Foundations and Trends$^{\text{char}R\infty\ni}$ in
sample
Vol. xx, No xx (xxxx) 1–138
charc$\infty\ni$ xxxx xxxxxxxxx
DOI: xxxxxx

# Partial Derivatives in Arithmetic Complexity (and beyond)

**Xi Chen[1], Neeraj Kayal[2] and Avi Wigderson[3]**

[1] *Columbia University, New York 10027, USA, xichen@cs.columbia.edu*
[2] *Microsoft Research, Bangalore 560080, India, neeraka@microsoft.com*
[3] *IAS, Princeton 08540, USA, avi@ias.edu*

## Abstract

How complex is a given multivariate polynomial? The main point of this survey is that one can learn a great deal about the structure and complexity of polynomials by studying (some of) their partial derivatives. The bulk of the survey shows that partial derivatives provide essential ingredients in proving both upper and lower bounds for computing polynomials by a variety of natural arithmetic models. We will also see applications which go beyond computational complexity, where partial derivatives provide a wealth of structural information about polynomials (including their number of roots, reducibility and internal symmetries), and help us solve various number theoretic, geometric, and combinatorial problems.

# Contents

# 1

# Introduction

## 1.1 Motivation

Polynomials are perhaps the most important family of functions in mathematics. They feature in celebrated results from both antiquity and modern times, like the unsolvability by radicals of polynomials of degree $\geq 5$ of Abel and Galois, and Wiles' proof of Fermat's "last theorem". In computer science they feature in, e.g., error-correcting codes and probabilistic proofs, among many applications. The manipulation of polynomials is essential in numerous applications of linear algebra and symbolic computation. This survey is devoted mainly to the study of polynomials from a computational perspective. The books [BCS97, BM75, vzGG99] provide wide coverage of the area.

Given a polynomial over a field, a natural question to ask is how complex it is? A natural way to compute polynomials is via a sequence of arithmetic operations, e.g., by an arithmetic circuit, as shown in Figure 1.1 (formal definitions will be given in Section 1.2). One definition of how complex a polynomial is can be the size of the smallest arithmetic circuit computing it. A weaker model, often employed by mathematicians, is that of a formula (in which the underlying circuit structure must be a tree), and another definition of complexity may be the formula size.

There are many ways to compute a given polynomial. For example,

$$f(x_1, x_2) = x_1 \times (x_1 + x_2) + x_2 \times (x_1 + x_2) = (x_1 + x_2) \times (x_1 + x_2)$$

are two formulae for the same polynomial $f$, the first requiring 5 operations and the second only 3 operations. Finding the optimal circuit or formula computing a given polynomial is a challenging task, and even estimating that minimum size by giving upper and lower bounds is very difficult. Of course, the same is also true for the study of Boolean functions and their complexity (with respect to Boolean circuits and formulae, or Turing machines), but in the Boolean case we have a better understanding of that difficulty (via results on relativization by Baker, Gill and Solovay [BGS75], natural proofs due to Razborov and Rudich [RR94] and algebrization due to Aaronson and Wigderson [AW09]). For the arithmetic setting, which is anyway more structured, there seem to be more hope for progress.

Proving lower bounds for the complexity of polynomials has been one of the most challenging problems in theoretical computer science. Although it has received much attention in the past few decades, the progress of this field is slow. The best lower bound known in the general arithmetic circuit setting is still the classical $\Omega(n \log d)$ result by Baur and Strassen [BS83] (for some natural degree-$d$ polynomials over $n$ variables). Even for some very restricted models (e.g., constant-depth arithmetic circuits or multilinear formulae), a lot of interesting problems remain widely open. In this survey, we focus on the use of *partial derivatives* in this effort.

The study of upper bounds — constructing small circuits for computing important polynomials — is of course important for practical applications, and there are many non-trivial examples of such algorithms (e.g., Strassen's matrix multiplication algorithm [Str69], Berkowitz's algorithm for the determinant [Ber84] [1] and Kaltofen's black-box poly-

[1] The first NC algorithm for the determinant, based on Leverier's method, was given by Csanky in 1976 [Csa76]. However, Csanky's algorithm used divisions and was unsuitable for arbitrary fields. Around 1984, Berkowitz [Ber84] and independently, Chistov [Chi85]

nomial factorization algorithm [Kal89]). As we focus here on the uses of partial derivatives, we will see relatively few upper bounds, but we are certain that there is room for more, faster algorithms that use the partial derivatives of a polynomial when computing it.

The task of understanding arithmetic circuits and formulae naturally leads to the task of understanding the basic algebraic properties of the polynomials computed by such circuits and formulae. One such question is the following: given an arithmetic circuit, determine whether the polynomial computed by it is the identically zero polynomial or not. It turns out that besides being a natural scientific question, this question is also closely related to proving arithmetic circuit lower bounds, as shown by Impagliazzo and Kabanets [KI04]. Other natural structural questions relate to the symmetries of polynomials, the algebraic independence of systems of polynomials and more. Again, we will demonstrate the power of partial derivatives to help understand such structural questions.

### Organization

The rest of this chapter is devoted to formal definitions of the computational models (arithmetic circuits and formulae, and their complexity measures), and of partial derivatives.

In Part One, we demonstrate how partial derivatives can be used to probe the structure of polynomials, via a list of very different examples. In particular, we will see how to use them to prove that algebraic independence has matroid structure, and to determine the symmetries of a given family of polynomials. Along the way we will see that "most" polynomials have high arithmetic complexity. We will use partial derivatives to derive simple linear algebraic proofs to some important results on the number of solutions of polynomial equations whose initial proofs used algebraic geometry. (These will include Wooley's proof of Bezout's theorem and Stepanov's proof of Weil's theorem). We will also see the power of partial derivatives in resolving a long-standing problem in combinatorial geometry [GK08, KSS10].

came up with polylogarithmic depth arithmetic circuits for computing the determinant (and therefore also an NC algorithm for the determinant over arbitrary fields.)

In Part Two, we will review some of the most elegant lower bound proofs in the field, which use partial derivatives as a basic tool. Other than the $\Omega(n \log d)$ lower bound by Baur and Strassen for general arithmetic circuits, we will also be looking at some very restricted models of computation. The simplest one is based on the observation that every polynomial of degree $d$ can be expressed as the sum of $d$-th powers of affine linear forms. We will see that partial derivatives allow us to prove pretty sharp lower bounds in this model. We will also use partial derivatives to derive lower bounds for depth-3 arithmetic circuits and multilinear formulae. Another model of computation is based on the observation that every polynomial can be expressed as the determinant of a square matrix whose entries are affine linear forms. We will show how the second-order partial derivatives can be used to prove a quadratic lower bound for the permanent polynomial in this model.

Finally, in Part Three we will see how partial derivatives help in deriving upper bounds for various algebraic problems related to arithmetic circuits, such as identity testing, irreducibility testing and equivalence testing.

Many of the chapters in these three parts can be read independently. For the few which need background from previous chapters, we specify it in the abstract.

## 1.2 Arithmetic circuits

In this section, we define arithmetic circuits.

Let $\mathbb{F}$ be a field. Most of the time, it is safe to assume that $\mathbb{F}$ is of characteristic 0 or has a very large characteristic, e.g., $\mathrm{char}(\mathbb{F})$ is much larger than the degree of any relevant polynomial. We will point out explicitly when the results also hold for fields of small characteristic.

The underlying structure of an arithmetic circuit $\mathcal{C}$ is a directed acyclic graph $G = (V, E)$. We use $u, v$ and $w$ to denote vertices in $V$, and $uv$ to denote a directed edge in $E$. The role of a vertex $v \in V$ falls into one of the following cases:

(1) If the in-degree of $v$ is 0, then $v$ is called an input of the arithmetic circuit;

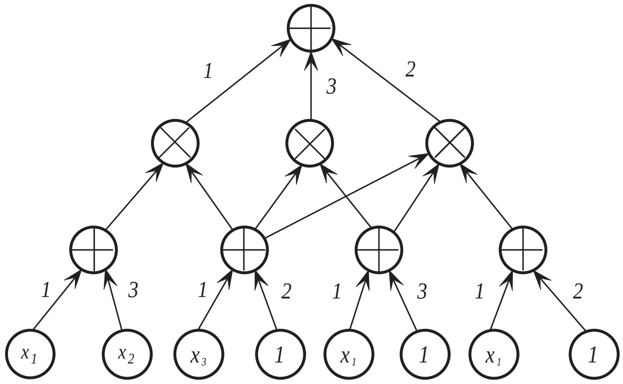

Fig. 1.1 A depth-3 Arithmetic Circuit over $\mathbb{F}[x_1, x_2, x_3]$

(2) Otherwise, $v$ is called a *gate*. In particular, if the out-degree of $v$ is 0, then $v$ is called an output (gate) of the circuit.

For most of the time, we will only discuss arithmetic circuits that compute one polynomial and have a single output gate. In this case, we will denote it by $\mathsf{out}_{\mathcal{C}} \in V$ (or simply $\mathsf{out} \in V$).

Every input vertex in $V$ is labeled with either one of the variables $x_1, \ldots, x_n$ or one of the elements in the field $\mathbb{F}$. Every gate is labeled with either "+" or "×", which are called *plus* gates and *product* gates respectively. Each edge $uv \in E$ entering a plus gate is also labeled with an element $c_{uv}$ in $\mathbb{F}$ (so plus gates perform "weighted addition" or in other words linear combinations of their inputs with field coefficients). See Figure 1.1 for an example.

Given an arithmetic circuit $\mathcal{C}$, we associate with each vertex $v \in V$ a polynomial $\mathcal{C}_v$, as the polynomial computed by $\mathcal{C}$ at $v$. Let $N^+(v)$ denote the set of successors and $N^-(v)$ denote the set of predecessors of $v$, then we define $\mathcal{C}_v$ inductively as follows: If $v$ is an input, then $\mathcal{C}_v$ is exactly the label of $v$. Otherwise (since $G$ is acyclic, when defining $\mathcal{C}_v$, we may assume the $\mathcal{C}_u$'s, $u \in N^-(v)$, have already been defined):

(1) If $v$ is a plus gate, then

$$\mathcal{C}_v = \sum_{u \in N^-(v)} c_{uv} \cdot \mathcal{C}_u, \quad \text{where } c_{uv} \in \mathbb{F} \text{ is the label of } uv \in E;$$

(2) If $v$ is a product gate, then

$$\mathcal{C}_v = \prod_{u \in N^-(v)} \mathcal{C}_u.$$

In particular, the polynomial $\mathcal{C}_{\mathsf{out}}$ associated with the output gate $\mathsf{out}$ is the polynomial computed by $\mathcal{C}$. We sometimes use $\mathcal{C}(x_1, \ldots, x_n)$ to denote the polynomial $\mathcal{C}_{\mathsf{out}}$ for short. We also need the notion of the *formal degree* of an arithmetic circuit, which is defined inductively using the following two basic rules:

(1) If $v \in V$ is a plus gate, then the formal degree of $v$ is the maximum of the formal degrees of the vertices $u \in N^-(v)$;

(2) If $v \in V$ is a product gate, then the formal degree of $v$ is the sum of the formal degrees of the vertices $u \in N^-(v)$.

**Definition 1.1.** The size of an arithmetic circuit, denoted by $\mathsf{S}(\mathcal{C})$, is the number of edges of its underlying graph.

Given a polynomial $f$, we let $\mathsf{S}(f)$ denote the size of the smallest arithmetic circuit computing $f$, that is,

$$\mathsf{S}(f) \overset{\text{def}}{=} \min_{\mathcal{C}\colon \mathcal{C}_{\mathsf{out}} = f} \mathsf{S}(\mathcal{C}).$$

The second way to define an arithmetic circuit (often referred to as a "straight-line program"), which is more convenient in certain situations, is to view it as a sequence of "+" and "×" operations:

$$\mathcal{C} = (g_1, \ldots, g_n, \ldots, g_m),$$

in which $g_i = x_i$ for all $i \in [n] = \{1, \ldots, n\}$. For each $k > n$, either

$$g_k = \sum_{i \in S} c_i \cdot g_i + c \quad \text{or} \quad g_k = \prod_{i \in S} g_i,$$

where $c, c_i \in \mathbb{F}$ and $S$ is a subset of $[k-1]$. Similarly, we can define a polynomial $\mathcal{C}_i$ for each $g_i$ and the polynomial computed by $\mathcal{C}$ is $\mathcal{C}_m$.

As a warm up, we take a brief look at the polynomials of the simplest form: univariate polynomials.

**Example 1.2.** $\mathsf{S}(x^d) = \Theta(\log d)$. This is done via "repeated squaring". Note that in an arithmetic circuit, the out-degree of a gate could be larger than 1 and there could be parallel edges.

**Example 1.3.** For every polynomial $f \in \mathbb{F}[x]$ of degree $d$, we have $\mathsf{S}(f) = O(d)$. For example, we can write $f = 3x^4 + 4x^3 + x^2 + 2x + 5$ as

$$f = x\,(x\,(x\,(3x+4)+1)+2)+5.$$

Although the two bounds above (the lower bound in Example 1.2 and the upper bound in Example 1.3) hold for every univariate polynomial, there is an exponential gap between them. It turns out that even for univariate polynomials, we do not have strong enough techniques for proving general size lower bounds.

**Open Problem 1.4.** Find an explicit family of polynomials

$$\left\{f_i\right\}_{i\in\mathbb{Z}^+} \subset \mathbb{F}[x], \quad \text{where } f_i \text{ has degree } i,$$

such that $\mathsf{S}(f_n) \neq (\log n)^{O(1)}$.

See Chapter 4 for some more discussion and clarification of what the word "explicit" means in the open problem above. We also provide a possible candidate for this open problem:

**Conjecture 1.5.** $\mathsf{S}\big((x+1)(x+2)\cdots(x+n)\big) \neq (\log n)^{O(1)}$.

This conjecture has a surprising connection to the (Boolean!) complexity of factoring integers.

**Exercise 1.6.** If *Conjecture 1.5* is false, then `Factoring` can be computed by polynomial size Boolean circuits.

As we go from univariate polynomials to multivariate polynomials, we encounter more algebraic structures, and the flavor of problems also changes. As Example 1.3 shows, every univariate polynomial of degree $n$ can always be computed by an arithmetic circuit of size $O(n)$. In contrast, the smallest arithmetic circuit for an $n$-variate polynomial of degree $n$ can potentially be exponential in $n$. However, no such explicit family of polynomials is known at present.

