algebraic combinatorics; see e.g. [AHK18]. As a first step towards extending our approach, we then study $\mathbf{R}(\mathcal{A}_f)$, the bilinear complexity of the apolar algebra of a polynomial f. This gives upper bounds on the number of non-scalar multiplications needed to compute (1) in the white-box setting (Proposition 5). We will show in Example 3 that in fact previous methods in exact algorithms (specifically, those for subset convolution) necessarily made use of upper bounds on this quantity.

To obtain further algorithmic improvements, we raise the following algebraic question:

Question 1. Let $\mathcal{T}_{n,d}$ be the set of all $f \in \mathcal{S}_d^n$ such that $f = \sum_{S \in \binom{[n]}{d}} c_S \prod_{i \in S} x_i$, where $c_S > 0$ for all S. What is $B(n,d) \coloneqq \min(\dim \mathrm{Diff}(f) : f \in \mathcal{T}_{n,d})$? Here $\mathrm{Diff}(f)$ denotes the vector space spanned by the partial derivatives of all orders of f.

This question was asked in [Pra19, Question 73], but it was not known that an answer would have algorithmic implications. Our algorithms make use of the upper bound $B(n,d) < \varphi^{2d}$, obtained by taking f to be the determinant of a symbolic Hankel matrix. We remark that it is not hard to show that $B(n,d) \geq 2^d$.

1.1. **Previous approaches to computing the inner product** (1)**.** One special case of (1) that has been the source of several recent breakthroughs is when f and g are *real stable* polynomials with nonnegative coefficients; see e.g. [Gur08, AGV18]. In this case $\langle f, g \rangle$ can be approximated (up to a factor of $e^{d+\varepsilon}$) in polynomial time by a reformulation as a convex program [AG17, Theorem 1.2]. For the cases we consider, however, f and g will not be real stable.

Another approach is based on *Waring rank* upper bounds [Bar96, Gur06, Gly13, Pra19]. The Waring rank of $f \in \mathcal{S}_d^n$, denoted $\mathbf{R}_S(f)$, is defined as the minimum r such that $f = \sum_{i=1}^r c_i \ell_i^d$ for linear forms $\ell_1, \ldots, \ell_r \in \mathcal{S}_1^n$ and scalars $c_1, \ldots, c_r$. For example, the identity

$$x_1x_2x_3 = \frac{1}{24}\left[(x_1+x_2+x_3)^3 - (x_1+x_2-x_3)^3 - (x_1-x_2+x_3)^3 - (-x_1+x_2+x_3)^3\right]$$

shows that $\mathbf{R}_S(x_1x_2x_3) \leq 4$. Waring rank has been studied in invariant theory and algebraic geometry since the 1850's [IK99, Introduction] and has gained recent attention for its applications to algebraic complexity [BIP19, CHI+18]. Its relevance to (1) is due to the following fact, which can be verified by a direct calculation: if $f = \sum_{i=1}^r c_i(a_{i,1}x_1 + \cdots + a_{i,n}x_i)^d$, then for all $g \in \mathcal{S}_d^n$,

$$\langle f, g \rangle = d! \sum_{i=1}^r c_i g(a_{i,1}, \ldots, a_{i,n}).$$

Hence upper bounds on $\mathbf{R}_S(f)$ yield algorithms for computing $\langle f, g \rangle$. Furthermore, it was shown in [Pra19, Theorem 6] that with only evaluation access to g, $\mathbf{R}_S(f)$ queries are *required* to compute this inner product. Unfortunately, $\mathbf{R}_S(f)$ is usually prohibitively large; for instance, the Waring rank of almost all $f \in \mathcal{S}_d^n$ is at least $\lceil \binom{n+d-1}{d}/n \rceil$ [Lan12, Section 3.2].

In [Pra19] this difficulty was overcome by studying relaxations of Waring rank. For instance, while the elementary symmetric polynomial $e_{n,d}$ is known to have Waring rank roughly $n^{d/2}$ [Lee16], for all $\varepsilon > 0$ there exists a polynomial $f_\varepsilon \in \mathcal{S}_d^n$ with Waring rank only $O(\frac{4.075^d \log n}{\varepsilon^2})$ that ε-*approximates* $e_{n,d}$, in the sense that for all $g \in \mathcal{S}_d^n$,

$$(1-\varepsilon)\langle e_{n,d}, g \rangle \leq \langle f_\varepsilon, g \rangle \leq (1+\varepsilon)\langle e_{n,d}, g \rangle.$$

This fact lends itself to parameterized algorithms for problems such as approximately counting simple cycles. We remark that this discrepancy between $\mathbf{R}_S(e_{n,d})$ and the Waring rank of polynomials "close" to $e_{n,d}$ can be understood from a parameterized algorithms perspective as reflecting the fact that exactly counting cycles of a given length is $\#W[1]$ hard [FG04], whereas the problem of approximately counting cycles has been known to admit parameterized algorithms since [AR02].

1.2. **Our approach.** In contrast to previous approaches, we consider the white-box setting where g is given as an arithmetic circuit C. For us, f will always be the determinant of a symbolic matrix X that is given as input. Our algorithms work by inductively evaluating C, computing at each gate the result of differentiating $\det X$ by the polynomial computed by C at that gate[1]. At the end of the algorithm, the output gate of C will therefore contain $\langle g, f\rangle = \langle f, g\rangle$. Here we make use of the fact that $\langle \cdot, \cdot \rangle$ is symmetric, so one can either think about differentiating f by g or vice versa.

The key to our approach is that for a symbolic d by d matrix X, the vector space of partial derivatives of $\det X$ has dimension at most 4^d, and in some important cases this bound can be improved to φ^{2d}. So while one might naïvely represent an element in this space as a linear combination of $\binom{n+d}{d}$ monomials, doing so generally includes a significant amount of unnecessary information. Instead, we represent elements in this space as linear combinations of minors (determinants of submatrices) of X, which are specified by pairs of increasing sequences.

We will start by giving in our Theorem 1 an algorithm for the special but important case when g is computed by a *skew* circuit, meaning one of the two operands to each multiplication gate is a variable or a scalar:

Theorem 1. *Let C be a skew arithmetic circuit computing $g \in \mathcal{S}_d^n$, and let $X = (\ell_{i,j})_{i,j\in[d]}$ be a symbolic matrix with entries in $\mathcal{S}_1^n$. Then we can compute $\langle \det X, g\rangle$ with $4^d|C|\operatorname{poly}(d)$ arithmetic operations.*

Our algorithm for Theorem 1 only uses linear algebra and basic properties of differentials.

Of particular interest will be the case of Theorem 1 when X is a Hankel matrix, meaning that $(X)_{i,j} = (X)_{i+k,j-k}$ for all $k = 0, \dots, j - i$. For example, the generic 3×3 Hankel matrix is

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ x_2 & x_3 & x_4 \\ x_3 & x_4 & x_5 \end{bmatrix}.$$

We show the following improvement in this special case:

Theorem 2. *Let C be a skew arithmetic circuit computing $g \in \mathcal{S}_d^n$, and let $X = (\ell_{i,j})_{i,j\in[d]}$ be a symbolic Hankel matrix with entries in $\mathcal{S}_1^n$. Then we can compute $\langle \det X, g\rangle$ with $\varphi^{2d}\operatorname{poly}(d)|C|$ arithmetic operations. Here $\varphi \coloneqq \frac{1+\sqrt{5}}{2}$ is the golden ratio.*

The improvement in Theorem 2 over Theorem 1 is facilitated by the fact that the space of partial derivatives of the determinant has dimension about 4^d, whereas the space of partial derivatives of the determinant of a Hankel matrix has dimension less than φ^{2d}. We also make use of use of linear relations in the space of minors of a Hankel matrix originally studied in commutative algebra [Con98].

1.3. **Applications to parameterized algorithms.** Theorem 1 yields faster algorithms for the k-matroid intersection and matroid k-parity problems. These are the following problems:

Problem 1 (Matroid k-Parity)**.** Suppose we are given a matrix $B \in \mathbb{Q}^{km\times kn}$ representing a matroid M with groundset $[kn]$, and a partition π of $[kn]$ into parts of size k. Decide if the union of any m parts in π are independent in M.

Problem 2 (k-Matroid Intersection)**.** Suppose we are given matrices $B_1, \dots, B_k \in \mathbb{Q}^{m\times n}$ representing matroids $M_1, \dots, M_k$ with the common groundset $[n]$. Decide if $M_1, \dots, M_k$ share a common base.

