Parameterized Applications of Symbolic Differentiation of (Totally) Multilinear Polynomials

Cornelius Brand
Charles University, Prague, Czech Republic

Kevin Pratt
Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

We study the following problem and its applications: given a homogeneous degree-d polynomial g as an arithmetic circuit C, and a $d\times d$ matrix X whose entries are homogeneous linear polynomials, compute $g(\partial/\partial x_1, \ldots, \partial/\partial x_n) \det X$. We show that this quantity can be computed using $2^{\omega d}|C| \operatorname{poly}(n, d)$ arithmetic operations, where ω is the exponent of matrix multiplication. In the case that C is skew, we improve this to $4^d|C| \operatorname{poly}(n, d)$ operations, and if furthermore X is a Hankel matrix, to $\varphi^{2d}|C| \operatorname{poly}(n, d)$ operations, where $\varphi = \frac{1+\sqrt{5}}{2}$ is the golden ratio.

Using these observations we give faster parameterized algorithms for the matroid k-parity and k-matroid intersection problems for linear matroids, and faster deterministic algorithms for several problems, including the first deterministic polynomial time algorithm for testing if a linear space of matrices of logarithmic dimension contains an invertible matrix. We also match the runtime of the fastest deterministic algorithm for detecting subgraphs of bounded pathwidth with a new and simple approach. Our approach generalizes several previous methods in parameterized algorithms and can be seen as a relaxation of Waring rank based methods [Pratt, FOCS19].

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Parameterized Algorithms, Algebraic Algorithms, Longest Cycle, Matroid Parity

Digital Object Identifier 10.4230/LIPIcs.ICALP.2021.38

Category Track A: Algorithms, Complexity and Games

Funding *Cornelius Brand*: The research was supported by OP RDE project No. CZ.02.2.69/0.0/0.0/18_053/0016976 International mobility of research, technical and administrative staff at Charles University.

Acknowledgements We would like to thank Ryan O'Donnell and several anonymous reviewers for their many helpful comments on earlier drafts of this paper. In particular we thank an anonymous reviewer for suggesting the name "totally multilinear."

1 Introduction

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

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

Explicitly, if $f = \sum_{i_1,\ldots,i_n} a_{i_1,\ldots,i_n} x_1^{i_1} \cdots x_n^{i_n}$ and $g = \sum_{i_1,\ldots,i_n} b_{i_1,\ldots,i_n} x_1^{i_1} \cdots x_n^{i_n}$, then

$$\langle f, g \rangle = \sum_{i_1,\ldots,i_n} i_1! \cdots i_n! a_{i_1,\ldots,i_n} b_{i_1,\ldots,i_n}.$$

48th International Colloquium on Automata, Languages, and Programming (ICALP 2021).
Editors: Nikhil Bansal, Emanuela Merelli, and James Worrell; Article No. 38; pp. 38:1–38:19
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

This inner product originated in 19th century invariant theory [39] 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 can be $\#P$ hard to compute exactly, 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 [25], sampling from determinantal point processes [3], Nash social welfare maximization [5], and approximately counting subgraphs [37].

As a motivating example, given a directed graph G with n vertices, let A_G be the $n \times n$ 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,

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

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

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

(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 $\langle \det(A_S)^2 \prod_{i\in S} x_i, \operatorname{tr}(A_G^d)\rangle$ is positive if there is a simple cycle on the vertices $\{v_i : i \in S\}$, and zero otherwise. It follows by linearity 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 this and other applications, we consider in our Theorems 7, 13, and 25 the algorithmic task of computing the inner product (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. As one consequence, starting from the observation of the above example we 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 $\varphi := \frac{1+\sqrt{5}}{2}$ is the golden ratio. Our algorithm generalizes to detecting subgraphs of bounded pathwidth, unexpectedly matching the runtime of the fastest known algorithm for this problem of [20].

Our main conceptual contribution is the observation that the following algebraic question is central to a handful of methods in parameterized algorithms. It is motivated by the observation that in order to obtain a (possibly randomized) algorithm for detecting cycles, it would suffice to compute $\langle f, \operatorname{tr}(A_G^d)\rangle$ for any $f \in \mathcal{S}_d^n$ that is supported exactly on the set of all degree-d square-free monomials; $\det X$ is just one such polynomial. We call such polynomials *totally multilinear*, and denote the set of all such polynomials in $\mathcal{S}_d^n$ by $\mathcal{T}_{n,d}$.