Let us say that a family of $n$-variate polynomials $\{f_n\}_{n \in \mathbb{Z}^+}$ has *low degree* if the degree of $f_n$ is $n^{O(1)}$. A large part of this survey is devoted to understanding families of low-degree polynomials. We will use partial derivatives as a tool to probe the structure of low-degree polynomials, and to prove lower bounds for them.

**Open Problem 1.7.** Find an explicit family of low-degree polynomials $\{f_n\}_{n \in \mathbb{Z}^+}$, $f_n \in \mathbb{F}[x_1, \ldots, x_n]$, such that $\mathsf{S}(f_n) \neq n^{O(1)}$.

For multivariate polynomials, it even makes sense to study families of constant-degree polynomials. The challenge is the following:

**Open Problem 1.8.** Find an explicit family of constant-degree polynomials $\{f_n\}_{n \in \mathbb{Z}^+}$, $f_n \in \mathbb{F}[x_1, \ldots, x_n]$, such that $\mathsf{S}(f_n) \neq O(n)$.

In other words, we want to find an explicit family of constant-degree polynomials for which the arithmetic complexity is superlinear, in the number of variables. Below we give a specific family of cubic (degree-3) polynomials for which resolving the above question is of significant practical importance. Let $f_n$ be the following polynomial in $3n^2$ variables $(x_{ij})_{1 \le i,j \le n}$, $(y_{ij})_{1 \le i,j \le n}$ and $(z_{ij})_{1 \le i,j \le n}$:

$$f_n \stackrel{\text{def}}{=} \sum_{i,j \in [n] \times [n]} z_{ij} \left( \sum_{k \in [n]} x_{ik} \cdot y_{kj} \right).$$

**Exercise 1.9.** For any $\omega \ge 2$, show that the product of two $n \times n$ matrices can be computed by arithmetic circuits of size $O(n^\omega)$ if and only if $\mathsf{S}(f_n) = O(n^\omega)$.

## 1.3 Formal derivatives and their properties

**Univariate polynomials.**

Let $\mathbb{F}$ denote a field, e.g., the set of real numbers $\mathbb{R}$. $\mathbb{F}$ could be finite but we normally assume its characteristic is large enough, e.g., much larger than the degree of any relevant polynomial. Let $\mathbb{F}[x]$ denote the set of univariate polynomials in $x$ over $\mathbb{F}$. Every $f \in \mathbb{F}[x]$ can be expressed as

$$f = a_m x^m + a_{m-1}x^{m-1} + \ldots + a_1 x + a_0,$$

where $m \in \mathbb{Z}_{\geq 0}$ and $a_i \in \mathbb{F}$ for all $0 \leq i \leq m$. The *formal derivative of $f$ with respect to $x$* is defined as

$$\frac{\partial f}{\partial x} \overset{\text{def}}{=} (ma_m)x^{m-1} + \big((m-1)a_{m-1}\big)x^{m-2} + \ldots + 2a_2 x + a_1.$$

It is called the formal derivative of $f$ because it does not depend on the concept of limit.

**Multivariate polynomials.**

Let $\mathbb{F}[x_1, \ldots, x_n]$, abbreviated as $\mathbb{F}[\mathbf{X}]$, denote the set of $n$-variate polynomials over $\mathbb{F}$, then every $f \in \mathbb{F}[\mathbf{X}]$ is a finite sum of monomials with coefficients in $\mathbb{F}$. For example,

$$f = x_1^2 x_2^3 x_3 + 2x_1^4 x_3^2$$

is a polynomial in $\mathbb{F}[x_1, x_2, x_3]$. Similarly we can define the formal partial derivative of $f$ with respect to $x_i$. To this end, we write $f$ as

$$f(x_1, \ldots, x_n) = g_m x_i^m + g_{m-1}x_i^{m-1} + \ldots + g_1 x_i + g_0,$$

where $g_i \in \mathbb{F}[x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$ for all $0 \leq i \leq m$. Then

$$\frac{\partial f}{\partial x_i} \overset{\text{def}}{=} (mg_m)x_i^{m-1} + \big((m-1)g_{m-1}\big)x_i^{m-2} + \ldots + (2g_2)x_i + g_1.$$

We use $\partial_{x_i}(f)$ as a shorthand for $\frac{\partial f}{\partial x_i}$. When the name of the variables is clear from the context, we shorten this further to simply $\partial_i(f)$.

Furthermore, we can take higher-order derivatives of $f$. Let $x_{i_1}, x_{i_2}, \ldots, x_{i_t}$ be a sequence of $t$ variables. Then we can take the $t$-th order derivative of $f$:

$$\frac{\partial}{\partial x_{i_t}}\left(\cdots\left(\frac{\partial}{\partial x_{i_1}}(f)\right)\right) \in \mathbb{F}[\mathbf{X}],$$

which we write compactly as $\partial_{i_t} \ldots \partial_{i_1}(f)$. Just like in calculus, it can be shown that the $t$-th order derivatives do not depend on the sequence but only depend on the multiset of variables $\{x_{i_1}, \ldots, x_{i_t}\}$.

Let $\mathbf{f} = (f_1, \ldots, f_k)$ be a sequence of $k$ polynomials, where $f_1, \ldots, f_k \in \mathbb{F}[\mathbf{X}]$. We define the *Jacobian matrix* of $\mathbf{f}$ as follows. For $f \in \mathbb{F}[\mathbf{X}]$ we use $\partial(f)$ to denote the $n$-dimensional vector:

$$\partial(f) \overset{\text{def}}{=} \begin{pmatrix} \partial_{x_1}(f) \\ \vdots \\ \partial_{x_n}(f) \end{pmatrix}.$$

Then the Jacobian matrix $\mathbf{J}(\mathbf{f})$ of $\mathbf{f}$ is the following $n \times k$ matrix:

$$\mathbf{J}(\mathbf{f}) \overset{\text{def}}{=} \big(\partial_{x_i}(f_j)\big)_{i\in[n],\, j\in[k]} = \big(\partial(f_1)\ \partial(f_2)\ \cdots\ \partial(f_k)\big).$$

**Exercise 1.10.** Show that given an arithmetic circuit $\mathcal{C}$ of size $s$, one can efficiently compute another arithmetic circuit of size $O(s \cdot n)$ with $n$ outputs, the outputs being the polynomials $\partial_{x_i}(\mathcal{C}(\mathbf{X}))$ for $i \in [n]$.

In [BS83], Baur and Strassen showed that these first-order partial derivatives of $\mathcal{C}(\mathbf{X})$ can actually be computed by an arithmetic circuit of size $O(s)$. We will see a proof in Chapter 9.

**Substitution maps.**

Consider now a univariate polynomial

$$f = a_m x^m + a_{m-1} x^{m-1} + \ldots + a_1 x + a_0$$

and its derivative

$$\frac{\partial f}{\partial x} = (m a_m) x^{m-1} + \big((m-1) a_{m-1}\big) x^{m-2} + \ldots + 2a_2 x + a_1.$$

Knowing $\partial_x(f)$ alone is not enough to determine $f$ itself, but observe that knowing $\partial_x(f)$ and the value $f(\alpha)$ of $f$ at *any* point $\alpha \in \mathbb{F}$, we can recover the polynomial $f$. More generally, for an $n$-variate polynomial

$f$, we can determine $f$ completely if we know all its first-order partial derivatives and the value $f(\alpha)$ for any point $\alpha \in \mathbb{F}^n$. This means that knowing the partial derivatives of $f$ and a substitution of $f$ is sufficient to determine all the properties of $f$, including its complexity. In some of the results presented in the survey, we will combine the use of partial derivatives with carefully chosen substitutions in order to enhance our understanding of a given polynomial $f$.

The substitution that is most natural and occurs frequently is the one where we substitute some of the variables to zero. For a polynomial $f \in \mathbb{F}[\mathbf{X}]$, we denote by $\sigma_i(f)$ the polynomial obtained by setting $x_i$ to zero. For example, for $f = x_1^2x_2^3x_3 + 2x_1^4x_3^2$, we have that $\sigma_1(f) = 0$ and $\sigma_2(f) = 2x_1^4x_3^2$.

**Exercise 1.11.** Let $f \in \mathbb{F}[x]$ be a univariate polynomial of degree at most $d$. Show that $f$ is the identically zero polynomial if and only if $\sigma(\partial^i(f)) = 0$ for all $0 \leq i \leq d$.

**Properties.**

The following properties of derivatives and substitution maps are easy to verify.

**Property 1.12.** For any $f, g \in \mathbb{F}[\mathbf{X}]$, $\alpha, \beta \in \mathbb{F}$ and $i \in [n]$:

- *Linearity of derivatives.* $\partial_i(\alpha f + \beta g) = \alpha \cdot \partial_i(f) + \beta \cdot \partial_i(g)$.
- *Derivative of product.* $\partial_i(f \cdot g) = \partial_i(f) \cdot g + f \cdot \partial_i(g)$.
- *Linearity of substitution.* $\sigma_i(\alpha f + \beta g) = \alpha \cdot \sigma_i(f) + \beta \cdot \sigma_i(g)$.
- *Substitution preserves multiplication.* $\sigma_i(f \cdot g) = \sigma_i(f) \cdot \sigma_i(g)$.

We also need the counterpart of the *chain rule* in calculus.

Let $g \in \mathbb{F}[z_1, \ldots, z_k] = \mathbb{F}[\mathbf{Z}]$, and $\mathbf{f} = (f_1, \ldots, f_k)$ be a tuple where each $f_i$ is a polynomial in $\mathbb{F}[\mathbf{X}]$. The composition $g \circ \mathbf{f}$ of $g$ and $\mathbf{f}$ is a polynomial in $\mathbb{F}[\mathbf{X}]$ where

$$g \circ \mathbf{f}(\mathbf{X}) = g\big(f_1(\mathbf{X}), f_2(\mathbf{X}), \ldots, f_k(\mathbf{X})\big).$$

**Property 1.13 (The Chain Rule).** For every $i \in [n]$, we have

$$\partial_{x_i}\big(g \circ \mathbf{f}\big) = \sum_{j=1}^{k} \partial_{f_j}(g) \cdot \partial_{x_i}(f_j),$$

where we use $\partial_{f_j}(g)$ to denote $\partial_{z_j}(g) \circ \mathbf{f} \in \mathbb{F}[\mathbf{X}]$ for all $j \in [k]$.

In the rest of this survey, unless mentioned otherwise, we will assume the underlying field $\mathbb{F}$ to be $\mathbb{C}$, the field of complex numbers. A notable exception is chapter 8, where we will work with finite fields. This is all that we need for now. We will introduce some shorthand notation later as needed.

# Part I: Structure

## Overview

Examining the partial derivatives can sometimes provide useful information about a polynomial $f$ that is far from being apparent if we just view $f$ as a list of coefficients. This is useful in understanding polynomials and collections of polynomials. In the first part of this survey, we will see some examples of this phenomenon. While the set of applications and their contexts are diverse, in almost all examples here a key part is a construction, usually via a dimension argument, of an auxiliary polynomial $F$ which vanishes on a chosen set of points, sometimes to high multiplicity. The power of this idea goes beyond the examples presented here, and we note, e.g., its applications to other areas, such as decoding, list-decoding and local decoding of error-correcting codes, as e.g. in Sudan [Sud97], Guruswami and Sudan [GS99] and Kopparty, Saraf and Yekhanin [KSY11]. Partial derivatives, especially the Jacobian, play various and sometimes multiple roles in these results. We now describe the structural applications discussed in this part.

In chapter 2, we will see how the partial derivatives can help us determine completely the symmetries of certain polynomials.

In chapter 3, we turn to algebraic independence of families of polynomials. Here we use partial derivatives to obtain the somewhat surprising fact that, like linear independence, algebraic independence is a matroid. The Jacobian of the family of polynomials (the matrix of all their first-order partial derivatives) plays a crucial role here.

In chapter 4, we use the notion of “annihilating polynomial” from the previous chapter to give a combinatorial proof that “most” polynomials have high arithmetic circuit complexity.

In chapter 5, we return to the Jacobian, and define and prove an affine version of Bezout’s theorem (a form convenient especially to computer scientists and analytic number theorists). Here we follow an elementary proof of Wooley, again using an “annihilating polynomial”, which works also over finite fields (and even some rings) in the same way it works over algebraically closed fields. In this proof we will see the role of the Jacobian in “Hensel Lifting”, a procedure analogous to “Newton Iteration”, but over finite fields.

In chapter 6, we bring two other aspects of the Jacobian. First we describe its use in the construction of polynomial maps that are “algebraic extractors”. Then we describe the “Jacobian conjecture”, a central long-standing problem about it.

In chapter 7, we show the usefulness of this approach of annihilating polynomials and partial derivatives for combinatorial geometry, explaining the solution of the “Joints conjecture”.

In chapter 8, we will explain the Stepanov method for bounding the number of common zeros of a pair of univariate polynomials. We will demonstrate it on two examples — a special case of the Weil bound on $\mathbb{F}_p$-rational points on curves (Stepanov’s original application), and a polynomial arising in the study of the Heilbronn exponential sum, due to Heath-Brown and Mit’kin.

# 2

# Symmetries of a polynomial

Examining the partial derivatives can sometimes provide useful information about a polynomial $f$ that is far from being apparent if we just view $f$ as a list of coefficients. In this chapter we look at symmetries of a polynomial, i.e. the set of all invertible linear transformations on a polynomial which keep it fixed. We will see how partial derivatives help us determine completely the set of symmetries of some polynomials. We will also the introduce the reader to the Hessian, a matrix of second-order partial derivatives that will be used for proving lower bounds in chapter 13.