We show in Theorems 3 and 4 that these can be solved in time $4^{km}\operatorname{poly}(N)$, where N denotes the size of the input. When $k = 2$ these are the classic matroid parity and intersection problems and can be solved in polynomial time, but for $k > 2$ they are NP-hard. The first algorithms for general k faster than naïve enumeration were given by Barvinok in [Bar95], and had runtimes $(km)^{2k+1}4^{km}\operatorname{poly}(N)$ and $(km)^{2k}4^{k^2m}\operatorname{poly}(N)$, respectively. A parameterized algorithm for Problem 1 was also given by Marx

[1]By "differentiating f by g" we mean applying the differential operator $g(\partial/\partial x_1, \dots, \partial/\partial x_n)$ to f.

in [Mar09] where it was used to give fixed-parameter tractable algorithms for several other problems, including Problem 2. The fastest algorithms prior to our work were due to Fomin et al. [FLPS16] and had runtime $2^{km\omega}\operatorname{poly}(N)$, where $\omega < 2.373$ is the exponent of matrix multiplication [LG14].

By combining Theorem 1 with a known construction of the determinant as a skew circuit [MV97], we obtain a faster deterministic algorithm for the following problem:

Problem 3 (SING)**.** Given matrices $A_1, \ldots, A_n \in \mathbb{Q}^{d\times d}$, decide if their span contains an invertible matrix. Equivalently, decide if $\det \sum_{i=1}^{n} x_i A_i \not\equiv 0$.

We show that SING can be solved in $4^d \operatorname{poly}(N)$ time in our Corollary 2. In particular, this establishes that SING $\in \mathcal{P}$ for subspaces of matrices of logarithmic dimension. The fastest previous algorithm, given by Gurvits in [Gur03], had runtime $2^d d! \operatorname{poly}(N)$ and made use of an upper bound of $2^d d!$ on $\mathbf{R}_S(\det_d)$. This problem was originally studied by Edmonds for its application to matching problems [Edm67]. While it is known to admit a simple randomized polynomial time algorithm as was first observed by Lovász [Lov79], a *deterministic* polynomial time algorithm would imply circuit lower bounds that seem far beyond current reach [KI04]. As a result, variants of SING have attracted attention, leading to a recent breakthrough in the non-commutative setting [GGOW19].

Theorem 2 yields the following applications:

Corollary 6. *The following problems admit deterministic algorithms running in time $\varphi^{2d}\operatorname{poly}(n)$:*

1. *Deciding whether a given directed n-vertex graph has a directed spanning tree with at least d non-leaf vertices,*
2. *Deciding whether a given edge-colored, directed n-vertex graph has a directed spanning tree containing at least d colors,*
3. *Deciding whether a given planar, edge-colored, directed n-vertex graph has a perfect matching containing at least d colors.*

The previous fastest algorithms for these problems had runtimes $3.19^d \operatorname{poly}(n)$, $4^d \operatorname{poly}(n)$, and $4^d \operatorname{poly}(n)$, respectively [Bra]. This built upon work of Gutin et al. [GRWZ18] Problem (1) is the best studied among these, with [GRWZ18, Table 1] listing eleven articles on this problem in the last fourteen years. It is noteworthy that our improvements do not rely on any problem-specific adaptations.

Theorem 2 also yields a $\varphi^{2d}\operatorname{poly}(n)$-time deterministic algorithm for detecting simple cycles of length d in an n vertex directed graph (and paths, and more generally subgraphs of bounded treewidth). While it is known that simple cycles of length d can be detected in randomized time $2^d \operatorname{poly}(n)$ [Wil09] ($1.66^d \operatorname{poly}(n)$ for undirected graphs [Bjö10]), it is a major open problem to achieve the same runtime deterministically. Our algorithm brushes up against the fastest-known deterministic algorithm for this problem which has runtime $2.55^d \operatorname{poly}(n)$ [Tsu19], and unexpectedly matches the runtime of a previous algorithm [FLPS16] while using a different (shorter) approach. Our approach differs from those of previous algorithms which have been based on paradigms such as *color coding*, *divide and color*, and *representative families* [CFK$^+$15, Chapter 5]. Whereas these methods make use of explicit constructions of pseudorandom objects such as perfect hash families, universal sets, and representative sets, our algorithm makes use of algebraic-combinatorial identities. This approach was foreshadowed in [BDH18, Theorem 2]. It is important to note that our algorithm only works for unweighted graphs (or weighted graphs with integer weights bounded by $\operatorname{poly}(n)$), while several previous algorithms work for weighted graphs. The algorithm of [FLPS16] also extends more generally to detect subgraphs of bounded treewidth.

1.4. **Algebraic considerations.** As we note in Remark 2, our Theorems 1 and 2 can be viewed as algorithms for multiplication by degree-1 elements in the apolar algebras of the determinant and the generic Hankel determinant, respectively. Algorithms for *general* multiplication in these algebras, however, would have applications to problems such as detecting subgraphs of bounded *treewidth* (rather than just pathwidth). As a first step towards this, we consider the quantity $\mathbf{R}(\mathcal{A}_f)$, the bilinear complexity of $\mathcal{A}_f$. This is at most twice the number of non-scalar multiplications needed to multiply two elements in $\mathcal{A}_f$, and we show in

Proposition 5 how it bounds the number of multiplications that can be used to compute (1). We note in our Theorem 3 that $\mathbf{R}(\mathcal{A}_f)$ is, up to a linear factor, a lower bound on the Waring rank of f.

We show that for the degree-n determinant polynomial $\det_n$, $\mathbf{R}(\mathcal{A}_{\det_n}) \leq O(n2^{\omega n})$ where $\omega < 2.373$ is the exponent of matrix multiplication [LG14]. Our upper bound on $\mathbf{R}(\mathcal{A}_{\det_n})$ follows by realizing the apolar algebra of the determinant as a limit of a tensor product of Clifford algebras, which are classically known to be isomorphic to matrix algebras. We point out that if our upper bound on $\mathbf{R}(\mathcal{A}_{\det_n})$ is optimal, by Theorem 3 we would have that $\mathbf{R}_S(\det_n) \geq \Omega(2^{\omega n})$. For reference, the best known lower bounds on $\mathbf{R}_S(\det_n)$ are roughly 4^n. Therefore if known lower bounds on $\mathbf{R}_S(\det_n)$ *and* our upper bound on $\mathbf{R}(\mathcal{A}_{\det_n})$ are roughly optimal, $\omega = 2$. The Waring rank of the determinant and ω have been studied primarily in different contexts (see [AR19, Sha15, BT20, DT15] for work on the former), and have only started to be related [CHI$^+$18].

1.5. **Paper outline.** In the next section we prove Theorems 1 and 2. In Section 3 we give our applications. These follow quickly from Theorems 1 and 2, using little more than Cauchy-Binet. In Section 4 we then define the apolar algebra of a polynomial, and briefly discuss tensor rank and bilinear complexity. Using these we then define $\mathbf{R}(\mathcal{A}_f)$. We show in Example 3 how the fast subset convolution algorithm of [BHKK07] is equivalent to an algorithm for multiplication in $\mathcal{A}_{x_1x_2\cdots x_n}$, and how one can deduce from known tensor rank upper bounds improved upper bounds on the number of non-scalar multiplications needed to compute subset convolution. Finally we give our upper bound on $\mathbf{R}(\mathcal{A}_{\det_n})$.

2. Computing the apolar inner product for skew circuits

We start by giving an algorithm for computing (1) in the case that g is the determinant of a symbolic matrix and f is computed by a skew arithmetic circuit C. This is a warmup for the special case when g is the determinant of a symbolic Hankel matrix.

We fix the following notation for the rest of the paper. We denote by $|C|$ the total number of gates in the circuit C. Let $\mathbb{N}_d^k$ be the set of k-tuples with elements in $[d]$, and let $I(d,k) \subseteq \mathbb{N}_d^k$ be the set of strictly increasing sequences of length k with elements in $[d]$; when $k = 0$ we include the empty sequence in this set. Given a $d \times d$ matrix X and tuples $\alpha, \beta \in I(d,k)$, we denote by $X[\alpha|\beta]$ the minor (determinant of a submatrix) of X with rows indexed by α and columns indexed by β. We declare the "empty minor" $X[\,|\,]$ to equal one. We use the convention of writing $\alpha_1, \ldots, \hat{\alpha_i}, \ldots, \alpha_k$ to denote the sequence $\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k$ obtained from α by omitting α_i. We call a monomial $x_1^{a_1} \cdots x_n^{a_n}$ *square-free* if $a_i \in \{0,1\}$ for all i.

For $f \in \mathcal{S}_d^n$, $\mathrm{Diff}(f)$ denotes the vector space spanned by the partial derivatives of f of all orders (this includes f itself). For example, $\mathrm{Diff}(x_1x_2)$ is the vector space spanned by x_1x_2, x_1, x_2, and 1. The next observation is a simple bound on this quantity for determinants of symbolic matrices, and has been essentially observed several times previously (e.g. [Sha15, Lemma 1.3]).