► **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 $c_S \neq 0$ for all S. What is $B(d, n) := \min(\dim \operatorname{Diff}(f) : f \in \mathcal{T}_{n,d})$? Here $\operatorname{Diff}(f)$ denotes the vector space spanned by the partial derivatives of all orders of f, including f itself.* [1]

Our algorithms, color coding [1], the group algebra approach [31], and the exterior algebra methods of [12, 11] are all closely related to the existence of polynomials in $\mathcal{T}_{n,d}$ (or related sets) with "unusually small" spaces of partial derivatives (see Section 5). In [37] it was shown

[1] For example, $\operatorname{Diff}(x_1x_2)$ is the vector space spanned by x_1x_2, x_1, x_2, and 1.

that a related quantity, namely the minimum *Waring rank* of any $f \in \mathcal{T}_{n,d}$, gives upper bounds on the complexity of certain parameterized problems. In general, the Waring rank of $f \in \mathcal{S}_d^n$ is lower bounded by $\frac{1}{d} \dim \mathrm{Diff}(f)$, and this bound is almost never optimal [32, Section 3.2]. In this paper, we exploit that fact that, provided f can be "efficiently differentiated," this lower bound can be used to *upper bound* the complexity of these parameterized problems! For instance, our $\varphi^{2d}\,\mathrm{poly}(n)$-time cycle detection algorithm relies on the fact that there is a spanning set of size $\varphi^{2d} < 2.62^d$ for the space of partial derivatives of the polynomial $\det X$ above, when A is a Vandermonde matrix, that we can differentiate $\det X$ "efficiently" with respect to. In contrast, the best best-known upper bound on the Waring rank of this polynomial is 6.75^d [37, Theorem 41].

We show in Proposition 10 that $B(n,d) \le O(2.6^d)$. Additionally, it is not difficult to show that $B(n,d) \ge 2^d$. A proof of this fact is as follows. First, observe that $\dim \mathrm{Diff}(f)$ does not increase under setting variables to zero. Hence for any $f \in \mathcal{T}_{n,d}$, $\dim \mathrm{Diff}(f) \ge \dim \mathrm{Diff}(c \cdot x_1 x_2 \cdots x_d)$ for some nonzero constant c. As $\mathrm{Diff}(c \cdot x_1 x_2 \cdots x_d)$ has as a basis the collection of products of subsets of the variables $x_1, \ldots, x_d$, the claim follows.

1.1 Previous approaches to computing the apolar inner product

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. [27, 4]. In this case $\langle f, g\rangle$ can be approximated (up to a factor of $e^{d+\varepsilon}$) in polynomial time [2, Theorem 1.2]. For the cases we consider, however, f and g will not both be real stable.

Another approach is based on Waring rank upper bounds [9, 26, 23, 37]. The Waring rank of $f \in \mathcal{S}_d^n$, denoted $\mathbf{R}(f)$, 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 $\mathbf{R}(x_1x_2x_3) \le 4$. Waring rank has been studied since the 1850's [29, Introduction] and has gained recent attention for its applications to algebraic complexity, see e.g. [13, 14]. Its relevance to the inner product (1) is due to the following fact, which can be verified by a straightforward 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 $\mathbf{R}(f)$ yield black-box algorithms for computing $\langle f, g\rangle$. Furthermore, it was shown in [37, Theorem 6] that with only evaluation access to g, $\mathbf{R}(f)$ queries are *required* to compute this inner product. Unfortunately, $\mathbf{R}(f)$ is usually prohibitively large; for instance, the Waring rank of almost all $f \in \mathcal{S}_d^n$ is at least $\lceil \binom{n+d-1}{d}/n \rceil$ [32, Section 3.2].

It is also worth pointing out the very recent works of Arvind et al. [7, 6], in particular, [7, Remark 4.3.], which set up a very similar framework based on non-commutative polynomials and algebraic branching programs, on which they prove bounds that also follow from bounds on spaces of partial derivatives.

1.2 Our approach

Given g as an arithmetic circuit C, we compute $\langle f, g\rangle$ symbolically. Our algorithms inductively compute at each gate in C the result of differentiating f by the polynomial computed by C at that gate[2]. At the output gate of C we will therefore have computed $\langle g, f\rangle = \langle f, g\rangle$.

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

At intermediate gates we compute and store elements of Diff(f), the vector space of partial derivatives of f, which we represent with respect to some spanning set for this space. This kind of symbolic manipulation of partial derivatives is reminiscent of the Baur-Strassen Theorem and its applications (see for example [17]).

We will be particularly interested in the case when f is the determinant of a symbolic matrix X. The advantage of this case is that for a symbolic $d \times d$ matrix X, the vector space spanned by the partial derivatives of $\det X$ of all orders has dimension at most 4^d, and in some algorithmically relevant cases this bound can be significantly improved. 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 7 a simple 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 7.** *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 $4^d|C|\operatorname{poly}(n,d)$ arithmetic operations.*

Our algorithm for Theorem 7 only uses linear algebra and basic properties of differentials.

Of particular interest will be the case of Theorem 7 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

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

This has applications to problems such as detecting cycles in graphs and more generally detecting square-free monomials in arithmetic circuits (Corollary 19). We show the following improvement in this case:

▶ **Theorem 13.** *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 $\varphi^{2d}|C|\operatorname{poly}(n,d)$ arithmetic operations. Here $\varphi := \frac{1+\sqrt{5}}{2}$ is the golden ratio.*

The improvement in Theorem 13 over Theorem 7 is facilitated by the fact that the space of partial derivatives of the determinant has dimension about 4^d, whereas the dimension of the space of partial derivatives of the determinant of a Hankel matrix is upper bounded by φ^{2d} (Proposition 10).

Let us point out that Hankel matrices (in their guise as squares of the Vandermonde) made appearances already in the exterior-algebraic framework [12, 11], so that their usefulness in our applications might perhaps not come as a complete surprise. Before this work, however, any connections between the exterior and the partial-differential approach remained unclear, and their exact nature remains to be determined.