The *symmetries* of an $n$-variate polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ is the set of all invertible linear transformations $A \in GL_n(\mathbb{F})$ such that

$$f(A \cdot \mathbf{X}) = f(\mathbf{X}).$$

That is, the symmetries of $f$ is the set of all invertible $n \times n$ matrices $A = (a_{ij})_{n \times n}$ such that

$$f(\mathbf{X}) = f(a_{11}x_1 + \ldots + a_{1n}x_n, \ldots, a_{n1}x_1 + \ldots + a_{nn}x_n).$$

The symmetries of $f$ clearly form a subgroup of $GL_n(\mathbb{F})$. This group is sometimes called the *automorphism group* of $f$. Understanding the symmetries can sometimes provide valuable insights into the structure of $f$. Consider for example the $n$-variate power symmetric polynomial

$$P_n^d(x_1, x_2, \ldots, x_n) \overset{\text{def}}{=} x_1^d + x_2^d + \ldots + x_n^d.$$

Note that if we swap any two variables, say $x_1$ and $x_2$, we get back the same polynomial:

$$P_n^d(x_2, x_1, x_3, \ldots, x_n) = P_n^d(x_1, x_2, x_3, \ldots, x_n).$$

Let $\omega$ be a primitive $d$-th root of unity. Over the field $\mathbb{C}$ of complex numbers one can take $\omega$ to be $e^{2\pi i/d}$. For the polynomial $P_n^d$ above, we also have the following symmetry:

$$P_n^d(\omega x_1, x_2, \ldots, x_n) = P_n^d(x_1, x_2, \ldots, x_n).$$

Applying these symmetries repeatedly, we get a large group of symmetries of the above polynomial $P_n^d$ consisting of permuting the variables and multiplying them by powers of $\omega$. Besides these symmetries, are there others that we might have missed? As we will see shortly, for $d \geq 3$, these are actually all the symmetries of $P_n^d$ and there are no more. The above discussion is summarized in the following lemma:

**Lemma 2.1.** Let us denote by $G$ the subgroup of $GL_n(\mathbb{F})$, which is generated by the set of permutation matrices and by diagonal matrices of the form

$$\begin{pmatrix} \omega^{i_1} & 0 & \ldots & 0 \\ 0 & \omega^{i_2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \omega^{i_n} \end{pmatrix}.$$

Then for $d \geq 3$ the automorphism group of $P_n^d = x_1^d + x_2^d + \ldots + x_n^d$ is precisely the group $G$ above.

Before proving Lemma 2.1, we give some more special properties of the family of power symmetric polynomials. Our discussion below will also set the stage for some of the lower bound proofs to be discussed in Part Two.

First, the power symmetric polynomial $P_n^d$ has a quite exceptional property — it is completely characterized by its symmetries:

**Exercise 2.2 (Characterization by symmetries property).** If the automorphism group of an $n$-variate homogeneous polynomial $f$ of degree $d$ contains the group $G$ as defined in Lemma 2.1, then $f$ is precisely the polynomial

$$c \cdot (x_1^d + x_2^d + \ldots + x_n^d)$$

for some constant $c \in \mathbb{F}$.

There are at least three other families of polynomials known to have this remarkable property that the polynomials are completely characterized by their symmetries. They are the *determinant*, the *permanent*, and the *sum-product polynomials* defined below:

$$S_m^n \stackrel{\text{def}}{=} \sum_{i=1}^{m} \prod_{j=1}^{n} x_{ij} \,.$$

**Exercise 2.3 (Characterization by symmetries property).** Let $G$ be the automorphism group of $S_m^n$. Show that if the automorphism group of a homogeneous polynomial of degree $n$ in $mn$ variables contains $G$ as a subgroup, then it must be a scalar multiple of $S_m^n$.

Second, these four families of polynomials — the power symmetric polynomials, the sum-product polynomials, the determinant, and the permanent — also share the following nice property. Let $\mathcal{P}$ denote any of these four families. Then *every* polynomial $f$ can be expressed as a '*projection*' of a (possibly larger) polynomial from $\mathcal{P}$. A polynomial $f$ is said to be a projection of another polynomial $g$ if $f$ can be obtained by replacing each variable of $g$ by an affine form in the variables of $f$. Let us illustrate this for the power symmetric polynomials using the following lemma due to Ellison [Ell69].

**Lemma 2.4.** Every polynomial $f \in \mathbb{C}[y_1, \ldots, y_m]$ of degree $d$ can be obtained by replacing each variable of $P_n^d$, for some $n$, with an affine form in $y_1, \ldots, y_m$. That is, given *any* polynomial $f \in \mathbb{C}[y_1, \ldots, y_m]$ of degree $d$, there exist an integer $n$ and a set

$$\bigl\{\alpha_{ij} : i \in [n], j \in [0 : m]\bigr\} \subset \mathbb{C}$$

such that

$$f(y_1, \ldots, y_m) = P_n^d\bigl(\ell_1, \ell_2, \ldots, \ell_n\bigr),$$

where $\ell_i = \alpha_{i0} + \alpha_{i1} y_1 + \ldots + \alpha_{im} y_m$ for any $i \in [n]$.

The proof is short and elegant, and we reproduce it at the end of this chapter. Analogous statements also hold for the other three families of polynomials. The projections of a sum-product polynomial correspond to depth-3 arithmetic circuits (see Chapter 11) for computing the projected polynomials. The projections of the determinant *roughly* correspond to formulae for computing the projected polynomials.

For concreteness, let us fix our attention on the determinant in the discussion that follows. Analogous statements can be made for any of the four families above. It turns out that even though any polynomial $f$ can be expressed as a projection of the determinant, it might well be that $f$ is a projection only of an *exponentially* large determinant. It therefore makes sense to ask the following question:

> Given a polynomial $f$, what is the *smallest* value of $n$ such that $f$ can be expressed as a projection of the $n$-by-$n$ determinant.

This is a particularly interesting question because it is known that the smallest value of $n$ above *roughly* corresponds to the size of the smallest formula for computing $f$ (cf. Valiant [Val79a]).

Our present state of knowledge is such that we do not know of any explicit family of polynomials which cannot be expressed as a projection of determinants of polynomial sized matrices. It is conjectured that the permanent is such a family. This problem is commonly referred to as the *determinant versus permanent* problem. There is an approach to this problem, which uses some exceptional properties of the determinant and permanent to translate the determinant versus permanent question into a problem in the representation theory of groups. This approach pursued by Mulmuley and Sohoni is now referred to as the Geometric Complexity Theory, and the reader is referred to the survey articles [Mul09b, Mul09a] and to the series of papers beginning with [Mul99, MS01, MS08]. In Chapter 13, we will give a quadratic lower bound for the determinant versus permanent problem due to Mignon and Ressayre [MR04].

The analogous question for the family of sum-product polynomials is also particularly interesting. The smallest $m$ such that $f$ is a projection of $S_m^d$ corresponds to its depth-3 arithmetic circuit complexity. The construction of an explicit family of polynomials which cannot be expressed as projections of polynomial-sized sum-product polynomials (equivalently, polynomials which cannot be computed by polynomial-sized depth-3 arithmetic circuits) remains a very tantalizing open problem. Some of the partial results in this direction will be discussed in Part Two.

Even though we have not made much progress and most of the questions posed above are still widely open, the situation for the power-symmetric polynomials is quite good. In chapter 10, we will see an explicit family of polynomials that can only be expressed as projections of exponential-sized power-symmetric polynomials. In chapter 9, we will also see an optimal lower bound for the arithmetic circuit complexity of the power-symmetric polynomials themselves.

## Proof of lemma 2.1

We now prove Lemma 2.1 as promised. The key idea involved here is to use the *Hessian* matrix.

**Definition 2.5.** Given an $n$-variate polynomial $f(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$, the *Hessian* matrix $H_f(\mathbf{X}) \in (\mathbb{F}[\mathbf{X}])^{n \times n}$ is defined as follows:

$$H_f(\mathbf{X}) \stackrel{\text{def}}{=} \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \cdot \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_1 \cdot \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \cdot \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \cdot \partial x_n} \end{bmatrix}.$$

The most interesting property of the Hessian matrix of a polynomial $f$ is the effect that a linear transformation of the variables has on it.

**Lemma 2.6.** Let $f(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ be an $n$-variate polynomial and $A \in \mathbb{F}^{n\times n}$ be a linear transformation. Let

$$F(\mathbf{X}) \overset{\text{def}}{=} f(A \cdot \mathbf{X}).$$

Then we have

$$H_F(\mathbf{X}) = A^T \cdot H_f(A \cdot \mathbf{X}) \cdot A.$$

In particular, $\textsc{Det}\big(H_F(\mathbf{X})\big) = \textsc{Det}(A)^2 \cdot \textsc{Det}\big(H_f(A \cdot \mathbf{X})\big)$.

*Proof.* By the chain rule for differentiation, we have for all $1 \leq i \leq n$:

$$\frac{\partial F}{\partial x_i} = \sum_{k=1}^{n} a_{ki} \cdot \frac{\partial f}{\partial x_k}\big(A \cdot \mathbf{X}\big).$$

Therefore for all $1 \leq i, j \leq n$:

$$\begin{aligned} \frac{\partial^2 F}{\partial x_i \cdot \partial x_j} &= \sum_{k=1}^{n} a_{ki} \cdot \left( \sum_{\ell=1}^{n} a_{\ell j} \cdot \frac{\partial^2 f}{\partial x_k \cdot \partial x_\ell}\big(A \cdot \mathbf{X}\big) \right) \\ &= \sum_{k,\ell \in [n]} a_{ki} \cdot \frac{\partial^2 f}{\partial x_k \cdot \partial x_\ell}\big(A \cdot \mathbf{X}\big) \cdot a_{\ell j}. \end{aligned}$$

Putting these equations into matrix form gives us the lemma. □

Lemma 2.6 above has a very useful generalization that we give below. We leave the proof as an exercise. In Chapter 13, we will see how to use this generalization to prove a lower bound on the determinental complexity of the permanent.

**Lemma 2.7. Generalization of Lemma 2.6.** Let $f(x_1, \ldots, x_m)$ be an $m$-variate polynomial. Let $A \in \mathbb{F}^{m\times n}$ be a matrix and $\mathbf{b} \in \mathbb{F}^m$ be a vector. Let $F(\mathbf{y}) \in \mathbb{F}[\mathbf{y}]$ be an $n$-variate polynomial such that

$$F(\mathbf{y}) = f(A \cdot \mathbf{y} + \mathbf{b}).$$

In other words, the polynomial $F$ is obtained from $f$ by replacing the $i$-th variable of $f$ by the affine form

$$(a_{i1}y_1 + a_{i2}y_2 + \ldots + a_{in}y_n + b_i).$$

Then we have

$$H_F(\mathbf{y}) = A^T \cdot H_f(A \cdot \mathbf{y} + \mathbf{b}) \cdot A$$

---

We now apply Lemma 2.6 to the power-symmetric polynomial $P_n^d$ and take $A \in GL_n(\mathbb{F})$ to be an automorphism of $P_n^d$. We then have

$$\textsc{Det}\big(H_{P_n^d}(\mathbf{X})\big) = \textsc{Det}(A)^2 \cdot \textsc{Det}\big(H_{P_n^d}(A \cdot \mathbf{X})\big).$$

For the power-symmetric polynomial, the Hessian matrix is a diagonal matrix with the $(i,i)$-th entry being $d(d-1)x_i^{d-2}$. Thus

$$\textsc{Det}\big(H_{P_n^d}(\mathbf{X})\big) = \prod_{i=1}^{n} d(d-1)x_i^{d-2}.$$

The above equation then yields

$$\prod_{i=1}^{n} x_i^{d-2} = \textsc{Det}(A)^2 \cdot \prod_{i=1}^{n} \left( \sum_{j=1}^{n} a_{ij}x_j \right)^{d-2}, \tag{2.1}$$

where the $a_{ij}$'s are the entries of the automorphism $A$. By unique factorization we get that each $\sum_{j\in[n]} a_{ij}x_j$ is a scalar multiple of some $x_k$. Put differently, applying unique factorization to equation (2.1) yields that the matrix $A$ is the product of a permutation matrix with a diagonal matrix. We already know that any permutation of the $x_i$'s is an automorphism of $P_n^d$, so let us look at the diagonal part. Note that if

$$(\lambda_1 x_1)^d + \ldots + (\lambda_n x_n)^d = x_1^d + \ldots + x_n^d,$$

then each $\lambda_i$, $i \in [n]$, is a $d$-th root of unity. This means that the matrix $A$ must be a product of a permutation matrix with a diagonal matrix consisting of $d$-th roots of unity on the diagonal. This completes the proof of Lemma 2.1. □

The Hessian matrix can also be used to deduce the automorphisms of sum-product polynomials.

**Exercise 2.8.** The automorphism group of the sum-product polynomial $S_m^n$ is generated by the following three kinds of automorphisms:

- Let $\sigma \in S_m$ be a permutation on $m$ elements. The automorphism $A_\sigma$ which sends $x_{ij}$ to $x_{\sigma(i)j}$;
- Let $\pi \in S_n$ be a permutation on $n$ elements. The automorphism $A_\pi$ which sends $x_{ij}$ to $x_{i\pi(j)}$;
- Let $\lambda \in \mathbb{F}^*$ be a nonzero field element. The automorphism $A_\lambda$ defined as follows:

$$A_\lambda : \begin{cases} x_{11} \mapsto \lambda x_{11} \\ x_{12} \mapsto \lambda^{-1} x_{12} \\ x_{ij} \mapsto x_{ij} \quad \text{if } (i,j) \notin \{(1,1),(1,2)\} \end{cases}$$

In general, computing the automorphism group of a given polynomial is a very difficult task, and no efficient algorithm is known even for cubic polynomials. The following exercise suggests an explanation by showing that this problem is at least as hard as Graph Automorphism.

**Exercise 2.9.** Show that computing the symmetries of a given cubic polynomial $f$ is at least as hard as the problem of computing the automorphisms of a given graph.
(Hint: Let $G = (V, E)$ be a graph. Consider the following polynomial

$$\textstyle\sum_{i \in [|V|]} x_i^3 + \sum_{\{i,j\} \in E} x_i x_j$$

over variables $x_1, \ldots, x_{|V|}$.) [1]

In chapter 4 we will see how to use partial derivatives to determine the symmetries of some other interesting families of polynomials.

[1] The exercise can be modified slightly to show that the problem of deciding whether two cubic polynomials are equivalent or not is at least as difficult as Graph Isomorphism. This gives a different proof of a slightly weaker form of a theorem due to Agrawal and Saxena [AS06].