Proposition 1. *Let $X = (\ell_{i,j})_{i,j\in[d]}$ be a symbolic matrix with entries in $\mathcal{S}_1^n$. Then $\mathrm{Diff}(\det X)$ is contained in the space of minors of X. Hence*

$$\dim \mathrm{Diff}(\det X) \leq \sum_{i=0}^{d} \binom{d}{i}^2 = \binom{2d}{d} < 4^d.$$

Proof. Let $\mathfrak{S}_d$ denote the symmetric group on d elements. By the Leibniz formula for the determinant and the product rule, for any $l \in [n]$,

$$\begin{aligned}\frac{\partial \det X}{\partial x_l} &= \sum_{\sigma\in\mathfrak{S}_d} \mathrm{sgn}(\sigma) \sum_{i=1}^{d} \frac{\partial \ell_{i,\sigma(i)}}{\partial x_l} \prod_{j\neq i} \ell_{j,\sigma(j)} = \sum_{1\leq i,j\leq d} \frac{\partial \ell_{i,j}}{\partial x_l} \sum_{\sigma\in\mathfrak{S}_d,\sigma(i)=j} \mathrm{sgn}(\sigma) \prod_{m\neq i} \ell_{m,\sigma(m)} \\ &= \sum_{1\leq i,j\leq d} (-1)^{i+j} \frac{\partial \ell_{i,j}}{\partial x_l} X[1,\ldots,\hat{i},\ldots,d|1,\ldots,\hat{j},\ldots,d].\end{aligned}$$

Note that $\frac{\partial \ell_{i,j}}{\partial x_l}$ is just a scalar. To see the last equality, consider the martix $X^{(ij)}$ obtained by setting the (i,j)th entry of X to 1, and all other entries in the ith row of X to zero. Then $\det X^{(ij)} = \sum_{\sigma \in \mathfrak{S}_d, \sigma(i)=j} \operatorname{sgn}(\sigma) \prod_{m \neq i} \ell_{m,\sigma(m)}$, but at the same time by Laplace expansion along the ith row of $X^{(ij)}$, $\det X^{(ij)} = (-1)^{i+j} X[1,\dots,\hat{i},\dots,d|1,\dots,\hat{j},\dots,d]$.

This shows that the space of order-1 partial derivatives of $\det X$ is contained in the span of the degree-$(d-1)$ minors of X. That $\operatorname{Diff}(\det X)$ is contained in the space of minors of X follows by repeated application of this fact. Furthermore, since square $k \times k$ submatrices of X can be identified by pairs of elements in $I(d,k)$ (their row and column indices), the vector space spanned by all minors of X has dimension at most $\sum_{k=0}^{d} |I(d,k)|^2 = \sum_{k=0}^{d} \binom{d}{k}^2 = \binom{2d}{d}$. □

Lemma 1. *Given as input a symbolic matrix $X = (\ell_{i,j})_{i,j\in[d]}$ with entries in $\mathcal{S}_1^n$, a linear combination P of minors of X, and $l \in [n]$, we can compute a representation for $\frac{\partial P}{\partial x_l}$ as a linear combination of minors of X with $4^d \operatorname{poly}(d)$ arithmetic operations.*

Proof. Let $P = \sum_{k=0}^{d} \sum_{\alpha,\beta \in I(d,k)} c_{\alpha,\beta} X[\alpha|\beta]$ and let $a_{i,j}^{(l)}$ be the coefficient of x_l in $\ell_{i,j}$ (so the input consists of l and the vectors $(c_{\alpha,\beta}) \in \mathbb{R}^{\binom{2d}{d}}$, $(a_{i,j}^{(k)}) \in \mathbb{R}^{d^2 n}$). Then by the same considerations as in the proof of Proposition 1,

$$\frac{\partial P}{\partial x_l} = \sum_{k=1}^{d} \sum_{\alpha,\beta \in I(d,k)} \sum_{1 \le i,j \le k} c_{\alpha,\beta} (-1)^{i+j} a_{i,j}^{(l)} X[\alpha_1,\dots,\hat{\alpha}_i,\dots,\alpha_k|\beta_1,\dots,\hat{\beta}_j,\dots,\beta_k].$$

Note that for $\alpha, \beta \in I(d,k)$, the coefficient of $X[\alpha|\beta]$ in the above equals

$$\sum_{1 \le i,j \le k} \sum_{\substack{\alpha',\beta' \in I(d,k+1) \\ \alpha = \alpha'_1,\dots,\hat{\alpha'_i},\dots,\alpha'_{k+1} \\ \beta = \beta'_1,\dots,\hat{\beta'_j},\dots,\beta'_{k+1}}} (-1)^{i+j} a_{i,j}^{(l)} c_{\alpha',\beta'}.$$

The numbers of pairs of sequences α', β' considered by the inner sum is naïvely bounded by d^4 (there are d positions in α where we could try to insert a number in $[d]$ into to get an increasing sequence, and similarly for β), and hence the coefficient of each minor can be computed with $O(d^6)$ arithmetic operations. Since there are $\binom{2d}{d}$ minors, all coefficients can be computed with the stated number of operations. □

Theorem 1. *Let C be a skew arithmetic circuit computing $g \in \mathcal{S}_d^n$, and let $X = (\ell_{i,j})_{i,j\in[d]}$ be a symbolic matrix with entries in $\mathcal{S}_1^n$. Then we can compute $\langle \det X, g \rangle$ with $4^d |C| \operatorname{poly}(d)$ arithmetic operations.*

Proof. Say that gate v in C computes the polynomial C_v. We will compute the inner product (1) inductively: at gate v we will compute and store C_v^∂, a representation for $C_v(\frac{\partial}{\partial x_1},\dots,\frac{\partial}{\partial x_n}) \det A$ as a linear combination of minors of X. C_v^∂ will be stored as a vector of length $\binom{2d}{d}$ indexed by pairs of row and column sets. At the end of the algorithm we will have computed $f(\frac{\partial}{\partial x_1},\dots,\frac{\partial}{\partial x_n}) \det X = \langle f, \det X \rangle$ at the output gate.

We start by computing and storing $\frac{\partial}{\partial x_l} \det X$ at input gate x_l, which by Lemma 1 can be done in $4^d \operatorname{poly}(d)$ time. Now suppose that gate v takes input from gates v' and v'', and that we have already computed $C_{v'}^\partial$ and $C_{v''}^\partial$. To compute C_v^∂, there are two cases to consider:

(1) $C_v = x_i \cdot C_{v'}$. Then $C_v^\partial = \frac{\partial}{\partial x_i} C_{v'}(\frac{\partial}{\partial x_1},\dots,\frac{\partial}{\partial x_n}) \det A = \frac{\partial}{\partial x_i} C_{v'}^\partial$. Using Lemma 1 this can be computed with $4^d \operatorname{poly}(d)$ operations.

(2) $C_v = C_{v'} + C_{v''}$. Since differentiation is linear, $C_v^\partial = C_{v'}^\partial + C_{v''}^\partial$. Since $C_{v'}^\partial$ and $C_{v''}^\partial$ are vectors of length $\binom{2d}{d}$, it takes $\binom{2d}{d}$ operations to add them.

Hence at each gate we use at most $4^d \operatorname{poly}(d)$ arithmetic operations, for a total of $4^d \operatorname{poly}(d)|C|$. □

We now show how Theorem 1 can be applied to obtain a deterministic algorithm for detecting simple cycles in graphs. This is not competitive, but it motivates our following improvement.

Proposition 2. *Let G be a graph on n vertices. We can decide in $4^d \operatorname{poly}(n)$ time if G contains a simple cycle of length d.*

Proof. Let $V \in \mathbb{Q}^{d\times n}$ be the Vandermonde matrix with $(V)_{i,j} = j^i$. Let $X = V \cdot \operatorname{diag}(x_1, \ldots, x_n) \cdot V^T$. By the Cauchy-Binet Theorem,

$$\det X = \sum_{\alpha\in I(n,d)} V[1, \ldots, d|\alpha]^2 \prod_{i\in S} x_i.$$

Since any d columns in V are linearly independent, $V[1, \ldots, d|\alpha]^2 > 0$ for all $\alpha \in I(n, d)$. Furthermore, observe that $tr(A_G^d)$ has nonnegative coefficients and contains a square-free monomial if and only if G contains a simple cycle of length d. It follows that $\langle \det A, tr(A_G^d)\rangle \neq 0$ if and only if G contains such a cycle. In addition, $tr(A_G^d)$ can be naïvely computed by a skew circuit of size $O(dn^3)$. The theorem follows by applying Theorem 1, noting that we only perform arithmetic with $\operatorname{poly}(n)$-bit integers. □

Note that the (i, j)th entry in the matrix X in the proof of Proposition 2 equals $\sum_{k=1}^n k^{i+j}x_k$, and therefore X is Hankel. We now show how this additional structure can be exploited.