Still, we do gain one such connection here: For general (not necessarily skew) circuits, we exploit a connection between the *apolar algebra* of the determinant and the exterior algebra (Lemma 24) to show the following:

▶ **Theorem 25.** *Let C be an 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 $2^{\omega d}|C|\operatorname{poly}(n,d)$ arithmetic operations, where $\omega < 2.373$ is the exponent of matrix multiplication.*

1.3 Applications

Theorem 7 yields faster algorithms for the k-matroid intersection and matroid k-parity 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 17 and 18 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 [8], 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 in [36] 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. [20] and had runtime $2^{km\omega}\operatorname{poly}(N)$, where $\omega < 2.373$ is the exponent of matrix multiplication [33].

By combining Theorem 7 with a known construction of the determinant as a skew circuit [35], we obtain a faster deterministic algorithm for the following problem:

▶ **Problem 3** (SING). *Given matrices $A_1, \dots, A_n \in \mathbb{Q}^{d\times d}$, 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 16. In particular, this establishes that SING $\in \mathcal{P}$ for subspaces of matrices of logarithmic dimension. The fastest previous deterministic algorithm, due to an observation of Gurvits in [24], had runtime $2^d d!\operatorname{poly}(N)$ and made use of an upper bound of $2^d d!$ on $\mathbf{R}(\det_d)$. This problem was originally studied by Edmonds for its application to matching problems [18]. While it is known to admit a simple randomized polynomial time algorithm as was first observed by Lovász [34], a *deterministic* polynomial time algorithm would imply circuit lower bounds that seem far beyond current reach [30]. As a result, variants of SING have attracted attention, leading to a recent breakthrough in the non-commutative setting [22].

Using Theorem 13, we give in Corollary 19 a deterministic $\varphi^{2d}\operatorname{poly}(|C|)$-time algorithm for detecting square-free monomials of degree-d in a polynomial with non-negative coefficients computed by a skew arithmetic circuit. Combining this with observations in [11], we obtain the following applications:

▶ **Corollary 20.** *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 [11]. This built upon work of Gutin et al. [28] Problem (1) is the best studied among these, with [28, Table 1] listing eleven articles on this problem in the last fourteen years. It is worth noting that our improvements do not rely on any problem-specific adaptations.

Theorem 13 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 pathwidth). While it is known that simple cycles of length d can be detected in randomized time $2^d \operatorname{poly}(n)$ [41] ($1.66^d \operatorname{poly}(n)$ for undirected graphs [10]), it is a major open problem to achieve the same runtime deterministically. Finding a better upper bound on $B(n, d)$ witnessed by a polynomial with nonnegative coefficients seems to us a promising approach for obtaining faster deterministic algorithms for this problem.

Our cycle detection algorithm brushes up against the fastest known deterministic algorithm for this problem which has runtime $2.55^d \operatorname{poly}(n)$ [40], and unexpectedly matches the runtime of a previous algorithm [20] 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* [16, Chapter 5] [43]. 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 [12, 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 [20] also extends more generally to detect subgraphs of bounded treewidth.

1.4 Algebraic considerations; the potential for improvement

In Section 4 we note that our algorithms for computing special cases of (1) yield algorithms for performing arithmetic in a certain algebra $\mathcal{A}_f$ associated to f, namely the *apolar algebra* of f. We show in our Lemma 24 that the apolar algebra of the determinant is isomorphic to the diagonal subalgebra of the tensor square of the exterior algebra. This algebra was previously identified in [12] for its applications to detecting subgraphs of bounded pathwidth. By combining this observation with known algorithms for arithmetic in the exterior algebra, we derive our general algorithm for computing $\langle \det X, g \rangle$.

To obtain faster deterministic algorithms for several problems such as detecting simple cycles, we ask Question 1. Our Corollary 11 shows that $B(d, n)$ is at most $(\sqrt{27}/2)^d d$. This upper bound is witnessed by the polynomial

$$\sum_{1 \le i_1 < \cdots < i_d \le n} \prod_{j<k} (i_j - i_k)^2 \prod_{j=1}^{d} x_{i_j}.$$

► Remark 4. The discrepancy between $(\sqrt{27}/2)^d d \le O(2.6^d)$ and the base of the exponent $\varphi^{2d} > 2.6^d$ in our runtimes is due to the fact that we do not know how to differentiate in nearly-linear time with respect to a basis for the above polynomial; see Further Questions (6). Instead, we compute them with respect to a larger spanning set.

1.5 Paper outline

In the next section we prove Theorems 7 and 13. We will motivate them with the running application of detecting simple cycles, giving in Corollary 14 our $\varphi^{2d} \operatorname{poly}(n)$-time algorithm. The rest of our applications can be found in Section 3, and follow quickly from Theorems 7 and 13, using little more than the Cauchy-Binet formula. In Section 4 we define the apolar algebra of a polynomial. We relate the apolar algebra of the determinant to the tensor square of the exterior algebra in Lemma 24, and use this to prove Theorem 25. We then explain the connection to other methods in parameterized algorithms in Section 5.