## Proof of lemma 2.4

*Proof.* It is sufficient to prove that every monomial of degree $d$ can be expressed as a sum of the $d$-th powers of linear forms. Let $x_1^{\alpha_1} \dots x_n^{\alpha_n}$ be a monomial of total degree $d$. Indeed, it suffices to prove it for the case when $n = 2$, because if we have this case then

$$x_1^{\alpha_1} x_2^{\alpha_2} = \sum_i \ell_i^{\alpha_1+\alpha_2}(x_1, x_2)$$

and

$$x_1^{\alpha_1} x_2^{\alpha_2} x_3^{\alpha_3} = \sum_i \ell_i^{\alpha_1+\alpha_2}(x_1, x_2) \cdot x_3^{\alpha_3}.$$

By the case of $n = 2$,

$$\ell_i^{\alpha_1+\alpha_2} \cdot x_3^{\alpha_3} = \sum_j m_j^{\alpha_1+\alpha_2+\alpha_3}(x_1, x_2, x_3)$$

and thus, we have the cases of $n = 3, 4, \dots$ by induction (both the $\ell_i$'s and the $m_j$'s above are linear forms).

**The case of $n = 2$.** Note that since we are working over the complex numbers, it suffices to show that for every $\alpha_1, \alpha_2 \in \mathbb{Z}_{\geq 0}$, we can express the monomial $x_1^{\alpha_1} x_2^{\alpha_2}$ in the following way:

$$x_1^{\alpha_1} x_2^{\alpha_2} = \sum_{i=0}^{\alpha_1+\alpha_2} \beta_i \cdot (x_1 + i x_2)^{\alpha_1+\alpha_2}$$

In other words, it suffices to show that the following $d+1$ polynomials:

$$\left\{(x_1 + i x_2)^d : i \in [0 : d]\right\}$$

are linearly independent when viewed as vectors in $\mathbb{C}^{d+1}$, which follows directly from the invertibility of the Vandermonde matrix $((i^j))_{0 \leq i,j \leq d}$. This completes the proof of Lemma 2.4. □

# 3

# Algebraic independence

In this chapter, we introduce the reader to the notion of algebraic independence and show that it shares the matroid structure of linear independence. In doing so we use the Jacobian matrix, a matrix consisting of partial derivatives that helps us decide whether a set of polynomials is algebraically independent or not.

In this chapter, we will see how partial derivatives help in deciding whether a set of polynomials is *algebraically independent* or not. Algebraic independence is a significant, non-linear generalization of linear independence. However, by using partial derivatives we will show that the basic *matroid* structure of linear independence is also shared by algebraic independence [ER93, Oxl06]. This was observed by van der Waerden in his "Moderne Algebra" [vdW37].

**Definition 3.1.** Let $f_1, f_2, \ldots, f_k$ be $k$ polynomials in $\mathbb{F}[x_1, \ldots, x_n] = \mathbb{F}[\mathbf{X}]$. They are said to be *algebraically dependent* if there is a nonzero $g \in \mathbb{F}[z_1, \ldots, z_k] = \mathbb{F}[\mathbf{Z}]$ such that $g \circ \mathbf{f} \equiv 0$, where $\mathbf{f} := (f_1, \ldots, f_k)$. Otherwise, $f_1, \ldots, f_k$ are algebraically independent. We call a nonzero polynomial $g$ an *annihilating polynomial* of $\mathbf{f}$ if $g \circ \mathbf{f} \equiv 0$.

We will use the following fact without a proof and leave its simplest case as an exercise. A hint is to count the degrees of freedom in

$$f_1^{i_1}\cdots f_k^{i_k}, \qquad \text{where } i_1\cdot\deg(f_1)+\cdots+i_k\cdot\deg(f_k)\le D$$

for large enough $D$.

**Fact 3.2.** If $k > n$, then $f_1,\ldots,f_k$ are algebraically dependent.

**Exercise 3.3.** Show that for any $f_1$ and $f_2 \in \mathbb{F}[x]$, there must exist a nonzero $g \in \mathbb{F}[z_1, z_2]$ such that $g \circ (f_1, f_2) \equiv 0$.

Algebraic independence is a generalization of linear independence (over field $\mathbb{F}$) since $f_1,\ldots,f_k$ are linearly independent over $\mathbb{F}$ iff there is a linear form $g=\sum_{i=1}^k c_i z_i \in \mathbb{F}[\mathbf{Z}]$ such that $g\circ\mathbf{f}\equiv 0$. As we know, linear independence has the following *extension property*:

> Suppose $f_1,\ldots,f_k$ are linearly independent and $f_1',\ldots,f_{k+1}'$ are also linearly independent. Then one can always find an $f_i'$ from the latter collection and move it to the former, so that $f_1,\ldots,f_k,f_i'$ are still linearly independent.

This is the defining property of *matroids*, a combinatorial extension of linear independence [Oxl06]. Indeed we will prove that it also holds for algebraic independence. We will show that $f_1,\ldots,f_k$ are algebraically independent if and only if the $k$ vectors $\partial(f_1),\ldots,\partial(f_k)$ are linearly independent over the polynomial ring $\mathbb{F}[\mathbf{X}]$. (The definition of $\partial(f_i)$ and $\mathbf{J}(f_1,\ldots,f_n)$ can be found in Section 1.3.)

**Theorem 3.4.** The polynomials $f_1,\ldots,f_n \in \mathbb{F}[\mathbf{X}]$ are algebraically independent if and only if $\mathbf{J}(f_1,\ldots,f_n)$ has full rank over $\mathbb{F}[\mathbf{X}]$ (which means that if

$$\mathbf{J}(f_1,\ldots,f_n)\cdot\begin{pmatrix} g_1\\ \vdots\\ g_n\end{pmatrix}\equiv\mathbf{0}$$

for $g_1,\ldots,g_n\in\mathbb{F}[\mathbf{X}]$, then $g_1\equiv\ldots\equiv g_n\equiv 0$).

Before proving this theorem, note that it implies the following extension property.

**Corollary 3.5.** The polynomials $f_1, \dots, f_k \in \mathbb{F}[\mathbf{X}]$ are algebraically independent iff $k \le n$ and $\mathbf{J}(f_1, \dots, f_k)$ has rank $k$ over $\mathbb{F}[\mathbf{X}]$.

**Corollary 3.6.** Suppose both $f_1, \dots, f_k$ and $f'_1, \dots, f'_k, f'_{k+1}$ are algebraically independent, then there exists an $i \in [k+1]$ such that $f_1, \dots, f_k, f'_i$ are still algebraically independent.

*Proof.* [Proof of Theorem 3.4] We use $\mathbf{f}$ to denote $(f_1, \dots, f_n)$, and $\mathbf{J}$ to denote the Jacobian matrix $\mathbf{J}(\mathbf{f})$.

($\Leftarrow$) Suppose $\mathbf{J}$ has full rank but there exists a nonzero $g$ such that $g \circ \mathbf{f} \equiv 0$. Let $g$ be one of such polynomials of minimum degree. Since $g \circ \mathbf{f} \equiv 0$, we have

$$\mathbf{0} = \begin{pmatrix} \partial_{x_1}(g \circ \mathbf{f}) \\ \vdots \\ \partial_{x_n}(g \circ \mathbf{f}) \end{pmatrix} = \mathbf{J} \cdot \begin{pmatrix} \partial_{f_1}(g) \\ \vdots \\ \partial_{f_n}(g) \end{pmatrix}.$$

However, since $\mathbf{J}$ has full rank, we have $\partial_{f_i}(g) = \partial_{z_i}(g) \circ \mathbf{f} \equiv 0$ for all $i \in [n]$. Note that the degree of $\partial_{z_i}(g) \in \mathbb{F}[\mathbf{Z}]$ is smaller than that of $g$. Due to the assumption on $g$, we have $\partial_{z_i}(g) \equiv 0$ for all $i \in [n]$. As we assumed $\mathbb{F}$ to be $\mathbb{C}$, $g$ must be the identically zero polynomial, which contradicts with our assumption that $g$ is nonzero.

($\Rightarrow$) Suppose $f_1, \dots, f_n$ are algebraically independent. We will find an $n \times n$ matrix $\mathbf{J}^*$ such that $\mathbf{J} \cdot \mathbf{J}^*$ is a diagonal matrix and every diagonal entry is nonzero. As a result, $\mathbf{J}$ has full rank. First let $h_1 \in \mathbb{F}[\mathbf{X}]$ be $h_1(\mathbf{X}) = x_1$ (Here we use $h_1$, instead of using $x_1$ directly, to make the presentation easier to follow). Then by Fact 3.2 there must exist a nonzero $g_1 \in \mathbb{F}[z_1, \dots, z_{n+1}]$ such that $g_1(f_1, \dots, f_n, h_1) \equiv 0$. Again, we assume $g_1$ to be one of such polynomials of minimum degree. Using the chain rule, we have

$$\mathbf{0} = \partial\big(g_1\left(f_1, \ldots, f_n, h_1\right)\big) = \begin{pmatrix} \mathbf{J}(f_1, \ldots, f_n) & \begin{pmatrix}1\\0\\\vdots\\0\\0\end{pmatrix}\end{pmatrix} \begin{pmatrix}\partial_{f_1}(g_1)\\\partial_{f_2}(g_1)\\\vdots\\\partial_{f_n}(g_1)\\\partial_{h_1}(g_1)\end{pmatrix}.$$

After rearrangement, we get

$$\mathbf{J}(f_1, \ldots, f_n) \begin{pmatrix}\partial_{f_1}(g_1)\\\partial_{f_2}(g_1)\\\vdots\\\partial_{f_n}(g_1)\end{pmatrix} = \begin{pmatrix}-\partial_{h_1}(g_1)\\0\\\vdots\\0\end{pmatrix}$$

Note that $\partial_{h_1}(g_1) = \partial_{z_{n+1}}(g_1) \circ (f_1, \ldots, f_n, h_1) \in \mathbb{F}[\mathbf{X}]$ cannot be zero. This is because, if $\partial_{h_1}(g_1) \equiv 0$, then we must have $\partial_{z_{n+1}}(g_1) \equiv 0$ due to the assumption that $g_1$ is of the minimum degree. It then implies that $g_1$ is independent of $z_{n+1}$ and thus, $f_1, f_2, \ldots, f_n$ are algebraically dependent, contradicting with our assumption.

We repeat the process above for every $h_i(\mathbf{X}) = x_i$, $i \in [2 : n]$. For each $h_i$, we use $g_i$ to denote one of the polynomials of minimum degree such that $g_i(f_1, \ldots, f_n, h_i) \equiv 0$. Combining all the equations, we have

$$\mathbf{J} \cdot \begin{pmatrix}\partial_{f_1}(g_1) & \partial_{f_1}(g_2) & \cdots & \partial_{f_1}(g_n)\\\partial_{f_2}(g_1) & \partial_{f_2}(g_2) & \cdots & \partial_{f_2}(g_n)\\\vdots & \vdots & \ddots & \vdots\\\partial_{f_n}(g_1) & \partial_{f_n}(g_2) & \cdots & \partial_{f_n}(g_n)\end{pmatrix}$$

$$= (-1) \begin{pmatrix}\partial_{h_1}(g_1) & 0 & \cdots & 0\\0 & \partial_{h_2}(g_2) & \cdots & 0\\\vdots & \vdots & \ddots & \vdots\\0 & 0 & \cdots & \partial_{h_n}(g_n)\end{pmatrix},$$

and $\partial_{h_i}(g_i) \not\equiv 0$ for all $i \in [n]$. As a result, $\mathbf{J}$ must have full rank. □

**Exercise 3.7.** Given arithmetic circuits $\mathcal{C}_1, \ldots, \mathcal{C}_k$ in variables $x_1, \ldots, x_n$, show that the problem of testing if the polynomials they compute are algebraically independent is in $\mathsf{RP}$.

**Open Problem 3.8.** Is there a *deterministic polynomial-time* algorithm for testing algebraic independence?

In [Kay09], Kayal has given an explicit set of quadratic polynomials $f_1, f_2, \ldots, f_n \in \mathbb{F}[x_1, \ldots, x_n]$ which are algebraically dependent but the smallest polynomial $g$ such that $g \circ (f_1, f_2, \ldots, f_n) = 0$ has degree $2^n$. Moreover, there also exist algebraically dependent quadratic polynomials so that any polynomial $g$ satisfying $g \circ (f_1, f_2, \ldots, f_n) = 0$ has superpolynomial arithmetic circuit size (unless the polynomial hierarchy collapses). Our exposition above shows that partial derivatives can help us decide the existence of such a $g$ even though the polynomial $g$ itself may be too complex to write down explicitly.

# 4

# Polynomials with high arithmetic complexity

How complex is a *random* $n$-variate polynomial of degree $d$? In this chapter, we use the notion of algebraic independence introduced in chapter 3 to give an answer to this question.

In this chapter, we use the notion of algebraic dependence to show that there exist polynomials of small degree which have very high arithmetic complexity. It is a folklore result. In fact, like in the Boolean world, "most" polynomials have high arithmetic complexity. And just like in the Boolean world, we do not know any explicit family of polynomials having high complexity.

**Lemma 4.1.** Let $\mathbb{F}$ be a field and $f_1(\mathbf{X}), \ldots, f_m(\mathbf{X}) \in \mathbb{F}[x_1, \ldots, x_n]$ be algebraically dependent polynomials, then there exists $(\alpha_1, \ldots, \alpha_m) \in \mathbb{F}^m$ such that the system of equations

$$\begin{array}{rcl} f_1(\mathbf{X}) & = & \alpha_1 \\ f_2(\mathbf{X}) & = & \alpha_2 \\ \vdots & & \vdots \\ f_m(\mathbf{X}) & = & \alpha_m \end{array} \tag{4.1}$$

has no common solution in $\mathbb{F}^n$.

*Proof.* Since the $f_i$'s are algebraically dependent, there exists a nonzero polynomial $A(z_1, \ldots, z_m) \in \mathbb{F}[z_1, \ldots, z_m]$ such that

$$A\big(f_1(\mathbf{X}), \ldots, f_m(\mathbf{X})\big) = 0. \tag{4.2}$$

Since $A(\mathbf{z})$ is nonzero, there exists $(\alpha_1, \ldots, \alpha_m) \in \mathbb{F}^m$ such that

$$A(\alpha_1, \ldots, \alpha_m) \neq 0.$$

By the Schwartz-Zippel Lemma, "most" choices of $(\alpha_1, \ldots, \alpha_m) \in \mathbb{F}^m$ will in fact have this property. It is now easily verified that with such a choice of the $\alpha_i$'s, the system (4.1) has no common solution. □

We use the lemma above to show that there exist polynomials with high arithmetic complexity. Analogous statements hold in Boolean complexity where it is known that a random Boolean function has exponential Boolean complexity.