Fix linear forms $\ell_1, \ldots, \ell_{2d-1} \in \mathcal{S}_1^n$, and let C_d be the symbolic matrix

$$\begin{bmatrix}
\ell_1 & \ell_2 & \ell_3 & \cdots & \cdots & \cdots & \ell_{2d-2} & \ell_{2d-1} \\
\ell_2 & \ell_3 & \cdots & \cdots & \cdots & \cdots & \ell_{2d-1} & 0 \\
\ell_3 & \cdots & \cdots & \cdots & \cdots & \ell_{2d-1} & 0 & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 & \vdots \\
\vdots & \vdots & \vdots & \vdots & \cdots & \cdots & \cdots & \vdots \\
\vdots & \vdots & \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\
\ell_{2d-2} & \ell_{2d-1} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
\ell_{2d-1} & 0 & 0 & \cdots & \cdots & \cdots & 0 & 0
\end{bmatrix} \tag{2}$$

The minors of the form $C_d[1, 2, \ldots, k|b_1, \ldots, b_k]$, where $k \leq d$ and $b_k \leq 2d - k$, are called *maximal*. For brevity we denote such a minor by $C_d[b_1, \ldots, b_k]$. Let H_d be the submatrix of C_d with row and column subscripts $1, \ldots, d$. It is readily seen that H_d is a Hankel matrix.

Proposition 3. *$\operatorname{Diff}(\det H_d)$ is contained in the space of maximal minors of C_d. Furthermore, the number of maximal minors of C_d is at most φ^{2d}.*

Proof. The maximal minors of C_d span the space of minors of H_d by Corollary 2.2(c) of [Con98]. Hence by Proposition 1, they span the space of partial derivatives of $\det H_d$. The second claim follows by noting that the number of maximal minors of degree k equals $|I(2d - k, k)| = \binom{2d-k}{k}$. Hence the total number of maximal minors equals $\sum_{k=0}^d \binom{2d-k}{k} < \varphi^{2d}$. In the last step we used the facts that the dth Fibonacci number satisfies $F_d = \sum_{k=0}^{\lfloor \frac{d-1}{2} \rfloor} \binom{d+k-1}{k}$, and that $F_d \leq \varphi^{d-1}$. □

Lemma 2. *Given as input a linear combination P of maximal minors of C_d and $l \in [n]$, we can compute a representation for $\frac{\partial P}{\partial x_l}$ as a linear combination of maximal minors of C_d with $\varphi^{2d} \operatorname{poly}(d)$ arithmetic operations.*

Proof. For brevity we will write $[\alpha]$ for the minor $C_d[\alpha]$. Let $P = \sum_{k=0}^d \sum_{\beta\in I(2d-k,k)} c_\beta[\beta]$, and say that the coefficient of x_l in $(C_d)_{i,j}$ is $a_{i,j}^{(l)}$. As in Lemma 1,

$$\frac{\partial P}{\partial x_l} = \sum_{k=1}^d \sum_{\beta\in I(2d-k,k)} c_\beta \sum_{1\leq i,j\leq k} (-1)^{i+\beta_j} a_{i,\beta_j}^{(l)} [1, \ldots, \hat{i}, \ldots, k|\beta_1, \ldots, \hat{\beta}_j, \ldots, \beta_k].$$

Note that the only minors with nonzero coefficient in this expression are of the form $[1,\dots,\hat{i},\dots,k|\gamma]$ for $k\in[d], i\in[k]$ and $\gamma\in I(2d-k,k-1)$. Call the coefficient of this minor in the above $b(i,\gamma)$. Then

$$b(i,\gamma)=\sum_{1\le j\le k}\sum_{\substack{\beta\in I(2d-k,k)\\ \gamma=(\beta_1,\dots,\hat{\beta_j},\dots,\beta_k)}} c_\beta(-1)^{i+\beta_j}a^{(l)}_{i,\beta_j}.$$

The number of sequences β considered by the inner sum is at most $O(d^2)$, and hence $b(i,\gamma)$ can be computed with $O(d^3)$ additions and multiplications. We can thus compute

$$\frac{\partial P}{\partial x_l}=\sum_{k=1}^{d}\sum_{i=1}^{k}\sum_{\gamma\in I(2d-k,k-1)} b(i,\gamma)[1,\dots,\hat{i},\dots,k|\gamma] \tag{3}$$

with $d^4\sum_{k=1}^{d}|I(2d-k,k-1)|\le\varphi^{2d}\operatorname{poly}(d)$ arithmetic operations. Note that this expresses $\frac{\partial P}{\partial x_l}$ as a linear combination of minors that are not necessarily maximal. We now fix this.

We first claim that for all $i\in[k]$ and $\beta\in I(2d-k,k-1)$,

$$[1,\dots,\hat{i},\dots,k|\beta]=\sum_{J\subseteq[k-1],|J|=k-i}[e(J)+(1,\dots,k-1)|\beta]$$

where $e(J)$ is the indicator vector of the set J. This holds since when $J=\{i,\dots,k-1\}$, $e(J)+(1,\dots,k-1)=(1,\dots,\hat{i},\dots,k)$, and for all other J, $e(J)+(1,\dots,k-1)$ will have a repeated value and hence $[e(J)+(1,\dots,k-1)|\beta]=0$.

Given this claim, it follows from [Con98, Lemma 2.1(a)] that

$$[1,\dots,\hat{i},\dots,k|\beta]=\sum_{J\subseteq[k-1],|J|=k-i}[\beta+e(J)],$$

and so letting Q_k be the degree-k part of Equation 3,

$$Q_k=\sum_{i=1}^{k+1}\sum_{\beta\in I(2d-k-1,k)} b(i,\beta)\sum_{J\subseteq[k],|J|=k+1-i}[\beta+e(J)].$$

We now show how to efficiently compute the coefficients of the maximal minors in this expression from the already computed $b(i,\gamma)$'s.

Let $0\le k\le d-1$ be fixed. For $\beta\in I(2d-k-1,k)$ and integers i,j where $0\le i\le j\le k$, let $D(\beta,i,j,k)\subseteq\{0,1\}^k$ be the set of binary vectors of length k containing exactly i ones, whose last $k-j$ entries are zero, and whose summation with β is strictly increasing everywhere except possibly at positions j and $j+1$ (that is, we may have $w_j+\beta_j=w_{j+1}+\beta_{j+1}$). Define

$$A^k(i,j)\coloneqq\sum_{\beta\in I(2d-k-1,k)} b(k+1-i,\beta)\sum_{w\in D(\beta,i,j,k)}[\beta+w].$$

Note that $\sum_{i=0}^{k}A^k(i,k)=Q_k$, so it suffices to show how to compute $A^k(i,j)$ for all i,j. We do this with a dynamic program. When we store $A^k(i,j)$ we will store all coefficients of maximal minors arising in the above definition, even though such a minor might contain a repeated column and hence equal zero. The minors arising in this definition are specified by sequences of length k with maximum value $2d-k$ that are strictly increasing everywhere but possibly at one position. Hence the number of such sequences is at most $k\binom{2d-k}{k}$.

For the base cases, we have

$$A^k(0,j) = \sum_{\beta \in I(2d-k-1,k)} b(k+1,\beta)[\beta],$$
$$A^k(i,i) = \sum_{\beta \in I(2d-k-1,k)} b(k+1-i,\beta)[\beta + e(\{1,\dots,i\})].$$

Now suppose we have computed $A^k(i,j-1)$ and $A^k(i-1,j-1)$. Then

$$\begin{aligned} A^k(i,j) &= \sum_{\beta \in I(2d-k-1,k)} b(k+1-i,\beta) \left(\sum_{\substack{w \in B(\beta,i,j,k),\\ w_j=0}} [\beta + w] + \sum_{\substack{w \in D(\beta,i,j,k),\\ w_j=1}} [\beta + w] \right) \\ &= \sum_{\beta \in I(2d-k-1,k)} b(k+1-i,\beta) \sum_{\substack{w \in D(\beta,i,j-1,k),\\ \beta+w \text{ is strictly increasing}}} [\beta + w] \\ &\quad + \sum_{\beta \in I(2d-k-1,k)} b(k+1-i,\beta) \sum_{w \in D(\beta,i-1,j-1,k)} [\beta + w + e(\{j\})]. \end{aligned}$$

The first part of the sum can be computed from $A^k(i,j-1)$ by setting the coefficient of any maximal minor with a repeated column equal zero, and the second sum can be computed from $A^k(i-1,j-1)$ by setting the coefficient of $[\beta]$ to that of $[\beta - e(\{j\})]$. Hence $A^k(i,j)$ can be computed with $O(k\binom{2d-k}{k})$ arithmetic operations. It follows that we can represent $\frac{\partial P}{\partial x_l} = \sum_{i=0}^{d-1} Q_i$ in the space of maximal minors using $\varphi^{2d}\operatorname{poly}(d)$ arithmetic operations. □

With this we have the following analog of Theorem 1. We omit the proof as it is almost exactly the same, we just work in the space of maximal minors rather than minors, using Lemma 2 to differentiate instead of Lemma 1.