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 $I(d,k) \subseteq [d]^k$ 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 $\alpha_1, \dots, \widehat{\alpha}_i, \dots, \alpha_k$ to denote the sequence $\alpha_1, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_k$ 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. [38, Lemma 1.3]).

▶ **Proposition 5.** *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*

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

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 $l \in [n]$,

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

Note that $\frac{\partial \ell_{i,j}}{\partial x_l}$ is just a scalar. To see the last equality, consider the matrix $X^{(ij)}$ obtained by setting the (i,j)th entry of X to 1, and all other entries in the ith row of X to 0. Then $\det X^{(ij)} = \sum_{\sigma\in\mathfrak{S}_d,\sigma(i)=j} \mathrm{sgn}(\sigma) \prod_{m\neq i} \ell_{m,\sigma(m)}$, but at the same time by Laplace expansion along the ith row of $X^{(ij)}$, $\det X^{(ij)} = (-1)^{i+j} X[1,\dots,\widehat{i},\dots,d|1,\dots,\widehat{j},\dots,d]$.

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 $\mathrm{Diff}(\det X)$ 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 6.** *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 $4^d \operatorname{poly}(n,d)$ 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 $(c_{\alpha,\beta}) \in \mathbb{Q}^{\binom{2d}{d}}$, $(a_{i,j}^{(l)}) \in \mathbb{Q}^{d^2 n}$). Then by the same considerations as in the proof of Proposition 5,

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

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

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

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 7.** *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 $4^d|C| \operatorname{poly}(n,d)$ 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 $C_v(\frac{\partial}{\partial x_1},\ldots,\frac{\partial}{\partial x_n})\det X$ 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 $f(\frac{\partial}{\partial x_1},\ldots,\frac{\partial}{\partial x_n})\det X$ (which by symmetry of the apolar inner product equals $\langle \det X, f\rangle$) 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 6 can be done in $4^d \operatorname{poly}(n,d)$ 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 $C_v^{\partial} = \frac{\partial}{\partial x_i} C_{v'}(\frac{\partial}{\partial x_1},\ldots,\frac{\partial}{\partial x_n})\det X = \frac{\partial}{\partial x_i} C_{v'}^{\partial}$. Using Lemma 6 this can be computed with $4^d \operatorname{poly}(n,d)$ 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 $4^d \operatorname{poly}(n,d)$ arithmetic operations, for a total of $4^d \operatorname{poly}(n,d)|C|$. ◀

We now show how Theorem 7 can be applied to obtain a deterministic algorithm for detecting simple cycles in graphs. This motivates the following improvement.

▶ **Proposition 8.** *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 $X = V \cdot \mathrm{diag}(x_1,\ldots,x_n)\cdot V^{\mathrm{T}}$. By the Cauchy-Binet formula,

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

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 $\mathrm{tr}(A_G^d)$ has nonnegative coefficients and contains a square-free monomial if and only if G contains a simple cycle of length d. It follows that $\langle \det X, \mathrm{tr}(A_G^d)\rangle \neq 0$ if and only if G contains such a cycle. In addition, $\mathrm{tr}(A_G^d)$ can be naïvely computed by a skew circuit of size $O(dn^3)$. The theorem follows by applying Theorem 7, noting that we only perform arithmetic with $\mathrm{poly}(n)$-bit integers. ◀

Note that the (i,j)th entry in the matrix X in the proof of Proposition 8 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

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

The minors of the form $C_d[1,2,\ldots,k|b_1,\ldots,b_k]$, where $k \le d$ and $b_k \le 2d-k$, are called *maximal*. For brevity we will let $[\alpha|\beta] := C_d[\alpha|\beta]$, and if $[\alpha|\beta]$ is maximal (so $\alpha = 1,\ldots,k$) we further simplify this to $[\beta]$. 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.

We will need the following fact of Conca [15, Lemma 2.1(a)]. For a subset I, we let $e(I)$ be its indicator vector.

▶ **Lemma 9.** *Let $\alpha = \alpha_1,\ldots,\alpha_t$ and $\beta = \beta_1,\ldots,\beta_t$ be sequences of positive integers. Then for all $k = 1,\ldots,t$,*

$$\sum_{I\subseteq[t],|I|=k} [\alpha+e(I)|\beta] = \sum_{J\subseteq[t],|J|=k} [\alpha|\beta+e(J)].$$

Proof. We denote by α_I the subsequence of α indexed by the set I, and by $\alpha^{\hat{I}}$ the subsequence indexed by the complement of I in $[t]$. We let $\alpha+1 = \alpha_1+1,\ldots,\alpha_t+1$.