**Theorem 4.2.** Let $\mathbb{F}$ be a field and let $d, n$ be any two natural numbers. There exists a polynomial $f(\mathbf{X}) \in \mathbb{F}[x_1, \ldots, x_n]$ of degree $d$ with

$$\mathsf{S}(f) = \Omega\left(\sqrt{\binom{n+d}{d}}\right)$$

*Proof.* Consider an arithmetic straight-line program which evaluates $f(\mathbf{X})$ using $s$ multiplications. For $i \in [k]$, let $M_i$ denote the result of the $i$-th multiplication in the program. Since the program can do only linear combinations between computing $M_i$ and $M_{i+1}$, we have

$$\begin{aligned} M_{i+1} \;=\; & \left(\beta_{i01} + \sum_{j \in [n]} \beta_{ij1} x_j + \sum_{j \in [i]} \alpha_{ij1} M_j\right) \\ & \times \left(\beta_{i02} + \sum_{j \in [n]} \beta_{ij2} x_j + \sum_{j \in [i]} \alpha_{ij2} M_j\right), \end{aligned}$$

where the $\alpha_{ijk}$'s and the $\beta_{ijk}$'s are elements from the field $\mathbb{F}$. The coefficients of the output polynomial $f(\mathbf{X})$ are therefore polynomials in the set of variables

$$V \overset{\text{def}}{=} \left\{\alpha_{ijk} : i \in [s], j \in [i], k \in [2]\right\} \bigcup \left\{\beta_{ijk} : i \in [s], j \in [0 : n], k \in [2]\right\}$$

The coefficients of the output polynomial $f(\mathbf{X})$, thought of as polynomials over $V$, are algebraically dependent when their number exceeds $|V|$. By Lemma 4.1, if $|V|$ is less than the number of coefficients then "most" coefficient vectors do not satisfy this dependence and hence the corresponding polynomials cannot be computed by circuits of size $s$.

We have $|V| = O(s^2) + O(n^2)$, while the number of coefficients of a degree $d$ polynomial in $n$ variables is $\binom{n+d}{d}$. Thus "most" $n$-variate degree $d$ polynomials $f$ have

$$\mathsf{S}(f) = \Omega\left(\sqrt{\binom{n+d}{d}}\right)$$

□

Indeed one can bound the degree of the annihilating polynomial $A$ and the absolute values of the coefficients so as to obtain an "explicit" polynomial having exponential arithmetic complexity.

**Corollary 4.3.** Any arithmetic circuit (with arbitrary, unbounded coefficients) for computing the following polynomial $f(x) \in \mathbb{C}[x]$:

$$f(x) \overset{\text{def}}{=} \sum_{i=0}^{d} 2^{2^{2i}} x^i$$

has size at least $\Omega(2^{\frac{d}{2}})$.

The real challenge however is to come up with explicit polynomials with "small" (poly($d$) bit-length) coefficients that have large arithmetic complexity. The following result of Hrubes and Yehudayoff [HY09] shows that there exist polynomials with 0-1 coefficients which have arithmetic complexity pretty much like that of a random polynomial.

**Theorem 4.4.** Let $\mathbb{F}$ be a field and let $d, n$ be any two natural numbers. Then there exists a polynomial $f(\mathbf{X}) \in \mathbb{F}[x_1, \ldots, x_n]$ of degree $d$ **with 0-1 coefficients** such that

$$\mathsf{S}(f) = \Omega\left(\sqrt{\binom{n+d}{d}}\right).$$

The proof is a much more refined version of the proof of Theorem 4.2 above. It remains a big challenge to find such polynomials explicitly.

There appears to be a quadratic gap in our understanding of upper and lower bounds for arbitrary polynomials. The argument above shows that there are $n$-variate polynomials of degree $d$ requiring arithmetic circuits with at least

$$\Omega\left(\sqrt{\binom{n+d}{d}}\right)$$

multiplication gates. On the other hand, every $n$-variate polynomial of degree $d$ can be computed by a circuit with

$$O\left(\frac{1}{n}\binom{n+d}{d}\right)$$

multiplication gates.

---

**Open Problem 4.5.** Can every $n$-variate polynomial of degree $d$ over $\mathbb{C}$ be computed by an arithmetic circuit with just

$$O\left(\sqrt{\binom{n+d}{d}}\right)$$

multiplication gates (an arbitrary number of additions and scalar multiplications with constants from $\mathbb{C}$ are allowed)?

---

Very recently, Shachar Lovett [Lov11] has made progress on this question by showing that every $n$-variate polynomial of degree $d$ over $\mathbb{C}$ can indeed be computed by an arithmetic circuit with just

$$O\left(\sqrt{\binom{n+d}{d}}\right) \cdot \mathrm{poly}(nd)$$

multiplication gates.

# 5

# Bezout's theorem

Given a set of $n$-variate polynomials of degree $d$, what is the maximum number of common zeroes that this set of polynomials can have? In this chapter, we will use the Jacobian matrix introduced in chapter 3 to give an upper bound for this number. We will follow the proof of Wooley's theorem, which works over finite fields and rings as well as algebraically closed fields.

In chapter 3, we have seen how the Jacobian matrix, representing the first-order partial derivatives of a collection of polynomials, reduces non-linear questions like algebraic independence to linear algebra. In this chapter we continue demonstrating this, giving a linear algebraic proof of Bezout's theorem.

Bezout's theorem [Har77, I.7, Thm.7.7] is one of the most basic and useful results in algebraic geometry (which we shall meet and use later in chapter 9). It bounds the number of common zeros of a system of polynomials in terms of their degrees. It is usually stated and proved in geometric terms (number of components of the associated variety), and in projective space over fields of characteristic zero. Moreover, in order to apply Bezout's theorem, one needs to ensure that there are no components of dimension one or higher for otherwise the number of common zeroes may not even be finite. Computing the dimensions of the various components is in itself a difficult algorithmic task. A useful version of the theorem was proved by Wooley [Woo96], whose proof uses only linear algebra. We consider the following setting.

Let $\mathbb{F}$ be a field. Let $\mathbf{f} = (f_1, f_2, \ldots, f_n)$ be a vector of polynomials in $\mathbb{F}[x_1, x_2, \ldots, x_n]$. Let the degree of the $i$-th polynomial $f_i$ be $d_i$. We look at the common solutions to the following system of equations:

$$f_1(\mathbf{X}) = f_2(\mathbf{X}) = \ldots = f_n(\mathbf{X}) = 0, \tag{5.1}$$

abbreviated simply as $\mathbf{f}(\mathbf{X}) = \mathbf{0}$.
A crucial definition is that of *isolated roots* of the system of equations.

**Definition 5.1.** An assignment $\mathbf{a} \in \mathbb{F}^n$ to the $n$ variables is an *isolated root* if $\mathbf{f}(\mathbf{a}) = 0$ and $\det(\mathbf{J}(\mathbf{f})(\mathbf{a})) \neq 0$.

(Note that for such roots to exist the polynomials must be algebraically independent.) With this definition, Wooley proves an analog of Bezout's theorem in finite fields $\mathbb{F}_p$ and also over finite rings $\mathbb{Z}/p^k\mathbb{Z}$. We will state and prove this theorem later. We first state Bezout's theorem in this form (using isolated roots) for the case $\mathbb{F} = \mathbb{R}$, the field of real numbers, and then adapt Wooley's proof to this simpler situation. In the reals, the concept of isolated roots has very natural geometric intuition which will guide the proof. After that, we will see what is needed to convert it to the finite field and ring setting.

**Theorem 5.2.** Assume $\mathbb{F} = \mathbb{R}$. The number of isolated roots $\mathbf{a} \in \mathbb{R}^n$ to the system $\mathbf{f}(\mathbf{X}) = \mathbf{0}$ is at most $(d_1 \cdot d_2 \cdot \ldots \cdot d_n)$.

### A Property of Isolated Roots

We now characterize isolated roots of (5.1). Intuitively, they are the roots which contain no other root in their neighborhood. How do we make this intuition precise? A nice way to formalize this intuition is that if we perturb the (free terms of the) polynomials by a very small amount then every isolated root of the original system of equations has a *unique* neighbor which is a root of the perturbed system of equations. Let us introduce some terminology. For a point $\mathbf{a} \in \mathbb{R}^n$, the $\epsilon$-neighborhood of $\mathbf{a}$ is the set of all points $\mathbf{b} \in \mathbb{R}^n$ such that $|\mathbf{b} - \mathbf{a}| < \epsilon$. We will say that an $n$-tuple of polynomials $\mathbf{g} = (g_1(\mathbf{X}), g_2(\mathbf{X}), \ldots, g_n(\mathbf{X}))$ is *a $\delta$-perturbation* of $\mathbf{f}$ if for every $i \in [n]$:

$$g_i(\mathbf{X}) - f_i(\mathbf{X}) = \delta_i,$$

for some $\delta_i \in \mathbb{R}$ with $|\delta_i| < \delta$. We need a technical lemma to justify this intuition.

**Lemma 5.3.** Let $\mathbf{a} \in \mathbb{R}^n$ be a root of the equation $\mathbf{f}(\mathbf{X}) = \mathbf{0}$. If $\mathbf{a}$ is an isolated root, then for every small enough $\epsilon > 0$ there exists a $\delta$ (depending on $\epsilon$) such that for any $\delta$-perturbation $\mathbf{g}$ of $\mathbf{f}$, there is a unique root $\mathbf{b}$ of $\mathbf{g}(\mathbf{X}) = \mathbf{0}$ in the $\epsilon$-neighborhood of $\mathbf{a}$.

We first show how to use this lemma to prove Bezout's theorem while deferring its proof to a later point.

### Proof of Theorem 5.2

We will denote by $A_{\mathbf{f}} \subset \mathbb{R}^n$ the set of isolated solutions to (5.1) and by $N_{\mathbf{f}}$ the cardinality of this set. We denote by $D$ the integer $\prod_{i\in[n]} d_i$.

**Step One: Reduction to counting only the projection of $A_{\mathbf{f}}$ on the first coordinate.** Replace $x_1$ by a random linear combination of all the $n$ variables. This ensures that if we had more than $D$ solutions, we will get more than $D$ projections which will lead to a contradiction.

**Step Two: Using an elimination polynomial.**
The idea is the following. We look for a nonzero polynomial

$$H(y_1, y_2, \ldots, y_n, z)$$

with the following properties:

(P1). $H(f_1, f_2, \ldots, f_n, x_1) = 0$ identically.

(P2). The degree of $z$ in $H$ is small (say $B$).

(P3). $h(x_1) = H(0, 0, \ldots, 0, x_1)$ is a nonzero polynomial.

The idea is that every $\mathbf{a} = (a_1, a_2, \ldots a_n) \in A_{\mathbf{f}}$ must satisfy $h(a_1) = 0$ so that $N_{\mathbf{f}} \leq B$. We will be able to show that such a polynomial $H$ satisfying properties (P1) and (P2) exists with $B = D$. However, if (P3) is not satisfied we can replace it with:

(P3′). $h(x_1) = H(\delta_1, \delta_2, \ldots, \delta_n, x_1)$ is a nonzero polynomial, for $\delta_i$'s being 'very small'.

This happens for some choice of $\delta_i$'s — indeed very small random ones will do since $H$ is nonzero. By lemma 5.3, every isolated solution $\mathbf{a}$ of $\mathbf{f} = \mathbf{0}$ has a unique neighbour $\mathbf{b} \in \mathbb{R}^n$ such that

$$\mathbf{f}(\mathbf{b}) = (\delta_1, \delta_2, \ldots, \delta_n). \tag{5.2}$$

So we have

$$\begin{aligned} N_{\mathbf{f}} &\leq \text{Number of solutions of (5.2)} \\ &\leq \deg_{x_1}(H(\delta_1, \ldots, \delta_n, x_1)) \\ &= B \\ &= D \end{aligned}$$

**Step Three: Constructing the elimination polynomial.**
Consider the vector space $V$ of all polynomials in $\mathbb{R}[y_1, \ldots, y_n, z]$ with the degree with respect to $z$ being at most $B$, such that when evaluated at $(f_1, \ldots, f_n, x_1)$ have total degree at most $K$ in the $x_i$'s. Now consider an arbitrary polynomial $H$ in $V$. $H(f_1, \ldots, f_n, x_1)$ will be a polynomial in the $x_i$'s of degree at most $K$ so that in general $H$ will have

$$M_0 := \binom{n+K}{n}$$

monomials whose coefficients are homogeneous linear combinations of the undetermined coefficients of $H$. This means that once the dimension of $V$ exceeds $M_0$, we are guaranteed that a nonzero $H$ in $V$ exists such that $H(f_1, \ldots, f_n, x_1) = 0$. The dimension of $V$ is the number of monomials in $y_1, \ldots, y_n, z$ whose appropriate weighted degree is at most $K$. A monomial $y_1^{k_1} \cdot y_2^{k_2} \cdot \ldots y_n^{k_n} \cdot z^b$ is in $V$ if $b \leq B$ and

$$d_1 \cdot k_1 + d_2 \cdot k_2 + \ldots + d_n \cdot k_n + b \leq K. \tag{5.3}$$

How many such monomials are there? We use the following estimate from below.

**Fact 5.4.** The number of tuples $(k_1, k_2, \ldots, k_n) \in \mathbb{Z}^n_{\geq 0}$ satisfying (5.3) is at least

$$\frac{1}{d_1 \cdot d_2 \cdot \ldots \cdot d_n} \binom{n + K - b}{n}.$$

Summing over $b \in \{0, 1, \ldots, B\}$ and taking $K$ large enough, the dimension of $V$ is

$$\left[\frac{B+1}{D}\right] M_0 \left(1 - O\left(\frac{Bn}{K}\right)\right),$$

which for large $K$ is greater than $M_0$ for the choice $B = D$. So we have $H$ satisfying (P1) and (P2).

To guarantee (P3) or $(P3')$, we first note that if we take $H$ to be a polynomial of the lowest degree that satisfies (P1) and (P2), then it must depend on the variable $z$. This argument resembles ones we have seen in chapter 3 on algebraic independence. Assume $H$ does not depend on $z$, then $\frac{\partial H}{\partial z} = 0$. Now differentiating the identity

$$H(f_1, f_2, \ldots, f_n, x_1) = 0$$

with respect to each variable $x_i$, we have

$$\mathbf{J} \cdot \begin{pmatrix} \partial_1 H(\mathbf{f}, x_1) \\ \partial_2 H(\mathbf{f}, x_1) \\ \vdots \\ \partial_n H(\mathbf{f}, x_1) \end{pmatrix} = 0,$$

where $\partial_i H$ is a shorthand for $\frac{\partial H}{\partial y_i}$. Since $\mathbf{J}(\mathbf{X})$ is nonsingular we have $\partial_i H(\mathbf{f}, x_1) = 0$ for each $i \in [n]$. One of the $(\partial_i H)$'s must be a nonzero polynomial so that we get a polynomial of lower degree satisfying (P1) and (P2), a contradiction.