Theorem 2. *Let C be a skew arithmetic circuit computing $g \in \mathcal{S}_d^n$, and let $X = (\ell_{i,j})_{i,j\in[d]}$ be a symbolic Hankel matrix with entries in $\mathcal{S}_1^n$. Then we can compute $\langle \det X, g\rangle$ with $\varphi^{2d}\operatorname{poly}(d)|C|$ arithmetic operations. Here $\varphi := \frac{1+\sqrt{5}}{2}$ is the golden ratio.*

Corollary 1. *Let G be a graph on n vertices. We can decide in $\varphi^{2d}\operatorname{poly}(n)$ time if G contains a simple cycle of length d.*

Proof. Let $V \in \mathbb{Q}^{d\times n}$ be the Vandermonde matrix with $(B)_{i,j} = j^i$, and $X = V \cdot \operatorname{diag}(x_1,\dots,x_n) \cdot V^T$. By the argument of 2, $\langle \det X, tr(A_G)^d\rangle \neq 0$ if and only if G contains a simple cycle of length d. Note that the (i,j)th entry in X equals $\sum_{k=1}^n k^{i+j}x_k$, and therefore X is Hankel. We conclude by applying Theorem 2 to compute $\langle \det X, tr(A_G)^d\rangle$, as $tr(A_G^d)$ can be computed by a skew circuit of size $\operatorname{poly}(n)$. □

Remark 1. This algorithm extends to detecting subgraphs of bounded pathwidth on d vertices by using the construction of the subgraph generating polynomial given in [BDH18, Appendix B].

3. Applications

In this section we give our applications of Theorems 1 and 2.

Corollary 2. *Given matrices $A_1,\dots,A_n \in \mathbb{Q}^{d\times d}$, we can decide if their span contains an invertible matrix in time $4^d\operatorname{poly}(N)$, where N denotes the size of the input.*

Proof. Let $X = \sum_{i=1}^n x_i A_i$. First note that $\operatorname{span}(A_1,\dots,A_n)$ contains an invertible matrix if and only if $\det X \not\equiv 0$. Writing $\det X = \sum_{\alpha\in\mathbb{N}_d^n} c_\alpha x^\alpha$ for some coefficients c_α (at least one of which will be nonzero

iff the answer is "yes"), observe that $\langle \det X, \det X \rangle = \sum_\alpha c_\alpha^2 \alpha!$. It follows that $\mathrm{span}(A_1, \ldots, A_n)$ contains an invertible matrix if and only if this quantity is nonzero.

It is shown in [MV97] that $\det_d$ can be expressed as a skew circuit of size $O(d^4)$, and the construction of this circuit is linear in the output size. Hence we can construct a circuit for $\det X$ by replacing the input variable x_{ij} in this circuit with the (i,j)th entry of X. The theorem follows by applying Theorem 1 to the matrix X and this circuit, noting that all numbers have bit-length $\mathrm{poly}(N)$ throughout the algorithm. □

Corollary 3. *Suppose we are given a matrix $A \in \mathbb{Q}^{km \times kn}$, where $n \geq m$, representing a matroid M with groundset $[kn]$, and a partition π of $[kn]$ into parts of size k. Then we can decide if the union of any m parts in π are independent in M in time $4^{km}\,\mathrm{poly}(N)$, where N is the size of the input.*

Proof. Let $g \coloneqq (\sum_{S \in \pi} \prod_{i \in S} x_i)^m$. It is easily seen that the squarefree monomials appearing in g correspond to unions of m elements in π, and that g can be computed by a skew circuit of size $\mathrm{poly}(n)$. Next, let $X = A \cdot \mathrm{diag}(x_1, \ldots, x_n) \cdot A^T$. By Cauchy-Binet,

$$\det X = \sum_{S \in \mathrm{Bases}(M)} \det(B_S)^2 \prod_{i \in S} x_i,$$

Note that the same monomial appears in the expansion of g and $\det X$ exactly when there is such an independent set in M, and then since g and $\det X$ have non-negative coefficients, $\langle \det X, g \rangle \neq 0$ if and only if an independent set in M is the union of m blocks in π. We conclude by applying Theorem 1. □

Using the same trick as in [Mar09] we can use Corollary 3 to solve the k-matroid intersection problem.

Corollary 4 (k-Matroid Intersection)**.** *Suppose we are given matrices $B_1, \ldots, B_k \in \mathbb{Q}^{m \times n}$ representing matroids $M_1, \ldots, M_k$ with the common groundset $[n]$. We can decide if $M_1, \ldots, M_k$ share a common base in time $4^{km}\,\mathrm{poly}(N)$, where N is the size of the input.*

Proof. Let $M = \bigoplus_{i=1}^k B_k$ be the direct sum of the input matrices. We first partition $[kn]$ into n parts of size k as follow: for $i \in [n]$, let $S_i \coloneqq \{i, i+n, i+2n, \ldots, i+kn\}$. If a union of m of the blocks $S_1, \ldots, S_n$ are independent in the matroid represented by M, $M_1, \ldots, M_k$ share a common base. Conversely, if these matroids share a common base, some union of the S_i's are independent in the matroid represented by M. We conclude by applying Corollary 3 to the matrix $M \in \mathbb{Q}^{km \times kn}$ and the partition $S_1, \ldots, S_n$. □

Finally, we have our applications of Theorem 2. These follow immediately by a reduction given in [Bra, Theorem 1] to the following "square-free monomial detection" algorithm.

Corollary 5. *Let $g \in \mathbb{Q}[x_1, \ldots, x_n]_d$ be a homogeneous degree-d polynomial with nonnegative coefficients, computed by a skew arithmetic circuit C. Given as input C, we can decide in deterministic $\varphi^{2d}|C|\,\mathrm{poly}(n)$ time whether g contains a degree-d square-free monomial.*

Proof. Let $V \in \mathbb{Q}^{d \times n}$ be the Vandermonde matrix with $(B)_{i,j} = j^i$, and $X = V \cdot \mathrm{diag}(x_1, \ldots, x_n) \cdot V^T$. By Cauchy-Binet,

$$\det X = \sum_{S \subseteq \binom{[n]}{d}} \det(B_S)^2 \prod_{i \in S} x_i.$$

Since any d columns in B are linearly independent, $\det(B_S)^2 > 0$ for all S. It follows that since g has nonnegative coefficients, $\langle \det X, g \rangle \neq 0$ if and only if g contains a square-free monomial. Note that the (i,j)th entry in X equals $\sum_{k=1}^n k^{i+j} x_k$, and therefore X is Hankel. The theorem follows by invoking Theorem 2. □

Applying [Bra, Theorem 1], we have:

Corollary 6. *The following problems admit deterministic algorithms running in time $\varphi^{2d}\,\mathrm{poly}(n)$:*

(1) *Deciding whether a given directed n-vertex graph has a directed spanning tree with at least d non-leaf vertices,*

(2) *Deciding whether a given edge-colored, directed n-vertex graph has a directed spanning tree containing at least d colors,*
(3) *Deciding whether a given planar, edge-colored, directed n-vertex graph has a perfect matching containing at least d colors.*

4. The Bilinear complexity of apolar algebras

In this section we study the complexity of multiplication in apolar algebras as a first step towards generalizing Theorems 1 and 2. We will work over $\mathbb{C}$ for convenience.

4.1. **Algebraic preliminaries.**

4.1.1. *Apolarity.* Let $\mathcal{R}^n := \mathbb{R}[\partial_1, \dots, \partial_n]$ be the ring of partial differential operators. Elements of this ring are just multivariate polynomials in the variables $\partial_1, \dots, \partial_n$. For an n-tuple $\alpha \in \mathbb{N}^n$, we let ∂^α be the monomial $\partial_1^{\alpha_1} \cdots \partial_n^{\alpha_n}$, and let $|\alpha| = \sum_{i=1}^n \alpha_i$. For $h \in \mathcal{R}$ and $f \in \mathcal{S}$, we denote by $h \circ f$ the result of applying the differential operator h to f. For example,

$$(3 \cdot \partial_1\partial_2 + \partial_1^2) \circ x_1^2 x_2 = 3 \cdot \partial_1\partial_2 \circ x_1^2 x_2 + \partial_1^2 \circ x_1^2 x_2 = 6x_1 + 2x_2.$$

It is clear that when h and f are homogeneous of the same degree, $h \circ f$ is a scalar. In this case $f(\partial_1, \dots, \partial_n) \circ g = \langle f, g \rangle$, so computing $h \circ f$ is equivalent to computing the apolar inner product.