First, expanding $[\alpha+e(I)|\beta]$ with respect to the rows indexed by α_I+1:

$$\sum_I [\alpha+e(I)|\beta] = \sum_I\sum_J (-1)^{|I|}(-1)^{|J|}[\alpha_I+1|\beta_J][\alpha^{\hat{I}}|\beta^{\hat{J}}].$$

Since C_d is Hankel, $[\alpha_I+1|\beta_J] = [\alpha_I|\beta_J+1]$. So

$$\sum_I [\alpha+e(I)|\beta] = \sum_J\sum_I (-1)^{|I|}(-1)^{|J|}[\alpha_I|\beta_J+1][\alpha^{\hat{I}}|\beta^{\hat{J}}] = \sum_J [\alpha|\beta+e(J)]$$

where in the final equality we recognize that the middle equation equals $[\alpha|\beta+e(J)]$ expanded with respect to the columns indexed by β_J+1. ◀

▶ **Corollary 10.** $\mathrm{Diff}(\det H_d)$ *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. By Proposition 5, the space of partial derivatives of $\det H_d$ is spanned by the minors of C_d. We now show that the maximal minors of C_d span the space of minors of H_d. We will follow the proof of Corollary 2.2 in [15].

Let $[\alpha_1, \ldots, \alpha_t | \beta_1, \ldots, \beta_t]$ be a minor of C_d, where $t \le d$ and $\alpha_1 = 1$. Note that any minor of H_d can be expressed in this form by shifting the corresponding submatrix in C_d up and to the right. We also assume α and β are strictly increasing sequences (if this is not the case then $[\alpha|\beta]$ vanishes). We now give an inductive procedure that expresses $[\alpha|\beta]$ as a linear combination of maximal minors.

If $\alpha_t = t$, then this minor is maximal and we are done. Otherwise, let h be the smallest index where $\alpha_h > h$. We now apply Lemma 9 to the minor $[\alpha_1, \ldots, \alpha_{h-1}, \alpha_h - 1, \ldots, \alpha_t - 1|\beta]$ with $k = t - h + 1$. Doing so we obtain an expression for $[\alpha|\beta]$ as a linear combination of minors of C_d, each of which in turn have a larger "h." We conclude by induction.

Finally, note that the number of maximal minors of degree k is $|I(2d-k,k)| = \binom{2d-k}{k}$. 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 $F_d \le \varphi^{d-1}$. ◀

The next observation was noted in [37, Theorem 43].

▶ **Corollary 11.**

$$\dim \mathrm{Diff}(\det H_d) \le \sum_{i=0}^{d} \min\left\{ \binom{d+i}{d-i}, \binom{2d-i}{i} \right\} \le (\sqrt{27}/2)^d \cdot d.$$

Proof. By Corollary 10, for $i \le d/2$ the degree-i piece of $\mathrm{Diff}(H_d)$ has dimension at most $\binom{2d-i}{i}$. We conclude by the fact that $\mathrm{Diff}(H_d)_i \cong \mathrm{Diff}(H_d)_{d-i}$, i.e., the sequence of dimensions of space of partial derivatives is symmetric about $d/2$ [29, Definition 1.9]. The inequality on the right follows from Stirling's formula. ◀

▶ **Lemma 12.** *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* $\varphi^{2d}\,\mathrm{poly}(n,d)$ *arithmetic operations.*

Proof. For brevity we will write $[\alpha]$ for the minor $C_d[1, \ldots, |\alpha||\alpha]$. 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 6,

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

Note that the only minors with nonzero coefficient in this expression are of the form $[1, \ldots, \widehat{i}, \ldots, k|\gamma]$ 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

$$b(i,\gamma) = \sum_{1 \le j \le k} \sum_{\substack{\beta \in I(2d-k,k) \\ \gamma = (\beta_1, \ldots, \widehat{\beta}_j, \ldots, \beta_k)}} c_\beta (-1)^{i+\beta_j} a_{i,\beta_j}^{(l)}.$$

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

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

with $d^4\sum_{k=1}^{d}|I(2d-k,k-1)| \le \varphi^{2d}\,\mathrm{poly}(n,d)$ 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 $\beta\in I(2d-k,k-1)$,

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

where $e(J)$ is the indicator vector of the set J. This holds since when $J=\{i,\dots,k-1\}$, $e(J)+(1,\dots,k-1)=(1,\dots,\widehat{i},\dots,k)$, 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 Lemma 9 that

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

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

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

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

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

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 $\varphi^{2d}\operatorname{poly}(n, d)$ arithmetic operations. ◀

With this we have the following analog of Theorem 7. 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 12 to differentiate instead of Lemma 6.

▶ **Theorem 13.** *Let C be a skew arithmetic circuit computing $g \in \mathcal{S}^n_d$, and let $X = (\ell_{i,j})_{i,j\in[d]}$ be a symbolic Hankel matrix with entries in $\mathcal{S}^n_1$. Then we can compute $\langle \det X, g\rangle$ with $\varphi^{2d}|C|\operatorname{poly}(n, d)$ arithmetic operations. Here $\varphi := \frac{1+\sqrt{5}}{2}$ is the golden ratio.*

▶ **Corollary 14.** *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 $V_{i,j} = j^i$, and $X = V \cdot \operatorname{diag}(x_1, \dots, x_n) \cdot V^{\mathrm{T}}$. By the same argument of Proposition 8, $\langle \det X, \operatorname{tr}(A_G^d)\rangle \neq 0$ 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 13 to compute $\langle \det X, \operatorname{tr}(A_G)^d\rangle$, as $\operatorname{tr}(A_G^d)$ can be computed by a skew circuit of size $\operatorname{poly}(n)$. ◀

▶ Remark 15. This algorithm extends to detecting subgraphs of bounded pathwidth on d vertices by using the construction of the subgraph generating polynomial given in [12, Appendix B].