Thus $H$ depends on $z$ and so must have the form

$$H = H_0(\mathbf{y}) + H_1(\mathbf{y}) \cdot z + \ldots + H_r(\mathbf{y}) \cdot z^r,$$

with $H_r(\mathbf{y})$ nonzero. Finally, we pick a set of "very small" $\delta_i$'s satisfying

$$H_r(\delta_1, \delta_2, \ldots, \delta_n) \neq 0.$$

This ensures that either (P3) or $(P3')$ is also satisfied. This completes the proof of Theorem 5.2.

**Proof of lemma 5.3.**

To complete the presentation, we now give the proof of lemma 5.3. A simpler proof can be found in [Kud01] using the implicit function theorem, but here we present a more direct (but longer) proof. Intuitively, the nonvanishing of the Jacobian on the root $\mathbf{a}$ means that $\mathbf{f}$ can be well approximated by a linear map in a small $\epsilon$-neighborhood of $\mathbf{a}$. If $\mathbf{f}$ were indeed a linear map, it would be clear how to change $\mathbf{a}$ to a root $\mathbf{b}$ of the perturbed system $\mathbf{g}$. In the general case, the same change only gets us closer to the unique root, and so iterating it converges to the unique nearby root $\mathbf{b}$ of $\mathbf{g}$. This is Newton's iteration, an iterative process which we now describe formally.

We start from $\mathbf{b}_0 = \mathbf{a}$ and converges to $\mathbf{b}$. For the iterative step let

$$\mathbf{b}_{i+1} \overset{\text{def}}{=} \mathbf{b}_i - \mathbf{J}_{\mathbf{g}}(\mathbf{b}_i)^{-1} \cdot \mathbf{g}(\mathbf{b}_i). \tag{5.4}$$

The unique point in $\mathbb{R}^n$ to which this iterative process converges is the unique neighbor of $\mathbf{a}$ which is an isolated zero of the perturbed system $\mathbf{g}(\mathbf{X}) = \mathbf{0}$. Let us first make sure that the iterative step is well-defined, i.e. $\mathbf{J}_{\mathbf{g}}(\mathbf{b}_i)^{-1}$ exists for each $i$. For the starting point:

$$\mathbf{J}_{\mathbf{g}}(\mathbf{b}_0)^{-1}$$

exists because

$$\begin{aligned} \Delta &:= \det(\mathbf{J}_{\mathbf{g}}(\mathbf{b}_0)) \\ &= \det(\mathbf{J}_{\mathbf{f}}(\mathbf{a})) \\ &\neq 0 \end{aligned}$$

By continuity, there is a $\epsilon$-neighborhood of $\mathbf{a}$ within which the Jacobian determinant is between say $\frac{\Delta}{2}$ and $2\Delta$. We will choose $\delta$ small enough to make sure that the $\mathbf{b}_i$'s that we generate in this iterative process stay within the $\epsilon$-neighborhood of $\mathbf{a}$. This will ensure that $\mathbf{J}_{\mathbf{g}}(\mathbf{b}_i)^{-1}$ exists so that the iteration is well-defined. It now suffices to prove the following claim:

**Claim 5.5. Convergence of Newton Iteration** There exist constants $C_0, C_1, C_2$ such that for all $\delta < C_0$ and for all $i \geq 0$ we have:

(1) $|\mathbf{b}_{i+1} - \mathbf{b}_i| < C_1 \cdot \delta^{i+1}$ and

(2) $|\mathbf{g}(\mathbf{b}_i)| < C_2 \cdot \delta^{i+1}$ and

(3) $|\mathbf{b}_{i+1} - \mathbf{a}| < \epsilon$

**Proof of Claim** 5.5: We prove this claim by induction. We prove it only for the case $n = 1$. The proof below generalizes in a very natural way for higher values of $n$. We specify the choices of the constants $C_0, C_1$ and $C_2$ — they all depend on another constant $C_3$ which bounds the size of the error term in the Taylor expansion of $\mathbf{f}$ around $\mathbf{a}$ (truncating after the linear part).

$$\begin{aligned} C_0 &:= \min\left(\frac{1}{4}, \epsilon, \frac{2C_3}{\Delta}\epsilon\right) \\ C_1 &:= \frac{\Delta}{4C_3} \\ C_2 &:= \frac{\Delta^2}{8C_3} \end{aligned}$$

where

$$C_3 := \max_{k\geq 2} \max_{|\mathbf{y}-\mathbf{a}|\leq\epsilon} \left|\frac{\partial^k g(\mathbf{y}-\mathbf{a})}{k!}\right|.$$

We leave it to the reader to verify that with the choice of parameters above, the claim can be proved by induction on $i$. □

### Bezout's theorem over finite fields and rings

Observe that the same proof can essentially be carried out over some other fields of characteristic zero such as $\mathbb{Q}$ and $\mathbb{C}$. Indeed, as long as we have a norm on our field which satisfies the natural conditions (like subadditivity under addition and multiplication) it defines a metric on the field that enables the proof to go through. It may seem that concepts of distance, $\epsilon$-neighborhood and convergence are meaningless for finite field $\mathbb{F}_p$. Remarkably, there is a way to mimic the proof above over $\mathbb{F}_p$ and even rings $\mathbb{Z}/p^k\mathbb{Z}$. The appropriate "p-adic" norm and metric was developed by Hensel to enable analytic techniques in number theoretic problems. In particular, we will see the "Hensel lifting" technique which allows to perform the analog of "Newton iteration" in this setting. All polynomials we deal with in this subsection will have integer coefficients. This is convenient as integer coefficients can be taken modulo $p$ or modulo $p^k$ and we get the associated polynomials over these finite fields and rings. We note however that even when working modulo $p^k$ for $k > 1$ the notion of isolated roots will require that the Jacobian does not vanish modulo $p$.

We first state Wooley's theorem.

**Theorem 5.6.** Let $f_1(\mathbf{X}), \ldots, f_n(\mathbf{X}) \in \mathbb{Z}[x_1, \ldots, x_n]$ with $\deg(f_i) = d_i$. Let $p$ be a prime, and $k$ be any integer. The number of solutions with entries in $\{0, 1, \cdots, p^k - 1\}$ to

$$\mathbf{f}(\mathbf{X}) = \mathbf{0} \pmod{p^k} \quad \text{subject to} \quad \det(\mathbf{J})(\mathbf{X}) \neq 0 \pmod{p}$$

is at most $D = \prod_i d_i$.

We outline the main ideas involved in this process of translating the proof over the reals to this setting. It would be interesting to describe the proof even in the special case $k = 1$, namely when we seek to solve the system in the finite field $\mathbb{F}_p$. But even for this special case, as we shall see, we will need to pass through residues modulo $p^k$ for all $k$, including $k = \infty$. In other words, we will need the *p-adic numbers*, to which we first give a brief introduction.

### A brief introduction to the $p$-adic numbers, norm and metric

If $p$ is a fixed prime number, then any positive integer can be written uniquely in a base $p$ expansion in the form

$$\sum_{i=0}^{n} a_i p^i, \tag{5.5}$$

where the $a_i$'s are integers in $\{0, 1, 2, \ldots, p-1\}$. If we now consider *formal infinite* sums of the form

$$\sum_{i=0}^{\infty} a_i p^i, \quad \text{where each } a_i \in \{0, 1, \ldots, p-1\} \tag{5.6}$$

we obtain the $p$-adic integers that we denote by $\mathbb{Z}_p$. In other words $\mathbb{Z}_p$ is the set of all infinite formal sums of the form (5.6) above. Intuitively, addition and multiplication of elements are carried out in the same as we would have if the sequences were finite (treating $p$ as a formal variable). More concretely the first $k$ terms of the sum (respectively the product) of two sequences

$$\sum_{i=0}^{\infty} a_i p^i \quad \text{and} \quad \sum_{i=0}^{\infty} b_i p^i$$

is obtained as follows: truncate the two sequences to their first $k$ terms to obtain two integers

$$A = \sum_{i=0}^{k-1} a_i p^i \quad \text{and} \quad B = \sum_{i=0}^{k-1} b_i p^i.$$

The first $k$ terms of the sum (respectively the product) are the $k$ least significant 'digits' in the base-$p$ expansion of $A+B$ (respectively $A \cdot B$). Notice that finite sequences of the form (5.5) do not have additive inverses because they correspond to positive integers. But once we go to infinite sequences of the form (5.6), additive inverses appear and the set $\mathbb{Z}_p$ forms a group under addition. For example

$$-1 = \sum_{i=0}^{\infty} (p-1) \cdot p^i.$$

The set $\mathbb{Z}_p$ is in fact *almost* a field: if $a_0$ is different from 0 then

$$\sum_{i \geq 0} a_i p^i$$

has an inverse in $\mathbb{Z}_p$ as well. For example for $p = 3$,

$$2^{-1} = 2 + 1 \cdot 3 + 1 \cdot 3^2 + 1 \cdot 3^3 + \ldots$$

However $\mathbb{Z}_p$ is not a field. For example the inverse of the element $p = 0 + 1 \cdot p + 0 \cdot p^2 + 0 \cdot p^3 + \ldots$ does not exist in $\mathbb{Z}_p$. Luckily this can be fixed by slightly extending the $\mathbb{Z}_p$. We consider all infinite sequences of the form

$$\sum_{i=k}^{\infty} a_i p^i, \tag{5.7}$$

where $k$ is some (not necessarily positive) integer and as before each $a_i$ is in $\{0, 1, 2, \ldots, p-1\}$. [1] The set of formal sums of the form (5.7) we denote by $\mathbb{Q}_p$. $\mathbb{Q}_p$ forms a field and we call it the field of $p$-adic numbers. One very useful property of the $p$-adics is that it comes equipped with a useful notion of *absolute value* of an element (an absolute value for $\mathbb{Q}_p$ is a map from $\mathbb{Q}_p$ to $\mathbb{R}$). Let

$$\alpha = \sum_{i=k}^{\infty} a_i p^i \tag{5.8}$$

be an arbitary element in $\mathbb{Q}_p$. Let the *smallest* $i$ for which $a_i$ is nonzero be $t$. Then the $p$-adic absolute value of $\alpha$, denoted $|\alpha|_p$, is defined to be the real number $p^{-t}$. Finally, this metric satisfies the following useful properties: for all $a, b, c \in \mathbb{Q}_p$

$$\begin{aligned} |a+b|_p &\leq \max(|a|_p, |b|_p) && (5.9) \\ |a \cdot b|_p &= |a|_p \cdot |b|_p && (5.10) \\ |a-b|_p &\leq |a-c| + |c-b| \quad \text{(triangle inequality)} && (5.11) \end{aligned}$$

### Sketch of proof of theorem 5.6

With this description of the $p$-adic numbers in place we proceed to outline a proof of theorem 5.6. Consider a polynomial $f(\mathbf{X})$ with integer coefficients. Note that the integers form a subset of $\mathbb{Z}_p$ which is in turn a subset of $\mathbb{Q}_p$ so that $f(\mathbf{X})$, which has integer coefficients can also be thought of as a polynomial with coefficients in $\mathbb{Q}_p$. At the same time, by reducing each coefficient of $f(\mathbf{X})$ modulo $p$, we can also view $f(\mathbf{X})$ as a polynomial over the finite field $\mathbb{Z}/(p\mathbb{Z}) = \mathbb{F}_p$. In this way we will think

[1] For those aware of the terminology: $\mathbb{Z}_p$ is an integral domain, i.e. if the product of two elements of $\mathbb{Z}_p$ is zero then each of those elements must be zero. $\mathbb{Q}_p$ can alternatively be obtained as the field of fractions of this integral domain.

of polynomials with integer coefficients simultaneously as polynomials over $\mathbb{Z}$, $\mathbb{Q}_p$ and $\mathbb{F}_p$. The proof involves two steps. In the first step, we will see that every isolated root of $\mathbf{f}(\mathbf{X}) = \mathbf{0}$ over $\mathbb{F}_p$ can be *lifted up* to a unique isolated root over $\mathbb{Q}_p$. In the second step we will see that theorem 5.2 holds true with $\mathbb{F} = \mathbb{Q}_p$, the field of $p$-adic numbers, i.e. we obtain an upper bound on the number of isolated roots over $\mathbb{Q}_p$. Theorem 5.6 then follows.

**The first step: Lifting solutions modulo $p$ to solutions in $\mathbb{Q}_p$.** It is based on the following lemma.

**Lemma 5.7. Hensel Lifting.** Assume that for some $\mathbf{a} \in \mathbb{Z}^n$

(1) $\mathbf{f}(\mathbf{a}) = \mathbf{0} \pmod{p^k}$.

(2) $\det(\mathbf{J_f})(\mathbf{a}) \neq 0 \pmod{p}$.

Then for every $m > k$, there is a unique $\mathbf{b} \in (\mathbb{Z}/p^m\mathbb{Z})^n$ such that

(1) $\mathbf{b} = \mathbf{a} \pmod{p^k}$.

(2) $\mathbf{f}(\mathbf{b}) = \mathbf{0} \pmod{p^m}$.

The proof is a modification of Newton iteration, done in the $p$-adic metric. More concretely, this linearization process (say for $m = k + 1$, which can then be iterated for larger $m$) is made explicit as follows. For polynomial maps the "Taylor expansion" is a polynomial identity, and truncating it after the first term gives

$$\mathbf{f}(\mathbf{b}) = \mathbf{f}(\mathbf{a}) + \mathbf{J}_{(\mathbf{f})}(\mathbf{a}) \cdot (\mathbf{b} - \mathbf{a}) \pmod{p^{k+1}}.$$

In doing this truncation we use the fact that

$$\mathbf{b} - \mathbf{a} = \mathbf{0} \pmod{p^k}$$

so that

$$(\mathbf{b} - \mathbf{a})^2 = \mathbf{0} \pmod{p^{k+1}}$$

and therefore, the higher order terms in the Taylor expansion vanish modulo $p^{k+1}$. Then

$$\mathbf{b} = \mathbf{a} - \mathbf{J_f}(\mathbf{a})^{-1} \cdot \mathbf{f}(\mathbf{a}) \pmod{p^{k+1}}$$

satisfies the properties claimed in the lemma. Now if we take $m = \infty$ in the lemma above, the sequence of $\mathbf{b}$'s so obtained "converges" in the $p$-adic metric to some unique point in $\mathbb{Z}_p^n$. In other words, we get that corresponding to every isolated root $\mathbf{a} \in \mathbb{F}_p$ there is a unique $\mathbf{b} \in \mathbb{Z}_p^n$, the ring of $p$-adic integers such that $\mathbf{b}$ is an isolated solution of $\mathbf{f} = \mathbf{0}$.