Definition 1. For $f \in \mathcal{S}_d^n$, we define $\operatorname{Ann}(f)$ as the ideal of elements in $\mathcal{R}^n$ annihilating f under differentiation. We define the *apolar algebra* $\mathcal{A}_f$ as the quotient $\mathcal{R}^n / \operatorname{Ann}(f)$.

In other words, $\mathcal{A}_f$ is the ring of representatives of equivalence classes of differential operators subject to the equivalence relation $\sim$, where $h \sim h'$ if and only if $h \circ f = h' \circ f$. It follows that there is a vector space isomorphism $\mathcal{J}$ between $\mathcal{A}_f$ and $\operatorname{Diff}(f)$, sending $h \in \mathcal{A}_f$ to $h \circ f$. In particular, $(\mathcal{A}_f)_i \cong \operatorname{Diff}(f)_{d-i}$, where we denote by $(\mathcal{A}_f)_i$ the vector space of degree-i elements in $\mathcal{A}_f$.

Remark 2. Multiplication in $\mathcal{A}_f$ corresponds to differentiating by f: for $h_1, h_2 \in \mathcal{A}_f$, $\mathcal{J}(h_1 \cdot h_2) = h_1 \circ (h_2 \circ f)$. It follows that Lemmas 1 and 2 are algorithms for multiplication by ∂_l in $\mathcal{A}_{\det X}$, with respect to the spanning sets of $\mathcal{A}_{\det X}$ given by the inverse images of the minors (or maximal minors) of X.

Example 1. Let $f = x_1 x_2 \cdots x_n$. Note that for $1 \le i \le n$, $\partial_i^2 \circ f = 0$, and so $\partial_i^2 \in \operatorname{Ann}(f)$. Moreover, it is not hard to see that $\partial_1^2, \dots, \partial_n^2$ generate $\operatorname{Ann}(f)$. So the apolar algebra of f equals $\mathcal{A}_f = \mathbb{C}[\partial_1, \dots, \partial_n]/(\partial_1^2, \dots, \partial_n^2)$. This ring has as a basis the set of square-free monomials $\{\prod_{i \in S} \partial_i\}_{S \subseteq [n]}$, and the product of two basis elements is given by the rule

$$\partial_S \cdot \partial_T = \begin{cases} \partial_{S \cup T} & \text{if } S \cap T = \emptyset, \\ 0 & \text{else.} \end{cases}$$

4.1.2. *Bilinear complexity.* We give a brief primer on bilinear complexity. We refer to Chapter 14 of [BCS13] for an in-depth treathment of this topic.

Let U, V, W be finite dimensional complex vector spaces, and let $U \otimes V \otimes W$ be the vector space of three tensors. An element of $U \otimes V \otimes W$ of the form $u \otimes v \otimes w$ is called *simple*. The *rank* of a tensor $T \in U \otimes V \otimes W$, denoted $\mathbf{R}(T)$, is the smallest r such that T can be expressed as a sum of r simple tensors.

A $\mathbb{C}$-algebra $A = (V, \phi)$ is a complex vector space V with a multiplication operation defined by a bilinear map $\phi : V \times V \to V$. We say A is *associative* if $\phi(v_1, \phi(v_2, v_3)) = \phi(\phi(v_1, v_2), v_3)$ for all $v_1, v_2, v_3 \in V$, and *unital* if there is an element $e \in V$ such that $\phi(e, v) = \phi(v, e) = v$ for all $v \in V$. We will only be interested in unital associative algebras.

Let $\{e_1, \dots, e_n\}$ be a basis for V and $\{e_1^*, \dots, e_n^*\}$ be its dual basis. We can naturally identify A with its *structure tensor*

$$\sum_{i,j \in [n]} e_i \otimes e_j \otimes (e_i \cdot e_j) = \sum_{i,j,k \in [n]} e_k^*(\phi(e_i, e_j)) e_i \otimes e_j \otimes e_k \in V \otimes V \otimes V.$$

As an abuse of notation, we denote by $\mathbf{R}(A)$ the rank of the structure tensor of A. The algorithmic importance of this quantity is that its at most twice the minimum number of non-scalar multiplications needed to compute the product of two elements in A [BCS13, Equation 14.8]. Ranks of algebras are a classic topic in algebraic complexity (see [BCS13, Chapter 17], with the following being the most notorious example.

Example 2. Let $M_n = (\mathbb{C}^{n\times n}, \phi)$ be the algebra of $n \times n$ complex matrices, where ϕ is given by matrix multiplication. The vector space $\mathbb{C}^{n\times n}$ has as a basis the set of matrices $\{e_{ij}\}_{i,j\in[n]}$, where e_{ij} is the matrix whose (i, j)th entry equals one and all other entries equal zero. The multiplication of two basis elements is given by the rule $e_{ij} \cdot e_{kl} = e_{il}$ if $j = k$, and $e_{ij} \cdot e_{kl} = 0$ otherwise. Hence the structure tensor of A is $\langle n, n, n\rangle \coloneqq \sum_{i,j,k\in[n]} e_{ij} \otimes e_{jk} \otimes e_{ik}$, the *matrix multiplication tensor*. The exponent of matrix multiplication is defined as $\omega \coloneqq \inf_c\{\mathbf{R}(M_n) \leq O(n^c)\}$.

Example 3 (Fast subset convolution)**.** Let $f = x_1 \cdots x_n$. We claim that the problem of multiplying elements in $\mathcal{A}_f$ is exactly that of computing the *subset convolution*. Here the subset convolutions is defined for functions $\sigma, \tau : 2^{[n]} \to \mathbb{C}$ as the function $\sigma * \tau : 2^{[n]} \to \mathbb{C}$ such that

$$(\sigma * \tau)(S) = \sum_{U\subseteq S} \sigma(U)\tau(S - U).$$

The problem of computing $(\sigma * \tau)(S)$ for all S, given as input the 2^n values of τ and σ, has a handful of applications in exact algorithms [BHKK07, CFK$^+$15]. We now elaborate on the connection between subset convolution and $\mathcal{A}_f$.

Using the basis of square-free monomials as in Example 1, define the elements $a = \sum_{S\subseteq[n]} \sigma(S)\partial_S, b = \sum_{S\subseteq[n]} \tau(S)\partial_S$ of $\mathcal{A}_f$. Then by the equation for multiplication in $\mathcal{A}_f$ given in 1

$$a \cdot b = \sum_{S\subseteq[n]} (\sigma * \tau)(S)\partial_S,$$

so we can compute the subset convolution by computing $a \cdot b$ and reading off the coefficients of the result. It follows that the minimum number of non-scalar multiplications necessary to compute the subset convolution is at most $2\mathbf{R}(\mathcal{A}_f)$ (recalling [BCS13, Equation 14.8]). The structure tensor of $\mathcal{A}_f$ is

$$\sum_{S,T\subseteq[n]} \partial_S \otimes \partial_T \otimes (\partial_S \cdot \partial_T) = \sum_{S,T\subseteq[n],S\cap T=\emptyset} \partial_S \otimes \partial_T \otimes \partial_{S\cup T}.$$

This expression shows that the rank of this tensor is at most the number of pairs of disjoint subsets of $[n]$, which equals 3^n (each element in $[n]$ can be assigned to one of two subsets, or to none). In [BHKK07] an algorithm for computing subset convolution is given that uses just $O(\binom{n+2}{2}2^n)$ multiplications, thus showing that $\mathbf{R}(\mathcal{A}_f) \leq O(\binom{n+2}{2}2^n)$. In fact, one can say slightly more: the rank of this tensor has been studied in algebraic complexity, and it is known that $3 \cdot 2^n - o(2^n) \leq \mathbf{R}(\mathcal{A}_f) \leq (2n + 1)2^n$ [Zui17, Proposition 7,9].

Our next Theorem relates $\mathbf{R}(\mathcal{A}_f)$ to Waring rank.