3 Applications

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

▶ **Corollary 16.** *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 $X = \sum_{i=1}^n x_iA_i$. First note that $\operatorname{span}(A_1, \dots, A_n)$ contains an invertible matrix if and only if $\det X \not\equiv 0$. Writing $\det X = \sum_{\alpha\in[d]^n} c_\alpha x^\alpha$ 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 $\operatorname{span}(A_1, \dots, A_n)$ contains an invertible matrix if and only if this quantity is nonzero.

It is shown in [35] 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 7 to the matrix X and this circuit, noting that all numbers have bit-length $\mathrm{poly}(N)$ throughout the algorithm. ◀

▶ **Corollary 17.** *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.* [3]

Proof. Let $g := (\sum_{S\in\pi} \prod_{i\in S} x_i)^m$. It is easily seen that the square-free 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 $X = A \cdot \mathrm{diag}(x_1, \dots, x_n) \cdot A^{\mathrm{T}}$. By the Cauchy-Binet formula,

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

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 7. ◀

Using the same trick as in [36] we can use Corollary 17 to solve the k-matroid intersection problem.

▶ **Corollary 18** (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 $[n]$. We can decide if $M_1, \dots, M_k$ 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 follows: for $i \in [n]$, let $S_i := \{i, i + n, i + 2n, \dots, i + kn\}$. If a union of m of the blocks $S_1, \dots, S_n$ are independent in the matroid represented by M, then $M_1, \dots, M_k$ 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 17 to the matrix $M \in \mathbb{Q}^{km\times kn}$ and the partition $S_1, \dots, S_n$. ◀

Finally, we have our applications of Theorem 13. These follow immediately by a reduction given in [11, Theorem 1] to the following "square-free monomial detection" algorithm.

▶ **Corollary 19.** *Let $g \in \mathbb{Q}[x_1, \dots, x_n]_d$ 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 $V_{i,j} = j^i$, and $X = V \cdot \mathrm{diag}(x_1, \dots, x_n) \cdot V^{\mathrm{T}}$. By the Cauchy-Binet formula,

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

[3] Similar to the Theorem 1.1 of [36], this algorithm can be modified to work for finite fields of sufficiently large size with the addition of randomness.

Since any d columns in V are linearly independent, $\det(V_S)^2 > 0$ 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 13. ◀

By [11, Theorem 1], we then have:

▶ **Corollary 20.** *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 A general algorithm: the apolar algebra of the determinant

4.1 Algebraic preliminaries

Let $\mathcal{R}^n := \mathbb{Q}[\partial_1, \ldots, \partial_n]$ be the ring of partial differential operators. Elements of this ring are just multivariate polynomials in the variables $\partial_1, \ldots, \partial_n$. 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,

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

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

▶ **Definition 21.** *For $f \in \mathcal{S}_d^n$, we define $\mathrm{Ann}(f)$ as the ideal of elements in $\mathcal{R}^n$ annihilating f under differentiation. We define the* apolar algebra $\mathcal{A}_f$ *as the quotient $\mathcal{R}^n / \mathrm{Ann}(f)$.*

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 $\mathrm{Diff}(f)$, sending $h \in \mathcal{A}_f$ to $h \circ f$. In particular, $(\mathcal{A}_f)_i \cong \mathrm{Diff}(f)_{d-i}$, where we denote by $(\mathcal{A}_f)_i$ the vector space of degree-i elements in $\mathcal{A}_f$.

▶ Remark 22. 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 6 and 12 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.

▶ **Definition 23.** *$\Lambda(\mathbb{Q}^n) \otimes \Lambda(\mathbb{Q}^n)$ is the algebra with the basis of formal variables $\{(I|J) : I, J \subseteq [n]\}$, and where multiplication is given by extending bilinearly the rule*

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

where $\mathrm{sgn}(I, I') = (-1)^{|\{i \in I, i' \in I' : i > i'\}|}$.

▶ **Lemma 24.** *$\mathcal{A}_{\det_n}$ is isomorphic to the subalgebra of $\Lambda(\mathbb{Q}^n) \otimes \Lambda(\mathbb{Q}^n)$ generated by $\{v \otimes v : v \in \Lambda(\mathbb{Q}^n)\}$.*

Proof. We first 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

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

where $\mathrm{sgn}(I, I')$ denotes the sign of the permutation that brings the sequence $I_1, \dots, I_{k'}$ 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 $\tau \in \mathfrak{S}_k$, $(I|J) \circ \det_n = \mathrm{sgn}\, \tau^{-1} \cdot (\tau(I)|J) \circ \det_n$, which follows from the Leibniz formula for the determinant.

It follows from these observations that $\mathcal{A}_{\det_n}$ is the claimed subalgebra of $\Lambda(\mathbb{Q}^n) \otimes \Lambda(\mathbb{Q}^n)$. ◀

▶ **Theorem 25.** *Let C be an arithmetic circuit computing $g \in \mathcal{S}^n_d$, and let $X = (\ell_{i,j})_{i,j\in[d]}$ be a symbolic matrix with entries in $\mathcal{S}^n_1$. Then we can compute $\langle \det X, g\rangle$ with $2^{\omega d}|C| \,\mathrm{poly}(n, d)$ arithmetic operations, where $\omega < 2.373$ is the exponent of matrix multiplication.*

Proof. Assume that the entries of X are linearly independent. If this is not the case, add a new variable $x_{i,j}$ to the (i, j)th entry of X. Note that since these variables do not appear in g, this does not change the value of $\langle \det X, g\rangle$.