**The second step: Proving Bezout over $\mathbb{Q}_p$.**
This step mimics the proof over the reals. We give the key facts one needs to do this mimicry. $\mathbb{Q}_p$ is a field and hence the number of roots of any univariate polynomial is bounded by its degree. Furthermore, $\mathbb{Q}_p$ comes equipped with the $p$-adic metric $|\cdot|_p$ mentioned above, which allows us to talk about the neighborhood of a point. These properties of $\mathbb{Q}_p$ and its accompanying "valuation" suffice to mimic the proof over reals in a relatively straightforward way. We omit the details and leave the proof of theorem 5.6 as an exercise for the interested reader .

This completes our description of Wooley's proof of Bezout's theorem.

# 6

# Algebraic extractors and the Jacobian conjecture

A collection of $n$ polynomials $f_1, \ldots, f_n$ in $n$ variables over the field $\mathbb{F}$ can be viewed as a map $\Psi$ from the vector space $\mathbb{F}^n$ to itself in the natural way:

$$\Psi : (x_1, \ldots, x_n) \mapsto (f_1, \ldots, f_n).$$

A fundamental question is: is the map $\Psi$ a bijection? We will describe a well-known open problem in algebra called the Jacobian conjecture which seeks to characterize such bijective maps. It involves the Jacobian matrix introduced in chapter 3. Before that we outline some recent results on how to condense the output of a map while retaining the algebraic rank.

The material in the first part of this chapter is taken from Dvir, Gabizon and Wigderson [DGW09]. The paper deals with the extraction of randomness from sources sampled by polynomial maps over large finite fields. However, as a preliminary result of independent interest for us here we describe the so-called "rank-extractors", which is a natural object from an algebraic standpoint.

Consider $k$ polynomials $\mathbf{f} = (f_1, f_2, \ldots, f_k)$ in $\mathbb{F}[x_1, x_2, \ldots, x_n]$ each of degree at most $d$, and assume that the rank of the Jacobian $\mathbf{J}(\mathbf{f})$ is $r$ (note that $r$ is at most the minimum of $k$ and $n$). So there are some $r$ algebraically independent polynomials among the $f_i$'s. Is there a general way to "extract" this much algebraic independence from them? This question is formalized as follows.

**Definition 6.1.** A vector of polynomials $\mathbf{g} = (g_1, g_2, \ldots, g_r)$ in $\mathbb{F}[z_1, z_2, \ldots, z_k]$ is a $(k, n, d)$-*rank extractor*, if for *every* system $\mathbf{f}$ as above, the composition $\mathbf{g} \circ \mathbf{f}$ are algebraically independent. A rank extractor $g$ is explicit if all $g_i$ have arithmetic circuit of size polynomial in $k, n, d$, and their degree bounded by a polynomial in $k, n, d$.

**Theorem 6.2.** For every $k, n, d$, there is an explicit $\mathbf{g}$ which is a $(k, n, d)$-rank extractor for *every* field $\mathbb{F}$ of characteristic zero or larger than $(knd)^3$.

**The Construction.** We give the construction of the rank extractor here while omitting the proof of its correctness. Let $s_2 = dn + 1$ and $s_1 = (2dk + 1)s_2$. Let $l_{ij} = i \cdot (s_1 + js_2)$. Define for each $1 \leq i \leq r$

$$g_i(z_1, \ldots, z_k) := \sum_{j \in [k]} \frac{1}{l_{ij} + 1} z_j^{l_{ij}+1}$$

The proof of correctness of the construction above by Dvir, Gabizon and Wigderson [DGW09] heavily uses the matroid structure of algebraic independence and the chain rule for partial derivatives underlying the Jacobian, which we met in Chapter 3. These rank-extractors serve as a starting point for the construction of randomness extractors for polynomial sources, and Wooley's theorem of the previous chapter guarantees that such sources have high enough "min-entropy".

## 6.1 The Jacobian conjecture

Here we describe (only for dimension 2) the famous Jacobian conjecture which attempts to characterize bijective polynomial maps over the complex numbers.

Let us consider a pair of bivariate polynomials: $f(x, y)$ and $g(x, y) \in \mathbb{C}[x, y]$. This pair of polynomials naturally represents a map $\phi : \mathbb{C}^2 \mapsto \mathbb{C}^2$, with $(x, y) \mapsto (f(x, y), g(x, y))$. The question that we are interested in is whether this map is a bijection or not.

The following conjecture (generalized to $n$ polynomials in $n$ variables, known as the Jacobian conjecture), says that the bijectivity of this map $\phi$ is captured by the determinant of the Jacobian matrix.

**Conjecture 6.3.** The map $(x, y) \mapsto (f(x, y), g(x, y))$ is a bijection if and only if

$$\textsc{Det} \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{pmatrix} = c,$$

for some nonzero constant $c \in \mathbb{C}$.

[Kel39, Wri81, Yu95] is a partial list of references on this conjecture.

# 7

# The "Joints conjecture" resolved

Combinatorial geometry studies configurations of points and lines (and other geometric objects) in space satisfying certain properties. How do we reason about such a collection of points and lines? A simple and useful idea is to interpolate a polynomial through the set of points! In this chapter, we will see an application of an extension of this idea wherein partial derivatives of the interpolated polynomial will help us prove the Joints conjecture.

Here we briefly describe an application of polynomials and their partial derivatives to a problem in combinatorial geometry. This is only one instance of a recent slew of results of this type, starting with Dvir's resolution of the finite-field Kakeya problem [Dvi09b]. A survey of these results is in [Dvi09a].

We will work over Euclidean space $\mathbb{R}^d$, although with some care all definitions and results apply over finite fields as well. We shall think of $d$ as fixed. Now assume we have a set $L$ of lines in this space. A *joint* is the intersection of $d$ lines from $L$, of *linearly independent* directions.

What is the largest number of joints that can be spanned by $n$ lines? The regular lattice in $d$ dimensions, with side about $n^{1/(d-1)}$, shows that we can have $\Omega(n^{d/(d-1)})$ joints. The "Joints Conjecture" was that this is asymptotically tight. The conjecture was open for many years, even for dimension 3. This case was finally resolved by Guth and Katz [GK08], and their proof was simplified and extended to all dimensions by Kaplan, Sharir and Shustin [KSS10].

**Theorem 7.1.** *$n$ lines in $\mathbb{R}^d$ span at most $O((n^{d/(d-1)})$ joints (the implied constant depends only on $d$).*

*Proof.* Let $L$ be the set of $n$ lines, and $P$ be the set of joints they span. Without loss of generality we can assume that every line passes through at least $|P|/2n$ joints (otherwise we can remove the other lines, losing only at most half the joints). Now assume for contradiction that $|P| \geq Cn^{d/(d-1)}$ for a constant $C$ to be determined later. Let $P'$ be a subset of $P$ of cardinality exactly $|P'| = Cn^{d/(d-1)}$.

Let $f$ be a *non-zero* $d$-variate polynomial of *minimal degree* passing through all joints in $P'$. Note that the degree $r$ of $f$ can be chosen to be $O(n^{1/(d-1)})$ (This is by standard interpolation — we have enough degrees of freedom in a linear system of equations determining the coefficients of $f$). The constant $C$ is chosen such that $r < |P'|/2n$. The choice of $C$ and the assumption that every line passes through more than $r$ joints ensure that $f$ vanishes *identically* on every line in $L$.

Now consider the gradient $\partial(f)$, the $d$-dimensional vector of first-order partial derivatives $\partial_{x_i}(f)$. Since $f$ is nonzero, one of these derivatives, say $g = \partial_{x_1}(f)$ is nonzero as well, and has degree lower than $f$. The contradiction will follow by proving below that *all* these partial derivatives pass through all joints in $P'$ as well.

Consider evaluating $\partial(f)$ at any joint $p \in P'$. Let $v_1, v_2, \ldots, v_d$ be the $d$ *linearly independent* directions of the lines defining $p$ as a joint. Since $f$ vanishes on these lines, we have that the vector $\partial(f)(p)$ has inner product zero with all $v_i$. But since the $v_i$'s span $\mathbb{R}^d$, $\partial(f)(p)$ must be identically zero, namely all partials vanish on $p$. But this argument holds for all $p \in P'$ and we are done. □

# 8

# The Stepanov method

A univariate polynomial of degree $d$ over any field has at most $d$ zeroes. Now consider a bivariate polynomial $f(x, y)$ over a finite field $\mathbb{F}_p$. How many zeroes can it have? In other words, how many points $(a, b)$ are there in $\mathbb{F}_p \times \mathbb{F}_p$ such that $f(a, b) = 0$? In this chapter we will see how derivatives are useful in addressing this question. We will follow the "Stepanov method", and illustrate its power for such bounds and related exponential sums, of Weil, Heath-Brown and Mit'kin.

This chapter will have four parts. In Section 8.1, we give the background for one of the important achievements of algebraic geometry — Weil's bound on the number of rational points on curves, namely bivariate polynomials. We will explain in a special case (relevant to the commonly used Weil's exponential sum bound) the reduction to bounding the number of roots of univariate polynomials. This will motivate the first problem to be attacked by Stepanov's method: how many times a polynomial $h$ of the form $h = f^n$ can attain any particular value. The challenge is getting a bound which is *much* smaller than the degree of $h$.

In Section 8.2 we will first explain the intuition behind Stepanov's method; the use of an auxiliary polynomial which vanishes to large multiplicity on the roots of $h-a$, and the special differential structure of $h$'s for which this method is effective.

We then give the details of how the method applies to polynomials of the form $h = f^n$ in Section 8.3. In Section 8.4 we explain a result of Heath-Brown and Mit'kin, bounding the number of roots of a "transcendental-looking" polynomial, which follows the same recipe, but requires completely different choices of parameters and uses other properties of the given polynomial.

## 8.1 Weil's theorem on rational points on curves

Let $p$ be a prime, and $g(x,y) \in \mathbb{F}_p[x,y]$ be a bivariate polynomial of degree $d$. Denote by $A_g \subseteq \mathbb{F}_p \times \mathbb{F}_p$ the set of $\mathbb{F}_p$-zeroes of the polynomial $g(x,y) = 0$, i.e.

$$\big\{(a,b) \in \mathbb{F}_p \times \mathbb{F}_p : g(a,b) = 0\big\},$$

The set $A_g$ is also called the set of $\mathbb{F}_p$-*rational points* of the polynomial $g$. We tackle the following question: *what is the size of the set* $A_g$? When $p$ is much larger than $d$, say $p = \Omega(d^5)$, very good estimates for this quantity are known. Let the factorization of $g(x,y)$ over $\mathbb{F}_p$ be

$$g(x,y) = g_1(x,y)^{e_1} \cdot g_2(x,y)^{e_2} \cdot \ldots \cdot g_k(x,y)^{e_k}.$$

To answer the question posed above, we need a definition. Let us call a polynomial in $\mathbb{F}_p[x,y]$ *absolutely irreducible* if it is irreducible over the algebraic closure $\overline{\mathbb{F}}_p$ of $\mathbb{F}_p$.

**Example 8.1.** For example, $(y^2 - x^3) \in \mathbb{F}_7[x,y]$ is absolutely irreducible whereas $(y^2 + x^2) \in \mathbb{F}_7[x,y]$ is irreducible over $\mathbb{F}_7$ but factors into $(y + \sqrt{-1}x)(y - \sqrt{-1}x)$ over the extension $\mathbb{F}_{7^2} = \mathbb{F}_7(\sqrt{-1})$ and hence is not absolutely irreducible over $\mathbb{F}_7$.

Going back to the question posed above, let the number of distinct *absolutely irreducible* factors of $g(x,y)$ be $t$. Then the number of rational points is approximately $p \cdot t$. More precisely, this number $|A_g|$ satisfies the following upper and lower bounds:

$$pt - O\left(d^2\sqrt{p}\right) \leq |A_g| \leq pt + O\left(d^2\sqrt{p}\right) \qquad (8.1)$$

This chapter is devoted to presenting a proof of the above bound for a certain illustrative special case. We refer the reader to Schmidt's book [Sch04] for the general case. Our presentation is also based on this text.

**Historical Notes.** The bound given by (8.1) was first conjectured by Emil Artin in 1924.[1] In 1948, Andre Weil published the proof of Artin's conjecture. This was a landmark in both number theory and algebraic geometry. Subsequently, in 1949, Weil proposed far reaching generalizations of Artin's conjecture. Later, Grothendieck and Deligne employed profound innovations in algebraic geometry to carry Weil's work much further.

In what came as a surpise to the mathematics community, Sergei Stepanov gave elementary proofs of many of Weil's most significant results in a series of papers published between 1969 and 1974. Proofs of Weil's result in full generality, based on Stepanov's ideas were given independently by Wolfgang Schmidt and Enrico Bombieri in 1973. The text by Schmidt [Sch04] contains a very nice exposition of both methods. The presentation here is based on the first chapter of Schmidt's book.

In the next section, we first show that in order to prove (8.1), it suffices to prove the appropriate bound *for a single absolutely irreducible polynomial.* We do not prove the bound (8.1) for all absolutely irreducible polynomials but rather just for polynomials of the form

$$g(x, y) = y^r - f(x),$$

with $r$ being a prime not dividing $\deg(f(x))$.

### Reduction to the case of a single absolutely irreducible polynomial

Let the factorization of $g(x, y)$ over $\mathbb{F}_p$ be

$$g(x, y) = g_1(x, y)^{e_1} \cdot g_2(x, y)^{e_2} \cdot \ldots \cdot g_k(x, y)^{e_k}$$

[1] The conjecture of Emil Artin was formulated somewhat differently and is a little stronger than what is given here.