Theorem 3. *Let $f \in \mathcal{S}^n_d$ and let $\mathcal{A}_f$ be its apolar algebra. Then*

$$\mathbf{R}(\mathcal{A}_f) \leq (3d + 1)\mathbf{R}_S(f).$$

Proof. Suppose that $f = \sum_{i=1}^r b_i\ell_i^d$, where $\ell_i = (a_{i,1}x_1 + \cdots + a_{i,n}x_n)$. Let B be a monomial basis for $\mathcal{A}_f$. Let

$$T \coloneqq \sum_{\partial^\alpha,\partial^\beta\in B} \partial^\alpha \otimes \partial^\beta \otimes (\partial^{\alpha+\beta} \circ f) \in \mathcal{A}_f \otimes \mathcal{A}_f \otimes \mathrm{Diff}(f).$$

First note that by the Apolarity lemma [IK99, Lemma 1.15(i)],

$$\partial^{\alpha+\beta} \circ f = \frac{d!}{(d-|\alpha|-|\beta|)!} \sum_{i=1}^{r} c_i a_{i,1}^{\alpha_1+\beta_1} \cdots a_{i,n}^{\alpha_n+\beta_n} \ell_i^{d-|\alpha|-|\beta|}$$

and hence for an indeterminate ε,

$$T\varepsilon^d + \sum_{\substack{0\le i\le 3d \\ i\neq d}} T_i \varepsilon^i = \sum_{i=1}^{r} \left(\sum_{\partial^\alpha \in B} \partial^\alpha a_{i,1}^{\alpha_1} \cdots a_{i,n}^{\alpha_n} \varepsilon^{|\alpha|} \right) \otimes \left(\sum_{\partial^\beta \in B} \partial^\beta a_{i,1}^{\beta_1} \cdots a_{i,n}^{\beta_n} \varepsilon^{|\beta|} \right) \otimes \left(\sum_{j=0}^{d} \frac{c_i d!}{(d-j)!} \ell_k^{d-j} \varepsilon^{d-j} \right)$$

since if $|\alpha|+|\beta|+(d-j)=d$, then $|\alpha|+|\beta|=j$. Here the T_i's are "junk" tensors we'd like to get rid of. We do this with an interpolation trick. Let $\{\varepsilon_i\}_{0\le i\le 3d}$ be elements of $\mathbb{C}$ that are distinct and nonzero, and let $\{c_i\}$ be the solution to the Vandermonde system

$$\sum_{i=0}^{3d} \varepsilon_i^j c_i = \begin{cases} 1, & j = d, \\ 0, & j \in \{0, \dots, d-1, d+1, \dots, 3d\}. \end{cases}$$

Then $\sum_{i=0}^{3d} c_i (T\varepsilon_i^d + \sum_{j\neq d} T_j \varepsilon_i^j) = T$, and hence $\mathbf{R}(T) \le (3d+1)\mathbf{R}_S(f)$.

Finally, we claim that T is isomorphic to the structure tensor of $\mathcal{A}_f$. This follows by applying the vector space isomorphism between $\mathrm{Diff}(f)$ and $\mathcal{A}_f$ sending $h \circ f$ to h, which sends $\partial^{\alpha+\beta} \circ f$ to $\partial^\alpha \partial^\beta$. □

We also have the following simple lower bound:

Proposition 4. $\mathbf{R}(\mathcal{A}_f) \ge \dim \mathcal{A}_f = \dim \mathrm{Diff}(f)$.

Proof. As $\mathcal{A}_f$ is unital, the Proposition follows (see e.g. [Zui17, Section 2.1]). □

The algorithmic relevance of $\mathbf{R}(\mathcal{A}_f)$ to the computing apolar inner product is given explicitly by the following proposition.

Proposition 5. *Fix $f \in \mathcal{S}_d^n$, and let $g \in \mathcal{S}_d^n$ be given as an arithmetic circuit C. Then we can compute $\langle f, g\rangle$ using $O(\mathbf{R}(\mathcal{A}_f)|C|)$ non-scalar multiplications.*

Proof. Let $(\mathcal{A}_f)_1$ have the basis $\partial_1, \dots, \partial_k$ for some $k \le n$, and let $(\mathcal{A}_f)_d$ have the basis q. Let $h = g(\partial_1, \dots, \partial_n)$. Then the result of evaluating h over $\mathcal{A}_f$ equals $h \bmod Ann(f) = \frac{\langle f,g\rangle q}{\langle f,q\rangle}$. So, our algorithm will evaluate C over $\mathcal{A}_f$, obtaining $c \cdot q$ for some $c \in \mathbb{C}$. We then return $c\langle f, q\rangle$. Note that $\langle f, q\rangle$ does not depend on the input g.

To evaluate h, we first replace the input gates x_i in C by zero if $i > k$, and ∂_i otherwise. We then evaluate C inductively over $\mathcal{A}_f$. At each gate we store an element of $\mathcal{A}_f$, which can be encoded by a vector of length $\dim \mathcal{A}_f$. At addition gates we simply sum the two inputs, which is done with $\dim \mathcal{A}_f \le \mathbf{R}(\mathcal{A}_f)$ additions, where the inequality follows by Proposition 5. Multiplication gates can be computed with at most $2\mathbf{R}(\mathcal{A}_f)$ non-scalar operations by [BCS13, Equation 14.8]. □

4.2. The bilinear complexity of $\mathcal{A}_{\det_n}$.

Theorem 4. $\mathbf{R}(\mathcal{A}_{\det_n}) \le O\left(n 2^{\omega n}\right)$.

Proof. We first give a basis for $\mathcal{A}_{\det_n}$ and describe how multiplication behaves with respect to this basis. This tells us what the structure tensor of the apolar algebra is. We then show how to obtain this tensor from $4n+1$ copies of $\langle 2^n, 2^n, 2^n\rangle$. We do this somewhat indirectly, using the fact that complex Clifford algebras are isomorphic to matrix algebras [Por81, Chapter 13]. We assume that n is even for ease of exposition.

We claim that the set of monomials of the form $(I|J) \coloneqq \partial_{I_1,J_1} \cdots \partial_{I_k,J_k}$, where $I, J \in I(n,k)$ and $0 \le k \le n$, are a basis for $\mathcal{A}_{\det_n}$. This follows from the fact that there are $\binom{2n}{n}$ such monomials, $\dim \mathrm{Diff}(\det_n) = \binom{2n}{n}$, and the polynomials of the form $(I|J) \circ \det_n$ are linearly independent. The latter claim can be seen

by noting that if $(I|J) \neq (I'|J')$, $(I|J) \circ \det_n$ and $(I'|J') \circ \det_n$ have disjoint sets of monomials appearing in their expansion.

Next we claim that the product of two basis elements $(I|J)$ and $(I'|J')$ is given by the rule

$$(I|J) \cdot (I'|J') = \begin{cases} 0 \text{ if } I \cap I' \neq \emptyset \text{ or } J \cap J' \neq \emptyset, \\ \operatorname{sgn}(I, I') \operatorname{sgn}(J, J')(I \cup I'|J \cup J') \text{ else} \end{cases}$$

where $\operatorname{sgn}(I, I')$ denotes the sign of the permutation that brings the sequence $I_1, \ldots, I_k, I'_1, \ldots, I_{k'}$ into increasing order, and $I \cup I'$ denotes the resulting sorted sequence. Indeed, if $I \cap I' \neq \emptyset$, then $(I|J)(I'|J')$ is divisible by the product of two variables that have the same first (row) index. But then $(I|J)(I'|J') \circ \det_n = 0$, since all monomials in the determinant have different row indices. The second case follows from the fact that for $I, J \in I(n, k)$ and $\tau \in \mathfrak{S}_k$, $(I|J) \circ \det_n = \operatorname{sgn} \tau^{-1} \cdot (\tau(I)|J) \circ \det_n$, which follows from the Leibniz formula for the determinant. Therefore the structure tensor of $\mathcal{A}_{\det_n}$ equals

$$T = \sum_{\substack{I,J,I',J' \subseteq [n] \\ |I|=|J|,|I'|=|J'| \\ I \cap I' = J \cap J' = \emptyset}} sgn(I, I') sgn(J, J')(I|J) \otimes (I'|J') \otimes (I \cup I'|J \cup J').$$

Let $CL_n = (V_n, \cdot)$ be the Clifford algebra of a nondegenerate quadratic form on $\mathbb{C}^n$ (see [Por81, Chapter 13] for background). Concretely, V_n has the basis X_U for $U \subseteq [n]$, and the product of two basis elements is given by the rule

$$X_U \cdot X_{U'} = \operatorname{sgn}(U, U') X_{U \Delta U'}$$

where Δ denotes the symmetric difference of sets. Here $\operatorname{sgn}(U, U')$ is the sign of the permutation that brings U, U' into nondecreasing order, leaving the relative order of any repeated elements unchanged. So the structure tensor of CL_n is

$$T_n \coloneqq \sum_{U,U' \subseteq [n]} \operatorname{sgn}(U, U') X_U \otimes X_{U'} \otimes X_{U \Delta U'}.$$