Let $A : \mathcal{S}^n_1 \to \mathbb{Q}[y_{1,1}, \dots, y_{d,d}]_1$ be the linear transformation sending the linear form $X_{i,j}$ to the variable $y_{i,j}$. By [19, Corollary 3.1], $\langle \det X, g\rangle = \langle \det Y, g((A^{-1})^T(x_1, \dots, x_n))\rangle$, where $Y_{i,j} = y_{i,j}$. So we will first modify C by applying the linear transformation $(A^{-1})^T$ to the input gates, obtaining a new circuit C'. We then will evaluate C' over $\mathcal{A}_{\det Y}$, using the monomial basis of Lemma 24. Additions in $\mathcal{A}_{\det Y}$ can be done in time linear in the number of basis elements, which is bounded above by 4^d. By identifying $\mathcal{A}_{\det_d}$ with the subalgebra of diagonal elements in $\Lambda(\mathbb{Q}^d) \otimes \Lambda(\mathbb{Q}^d)$, and using the $2^{\omega d}\,\mathrm{poly}(d)$-time algorithm of [42, Theorem 14] for multiplying elements in $\Lambda(\mathbb{Q}^d) \otimes \Lambda(\mathbb{Q}^d)$, we can multiply elements in $\mathcal{A}_{\det_d}$ with $2^{\omega d}\,\mathrm{poly}(d)$ operations. Note that the highest degree element in this basis is $q \coloneqq \partial_{1,1}\partial_{2,2} \cdots \partial_{d,d}$. The output gate of of C' therefore equals $(A^{-1})^T \cdot g \bmod \mathrm{Ann}(\det Y) = \frac{\langle \det X, g\rangle q}{\langle \det Y, q\rangle} = \langle \det X, g\rangle q$. ◀

5 Totally Multilinear Polynomials and Previous Methods

We now explain how previous algorithms relate to answers to Question 1. Recall for comparison that Proposition 10 implies that $B(n, d) \le O(2.6^d)$, and is witnessed by the polynomial

$$f = \sum_{1\le i_1<\cdots<i_d\le n} \prod_{j<k} (i_j - i_k)^2 \prod_{j=1}^{d} x_{i_j}.$$

Furthermore, we showed in Lemma 12 that given an element of $\mathrm{Diff}(f)$, the linear map $\partial_i : \mathrm{Diff}(f) \to \mathrm{Diff}(f)$ could be computed in time $\varphi^{2d}\,\mathrm{poly}(n)$ with respect to the spanning set of maximal minors.

5.1 Color coding

Let $\mathcal{F}$ be an (n, d) *perfect hash family*, that is, a family of functions from $[n]$ to $[d]$ such that for any subset of $[n]$ of size d, some function in $\mathcal{F}$ is injective on $[d]$. It can be shown by a straightforward random argument that there exists such an $\mathcal{F}$ with size at most $e^d\,\mathrm{poly}(n)$ [1].

Now for $\pi \in \mathcal{F}$ and $i \in [d]$, define the linear forms $\ell_{\pi,i} = \sum_{j\in\pi^{-1}(i)} x_j$. Let $f = \sum_{\pi\in\mathcal{F}} \prod_{i=1}^{d} \ell_{\pi,i}$. Then from the definition of a perfect hash family it follows that $f \in \mathcal{T}_{n,d}$. As the space of partial derivatives of a product of d linear forms has dimension at most 2^d, $\dim \mathrm{Diff}(f) \le |\mathcal{F}|2^d \le (2e)^d\,\mathrm{poly}(n)$.

Explicitly, $\mathrm{Diff}(f)$ is spanned by $\prod_{i\in S} \ell_{\pi,i}$ for all $S \subseteq [d]$ and $\pi \in \mathcal{F}$. In addition, the operator ∂_j can efficiently be computed with respect to this spanning set; this follows from the fact that $\partial_j \prod_{i\in S} \ell_{\pi,i} = \prod_{i\in S: j\notin\pi^{-1}(i)} \ell_{\pi,i}$ (i.e., the matrix representing ∂_i is sparse). Hence color coding can be interpreted in our framework as using the polynomial f.

5.2 Waring rank

In [37] an improvement to the color-coding construction was given. Let $\mathcal{F}$ be an $(n, d, 1.55d)$-*splitter*, that is, a family of functions from $[n]$ to $[1.55d]$ such that for any subset S of $[1.55d]$ of size d, there exists some $\pi \in \mathcal{F}$ that is injective on S.