Since CL_n is isomorphic to the algebra of $2^{n/2} \times 2^{n/2}$ matrices [nLa20, Section 3], $\mathbf{R}(T_n) \leq O(2^{\omega n/2})$. Thus by submultiplicativity of tensor rank under the tensor product,

$$T_n \otimes T_n = \sum_{U,V,U',V' \subseteq [n]} sgn(U, U') sgn(V, V')(X_U \otimes X_V) \otimes (X_{U'} \otimes X_{V'}) \otimes (X_{U \Delta U'} \otimes X_{V \Delta V'})$$

has rank at most $O(2^{\omega n})$. Now define the linear transformations $M, M' : V_n \otimes V_n \to \mathcal{A}_{\det_n}(\varepsilon)$ given by $M(X_U \otimes X_V) = (U|V)\varepsilon^{|U|+|V|}$, and $M'(X_U \otimes X_V) = (U|V)\varepsilon^{-|U|-|V|}$, where ε is some indeterminate. Applying M to the first two factors of the above tensor and M' to the third factor,

$$(M, M, M') \cdot (T_n \otimes T_n) = T + \sum_{i=1}^{4n} \varepsilon^i H_i,$$

for some "junk" tensors H_i, since $|U| + |U'| = |U \Delta U'|$ if and only if U and U' are disjoint. Applying the interpolation trick as in Theorem 3, it follows that $\mathbf{R}(\mathcal{A}_{\det_n}) \leq O((4n + 1)2^{\omega n})$. □

Remark 3. In fact, the above proof shows that the border rank of the structural tensor of $\mathcal{A}_{\det_n}$ is at most $O(2^{\omega n})$.

5. Acknowledgments

We would like to thank several anonymous reviewers for their comments on an earlier draft of this paper.

References

[AG17] Nima Anari and Shayan Oveis Gharan. A generalization of permanent inequalities and applications in counting and optimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 384–396. ACM, 2017.

[AGR16] Nima Anari, Shayan Oveis Gharan, and Alireza Rezaei. Monte Carlo Markov chain algorithms for sampling strongly Rayleigh distributions and determinantal point processes. In *Conference on Learning Theory*, pages 103–115, 2016.

[AGV18] Nima Anari, Shayan Oveis Gharan, and Cynthia Vinzant. Log-concave polynomials, entropy, and a deterministic approximation algorithm for counting bases of matroids. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 35–46. IEEE, 2018.

[AHK18] Karim Adiprasito, June Huh, and Eric Katz. Hodge theory for combinatorial geometries. *Annals of Mathematics*, 188(2):381–452, 2018.

[AOGSS17] Nima Anari, Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. Nash social welfare, matrix permanent, and stable polynomials. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[AR02] Vikraman Arvind and Venkatesh Raman. Approximation algorithms for some parameterized counting problems. In *International Symposium on Algorithms and Computation*, pages 453–464. Springer, 2002.

[AR19] Jarod Alper and Rowan Rowlands. Syzygies of the apolar ideals of the determinant and permanent. *Journal of Algebraic Combinatorics*, pages 1–36, 2019.

[Bar95] Alexander I Barvinok. New algorithms for lineark-matroid intersection and matroid k-parity problems. *Mathematical Programming*, 69(1-3):449–470, 1995.

[Bar96] Alexander I Barvinok. Two algorithmic results for the traveling salesman problem. *Mathematics of Operations Research*, 21(1):65–84, 1996.

[BCS13] Peter Bürgisser, Michael Clausen, and Mohammad A Shokrollahi. *Algebraic complexity theory*, volume 315. Springer Science & Business Media, 2013.

[BDH18] Cornelius Brand, Holger Dell, and Thore Husfeldt. Extensor-coding. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 151–164, 2018.

[BHKK07] Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Fourier meets Möbius: fast subset convolution. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 67–74, 2007.

[BIP19] Peter Bürgisser, Christian Ikenmeyer, and Greta Panova. No occurrence obstructions in geometric complexity theory. *Journal of the American Mathematical Society*, 32(1):163–193, 2019.

[Bjö10] Andreas Björklund. Determinant sums for undirected hamiltonicity. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 173–182, 2010.

[Bra] Cornelius Brand. Patching colors with tensors. In *27th Annual European Symposium on Algorithms, ESA 2019, September 09-11, 2019, Munich, Germany*.

[BT20] Mats Boij and Zach Teitler. A bound for the Waring rank of the determinant via syzygies. *Linear Algebra and its Applications*, 587:195–214, 2020.

[CFK+15] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

[CHI+18] 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.

[Con98] Aldo Conca. Straightening law and powers of determinantal ideals of Hankel matrices. *Advances in Mathematics*, 138(2):263–292, 1998.

[DT15] Harm Derksen and Zach Teitler. Lower bound for ranks of invariant forms. *Journal of Pure and Applied Algebra*, 219(12):5429–5441, 2015.

[Edm67] Jack Edmonds. Systems of distinct representatives and linear algebra. *J. Res. Nat. Bur. Standards Sect. B 71*, (4):241–245, 1967.

[FG04] Jörg Flum and Martin Grohe. The parameterized complexity of counting problems. *SIAM Journal on Computing*, 33(4):892–922, 2004.

[FLPS16] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *J. ACM*, 63(4):29:1–29:60, 2016.

[GGOW19] Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. Operator scaling: theory and applications. *Foundations of Computational Mathematics*, pages 1–68, 2019.

[Gly13] David G Glynn. Permanent formulae from the Veronesean. *Designs, codes and cryptography*, 68(1-3):39–47, 2013.

[GRWZ18] Gregory Z. Gutin, Felix Reidl, Magnus Wahlström, and Meirav Zehavi. Designing deterministic polynomial-space algorithms by color-coding multivariate polynomials. *J. Comput. Syst. Sci.*, 95:69–85, 2018.

[Gur03] Leonid Gurvits. Classical deterministic complexity of Edmonds' problem and quantum entanglement. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 10–19, New York, NY, USA, 2003. ACM.

[Gur05] Leonid Gurvits. On the complexity of mixed discriminants and related problems. In *International Symposium on Mathematical Foundations of Computer Science*, pages 447–458. Springer, 2005.

[Gur06] Leonid Gurvits. Hyperbolic polynomials approach to Van der Waerden/Schrijver-Valiant like conjectures: sharper bounds, simpler proofs and algorithmic applications. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 417–426. ACM, 2006.

[Gur08] Leonid Gurvits. Van der waerden/schrijver-valiant like conjectures and stable (aka hyperbolic) homogeneous polynomials: one theorem for all. *The electronic journal of combinatorics*, 15(1):66, 2008.

[IK99] Anthony Iarrobino and Vassil Kanev. *Power sums, Gorenstein algebras, and determinantal loci*. Springer Science & Business Media, 1999.

[KI04] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *computational complexity*, 13(1-2):1–46, 2004.

[Lan12] Joseph M Landsberg. Tensors: geometry and applications. *Representation theory*, 381:402, 2012.

[Lee16] Hwangrae Lee. Power sum decompositions of elementary symmetric polynomials. *Linear Algebra and its Applications*, 492:89–97, 2016.

[LG14] François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th international symposium on symbolic and algebraic computation*, pages 296–303, 2014.

[Lov79] László Lovász. On determinants, matchings, and random algorithms. pages 565–574, 1979.

[Mac94] Francis Sowerby Macaulay. *The algebraic theory of modular systems*, volume 19. Cambridge University Press, 1994.

[Mar09] Dániel Marx. A parameterized view on matroid optimization problems. *Theor. Comput. Sci.*, 410(44):4471–4479, 2009.

[MV97] Meena Mahajan and V Vinay. A combinatorial algorithm for the determinant. In *In Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*. Citeseer, 1997.

[nLa20] nLab authors. Clifford algebra. `http://ncatlab.org/nlab/show/Clifford%20algebra`, April 2020. Revision 22.

[Por81] Ian R. Porteous. *Topological Geometry*. Cambridge University Press, 1981.

[Pra19] Kevin Pratt. Waring rank, parameterized and exact algorithms. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, MD, USA, November 9-12, 2019*, 2019.

[Sha15] Masoumeh Sepideh Shafiei. Apolarity for determinants and permanents of generic matrices. *Journal of Commutative Algebra*, 7(1):89–123, 2015.

[Syl52] J.J. Sylvester. On the principles of the calculus of forms. *Cambridge and Dublin Mathematical Journal*, 7:52–97, 1852.

[Tsu19] Dekel Tsur. Faster deterministic parameterized algorithm for k-Path. *Theoretical Computer Science*, 790:96–104, 2019.

[Wil09] Ryan Williams. Finding paths of length k in $O^*(2^k)$ time. *Inf. Process. Lett.*, 109(6):315–318, 2009.

[Zui17] Jeroen Zuiddam. A note on the gap between rank and border rank. *Linear Algebra and its Applications*, 525:33–44, 2017.