Let $e_{n,d}$ denote the elementary symmetric polynomial of degree d in n variables. For $\pi \in \mathcal{F}$ and $i \in [1.55d]$, define the linear forms $\ell_{\pi,i} = \sum_{j\in\pi^{-1}(i)} x_j$. Let

$$f = \sum_{\pi\in\mathcal{F}} e_{1.55d,d}(\ell_{\pi,1}, \ldots, \ell_{\pi,1.55d}).$$

Since $\mathcal{F}$ is a splitter it follows that $f \in \mathcal{T}_{n,d}$. By using bounds on $|\mathcal{F}|$ and the Waring rank of $e_{n,d}$, it was shown in [37][Theorem 7] that the Waring rank of f is at most $4.075^d\,\mathrm{poly}(n)$. Since in general $\dim \mathrm{Diff}(f) \le \mathbf{R}(f)/(d + 1)$, we conclude that $B(n, d) \le \dim \mathrm{Diff}(f) \le 4.075^d\,\mathrm{poly}(n)$.

5.3 Abelian 2 Groups

Let k be a field of characteristic 2 of size at least n, and let A be a $d \times n$ matrix, any d columns of which are linearly independent. It was shown in Section 3.3 of [37] that the polynomial $f = \sum_{S\in\binom{[n]}{d}} \det(A_S)^2 \prod_{i\in S} x_i$ has Waring rank at most $2^d - 1$, and hence $\dim \mathrm{Diff}(f) \le (d + 1)(2^d - 1)$.

6 Further questions

1. We showed that $2^d \le B(n, d) < (\sqrt{27}/2)^d d$. Can these bounds be improved?
2. Let X be a generic Hankel matrix. Can the bound $\dim \mathrm{Diff}(\det X) \le (\sqrt{27}/2)^d d$ be improved? We suspect that the base of the exponent is optimal.
3. Let X be a generic Hankel matrix. Can the linear map $A_i : \mathrm{Diff}(\det X) \to \mathrm{Diff}(\det X)$ given by differentiation with respect to the any variable x_i be computed in linear time with respect to a spanning set of size $(\sqrt{27}/2)^d d$ (rather than φ^{2d})?
4. Do the methods of [21, 43, 40] have an interpretation in our framework?

References

1 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *Journal of the ACM (JACM)*, 42(4):844–856, 1995.

2 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.

3 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.

4 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.

5 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.

6 Vikraman Arvind, Abhranil Chatterjee, Rajit Datta, and Partha Mukhopadhyay. Fast exact algorithms using Hadamard product of polynomials. In Arkadev Chattopadhyay and Paul Gastin, editors, *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2019, December 11-13, 2019, Bombay, India*, volume 150 of *LIPIcs*, pages 9:1–9:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.FSTTCS.2019.9.

7 Vikraman Arvind, Abhranil Chatterjee, Rajit Datta, and Partha Mukhopadhyay. On explicit branching programs for the rectangular determinant and permanent polynomials. *Chic. J. Theor. Comput. Sci.*, 2020, 2020. URL: http://cjtcs.cs.uchicago.edu/articles/2020/2/contents.html.

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

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

10 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. doi:10.1109/FOCS.2010.24.

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

12 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. doi:10.1145/3188745.3188902.

13 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.

14 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.

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

16 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

17 Marek Cygan, Harold N Gabow, and Piotr Sankowski. Algorithmic applications of baur-strassen's theorem: Shortest cycles, diameter, and matchings. *Journal of the ACM (JACM)*, 62(4):1–30, 2015.

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

19 Richard Ehrenborg and Gian-Carlo Rota. Apolarity and Canonical Forms for Homogeneous Polynomials. *European Journal of Combinatorics*, 14(3):157–181, 1993. doi:10.1006/eujc.1993.1022.

20 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. doi:10.1145/2886094.

21 Fedor V Fomin, Daniel Lokshtanov, and Saket Saurabh. Efficient computation of representative sets with applications in parameterized and exact algorithms. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 142–151. SIAM, 2014.

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

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

24 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. doi:10.1145/780542.780545.

25 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.

26 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.

27 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.

28 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. doi:10.1016/j.jcss.2018.01.004.

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

30 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004.

31 Ioannis Koutis and Ryan Williams. Limits and applications of group algebras for parameterized problems. In *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I*, pages 653–664, 2009. doi:10.1007/978-3-642-02927-1_54.

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

33 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.

34 László Lovász. On determinants, matchings, and random algorithms. *Fundamentals of Computation Theory*, pages 565–574, 1979.

35 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.

36 Dániel Marx. A parameterized view on matroid optimization problems. *Theor. Comput. Sci.*, 410(44):4471–4479, 2009. doi:10.1016/j.tcs.2009.07.027.

37 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.

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

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

40 Dekel Tsur. Faster deterministic parameterized algorithm for k-Path. *Theoretical Computer Science*, 790:96–104, 2019. `doi:10.1016/j.tcs.2019.04.024`.

41 Ryan Williams. Finding paths of length k in $O^*(2^k)$ time. *Inf. Process. Lett.*, 109(6):315–318, 2009. `doi:10.1016/j.ipl.2008.11.004`.

42 Michał Włodarczyk. Clifford algebras meet tree decompositions. *Algorithmica*, 81(2):497–518, 2019.

43 Meirav Zehavi. Mixing color coding-related techniques. In *Algorithms-ESA 2015*, pages 1037–1049. Springer, 2015.