Clearly, we can assume without loss of generality that the $e_i$'s are all 1 (as this does not change $A_g$ and decreases the degree) i.e.

$$g(x,y) = g_1(x,y) \cdot g_2(x,y) \cdot \ldots \cdot g_k(x,y),$$

where the $g_i(x,y)$'s are distinct $\mathbb{F}_p$-irreducible polynomials. Let us also assume without loss of generality that the first $t$ factors: $g_1, g_2, \ldots, g_t$ are absolutely irreducible while the rest are not. Let $A_{g_i}$ be the set of $\mathbb{F}_p$-rational points of $g_i(x,y)$. By the inclusion-exclusion principle,

$$\sum_{1 \le i \le k} |A_{g_i}| - \sum_{1 \le i < j \le k} \left| A_{g_i} \bigcap A_{g_j} \right| \le |A_g| \le \sum_{1 \le i \le k} |A_{g_i}| \tag{8.2}$$

Any point $P = (a,b)$ in $A_{g_i} \bigcap A_{g_j}$ is a multiple point on the curve $g(x,y) = 0$ and therefore simultaneously satisfies the equations

$$g(x,y) = \frac{\partial g}{\partial x} = \frac{\partial g}{\partial y} = 0. \tag{8.3}$$

Furthermore it can be shown that if $g_i(x,y)$ is $\mathbb{F}_p$-irreducible but not absolutely irreducible then any $\mathbb{F}_p$-rational point on $g_i(x,y)$ is also a multiple point of the curve $g(x,y) = 0$ and therefore satisfies equation (8.3) as well. We leave the proof of this as an exercise for the reader. We now use the squarefreeness of $g$ to upper bound the number of multiple points on the curve $g(x,y) = 0$ by $O(d^2)$.

**Lemma 8.2.** The number of solutions to (8.3) is at most $d^2$.

*Proof.* By making a suitable invertible linear transformation of the variables if necessary, we can assume without loss of generality that the leading term with respect to $x$ of the polynomial $g(x,y)$ is an element in $\mathbb{F}_p^*$. Then since $g(x,y)$ is squarefree, we have that $\gcd(g, \frac{\partial g}{\partial x}) = 1$. We now apply Bezout's theorem to $g(x,y)$ and $\frac{\partial g}{\partial x}$. Bezout's theorem states that the number of common zeroes of two coprime polyomials is at most the product of the degrees of the two polynomials. Applying Bezout's theorem to $g(x,y)$ and $\frac{\partial g}{\partial x}$, we get that the number of common zeroes and hence also the number of multiple points of $g$ is at most $d(d-1) < d^2$. □

Thus, we can now refine (8.2) to obtain:

$$\sum_{1\le i\le t} |A_{g_i}| - O(d^2) \le |A_g| \le \sum_{1\le i\le t} |A_{g_i}| + O(d^2) \qquad (8.4)$$

Note that in this estimate we are summing up only over the absolutely irreducible factors of $g$. Thus in order to prove the bound (8.1), it suffices to prove the following:

---

**Theorem 8.3.** (**Weil's Theorem**, also known as "**Riemann Hypothesis for curves over finite fields**".) Let $g(x, y) \in \mathbb{F}_p[x, y]$ be an absolutely irreducible polynomial of degree $d$. Then the number $|A_g|$ of $\mathbb{F}_p$-rational points on the curve $g(x, y)$ satisfies

$$\big||A_g| - p\big| \le O(d^2)\sqrt{p}.$$

---

If the $i$-th absolutely irreducible polynomial $g_i(x, y)$ has degree $d_i$, we use the fact that $\sum_i d_i^2 \le (\sum_i d_i)^2$ to derive the estimate (8.1) from the above theorem.

We now discuss Stepanov's proof of a significant special case of this theorem — when the polynomial $g(x, y)$ is of the form $y^r - f(x)$, $r$ a prime not dividing $\deg(f)$. Here we also sacrifice some accuracy in order to improve clarity — the "error term" in our proof shall be $O(d^{\frac{7}{2}})\sqrt{p}$ instead of $O(d^2)\sqrt{p}$ as stated above.

### Preliminary Observations

For the rest of this chapter we will fix $g(x, y)$ to be $y^r - f(x)$. All the proofs presented here have a relatively straightforward generalization even if $r$ is not prime but so long as $\gcd(r, \deg(f(x))) = 1$. For the sake of ease of presentation we deal only with the case that $r$ is a prime and does not divide $\deg(f)$. For a univariate polynomial $f(x)$ we will denote its (first) derivative simply by $f'$ and denote its $\ell$-th order derivative, namely $\frac{d^\ell f}{dx^\ell}$, by $f^{(\ell)}$. We begin with a proposition assuring us that in our situation, $y^r - f(x)$ is absolutely irreducible.

**Proposition 8.4.** The bivariate polynomial $g(x, y) = y^r - f(x)$ is reducible over the algebraic closure $\overline{\mathbb{F}}_p$ iff $f(x)$ is a perfect $r$-th power.

We leave the proof as an exercise for the reader. (Hint: If $h(x, y)$ divides $y^r - f(x)$ then so does $h(x, \omega y)$, where $\omega \in \overline{\mathbb{F}}_p$ satisfies $\omega^r = 1$.) This means that $y^r - f(x)$ is absolutely irreducible if $f(x)$ is not a perfect $r$-th power. The following observation obtains some elementary properties of the rational points of $y^r - f(x)$. It follows easily from the fact that the group of units $\mathbb{F}_p^*$ is cyclic.

**Fact 8.5.** The zeroes of $y^r - f(x)$ are characterized as follows:

- **Case I:** $r$ does not divide $p - 1$. In this case, every element in $\mathbb{F}_p$ has a unique $r$-th root inside $\mathbb{F}_p$ so that the number of zeroes of $y^r - f(x)$ is exactly $p$.
- **Case II:** $r$ divides $p - 1$. Then there exists a primitive $r$-th root of unity $\omega \in \mathbb{F}_p$. For any $a \in \mathbb{F}_p$, we have that either $f(a) = 0$ or

  $$f(a)^{\frac{p-1}{r}} \in \left\{1, \omega, \omega^2, \ldots, \omega^{r-1}\right\}$$

  Furthermore, there exists a $b$ satisfying $b^r = f(a)$ iff

  $$f(a)^{\frac{p-1}{r}} = 0 \text{ or } 1.$$

  Lastly, if $(a, b)$ is a rational point then so is $(a, \omega^i \cdot b)$ for all $i \in [r]$.

If $r$ does not divide $p - 1$, then we are done by the above lemma. For the remainder of this chapter, we will assume that $r$ is a prime that divides $p - 1$.

**Theorem 8.6. Upper bound on the number of zeroes of certain univariate polynomials.** Let $f$ be a polynomial of degree $m$, and let $r$ be a prime which divides $p - 1$ but does not divide $m$. Let

$\theta \in \mathbb{F}_p$ be any element of $\mathbb{F}_p$, then the number of $\mathbb{F}_p$-zeroes of the univariate polynomial

$$f(x)^{\frac{p-1}{r}} - \theta$$

is at most $\frac{p}{r} + O(mr^{\frac{1}{2}}p^{\frac{1}{2}})$.

---

Again, we stress that the upper bound on the number of roots given is far smaller than the degree. The ideas of how this magic happens is explained at high level in section 8.2, and then elaborated for this important case in section 8.3. Before turning to it, we explain how it yields Weil's theorem for these curves, and how it yields the (very useful) Weil's exponential sum bound.

The upper bound of the above theorem allows us to deduce *both upper bounds as well as lower bounds* on the number of $\mathbb{F}_p$-solutions to the bivariate equation under consideration.

---

**Corollary 8.7. Converting upper bounds into lower bounds.** The number of $\mathbb{F}_p$-zeroes of the equation $y^r - f(x) = 0$ is at least

$$p - O\left(mr^{\frac{5}{2}}p^{\frac{1}{2}}\right)$$

---

*Proof.* Let $\omega \in \mathbb{F}_p$ be a primitive $r$-th root of unity. By the theorem above, there are at most

$$(r-1) \cdot \left(\frac{p}{r} + O\left(mr^{\frac{1}{2}}p^{\frac{1}{2}}\right)\right)$$

elements $a$ in $\mathbb{F}_p$ such that

$$f(a)^{\frac{p-1}{r}} \in \{\omega, \omega^2, \ldots, \omega^{r-1}\}$$

For any $a \in \mathbb{F}_p$

$$f(a)^{\frac{p-1}{r}} \in \{1, \omega, \omega^2, \ldots, \omega^{r-1}\}$$

so that there are at least

$$p - \left((r-1) \cdot \left(\frac{p}{r} + O\left(mr^{\frac{1}{2}}p^{\frac{1}{2}}\right)\right)\right) = \frac{p}{r} - O\left(mr^{\frac{3}{2}}p^{\frac{1}{2}}\right)$$

elements $a \in \mathbb{F}_p$ for which $f(a)^{\frac{p-1}{r}} = 1$. Then by Fact 8.5, each such $a$ yields $r$ different $b$'s satisfying $b^r - f(a) = 0$. Therefore, the number of rational points is at least

$$r \cdot \left(\frac{p}{r} - O\left(mr^{\frac{3}{2}}p^{\frac{1}{2}}\right)\right) = p - O\left(mr^{\frac{5}{2}}p^{\frac{1}{2}}\right).$$

□

As this point we note that Theorem 8.6 and its corollary above have an alternative statement in terms of *exponential sums.*

**Exercise 8.8.** Let $\chi : \mathbb{F}_p^* \mapsto \mathbb{C}$ be a multiplicative character of order $r$, i.e. $\chi$ is a nontrivial homomorphism from the group of units $\mathbb{F}_p^*$ to the $r$-th roots of unity in $\mathbb{C}$ with

$$\chi(0) \stackrel{\text{def}}{=} 0.$$

Let $f(x) \in \mathbb{F}_p[x]$ be a polynomial of degree $m$ coprime to $r$, then 8.6 is equivalent to the estimate:

$$\left|\sum_{a \in \mathbb{F}_p *} \chi(f(a))\right| = O\left(mr^{\frac{5}{2}}p^{\frac{1}{2}}\right).$$

## 8.2 A high-level description of the Stepanov method

The Stepanov method is applied to the task of showing that for a given polynomial $h(x)$, the number of $\mathbb{F}_p$-zeroes of the univariate polynomial $h(x) - \theta$ is small for every $\theta \in \mathbb{F}_p$. Theorem 8.6 states this for $h(x) = f(x)^{\frac{p-1}{r}}$. We shall explain the intuition for this special case, but we try to have as much as the discussion apply to a general $h$, so as to build intuition on the structure required to carry the argument through.

Fix $\theta$, and let $P$ denote the set of $\mathbb{F}_p$-zeroes of the univariate polynomial $h(x) - \theta$. The crux of the proof involves the construction (partly via interpolation) of a nonzero polynomial $F(x)$ of relatively low degree (say $D$) which vanishes on every point of $P$ with high multiplicity (say $M$). It will then follow that $|P| \leq \frac{D}{M}$. We make this more precise below, summarizing the important properties of $F(x)$ while postponing the actual construction for a bit.

**Lemma 8.9.** There exists a polynomial $F(x) \in \mathbb{F}_p[x]$ with the following properties:

(1) $F(x)$ is nonzero as a polynomial.

(2) $F(x)$ vanishes with multiplicity at least
$$M = \Theta\left(\frac{p^{1/2}}{r^{1/2}}\right)$$
at all points in $P$.

(3) $D \overset{\text{def}}{=} \deg(F(x)) \leq \frac{p}{r} \cdot M + \Theta(p \cdot m)$.

The upper bound of Theorem 8.6 then follows easily from this construction.

### Proof of Theorem 8.6

The number of zeroes of a polynomial counted with multiplicity cannot exceed its degree so that we have
$$|P| \cdot M \leq \deg(F(x)) \leq \frac{p}{r}M + \Theta(pm)$$

Thus
$$|P| \leq \frac{p}{r} + \frac{\Theta(pm)}{M} \leq \frac{p}{r} + \Theta\left(mr^{\frac{1}{2}}p^{\frac{1}{2}}\right)$$

### Construction of the Polynomial $F(x)$.

**Overview.** We now look at the construction of the polynomial $F(x)$ of Lemma 8.9 as given by Stepanov. This is the core of Stepanov's method. Let us give some intuition before describing the actual construction. In all interesting cases, such as this one, the degree of the polynomial $h(x) - \theta$, is much larger than the upper bound for the number of its $\mathbb{F}_p$-roots that we seek to prove. Let us first get an "algebraic characterization" of the elements of $P$.

**Fact 8.10.** The elements of $P$ are precisely the roots of

$$\gcd\bigl(h(x)-\theta, x^p - x\bigr).$$

In particular, $|P| = \deg(\gcd(h(x)-\theta, x^p-x))$.

This means that any polynomial $F(x)$ which vanishes at every point of $P$ must be a multiple of $\gcd(h(x)-\theta, x^p-x)$. This in turn means $F$ must be of the form

$$F(x) = u(x)\bigl(h(x)-\theta\bigr) + v(x)\bigl(x^p - x\bigr), \tag{8.5}$$

because the gcd is of the above form. Conversely, any $F$ of the form (8.5) above vanishes on all points of $P$. If $M = 1$, we can choose say $u(x) = v(x) = 1$ and obtain $\deg(F) = \max(\deg(h), p)$. We can do even somewhat better and obtain $u(x)$ and $v(x)$ such that $F(x)$ has degree $\min(\deg(h), p)$ but it is not clear how to choose $u(x)$ and $v(x)$ so that the expression (8.5) has a much smaller degree.

How can we gain by using multiplicity? The answer is "amortized analysis" combined with the fact that *both* polynomials $h(x)-\theta$ and $x^p - x$ satisfy "nice differential equations". We need to satisfy

$$F^{(\ell)}(\alpha) = 0, \quad \text{for every } \alpha \in P \text{ and every } 0 \le \ell \le M-1.$$

If it happens that imposing any additional derivative to vanish is cheap (requires much smaller than $\deg(h)$ linear constraints), then we are in business. At a very high level the following will happen. Recall that $D$ is the degree of $F$. The first step of making $F$ vanish on $P$ will require only $D' \ll D$ linear constraints. Moreover, in every differentiation step of $F$ this parameter $D'$ will increase only by $m$, the maximum degree of the polynomials $u(x), v(x)$ above. This will make the total "cost" (in terms of linear constraints) of making $M$ derivatives of $F$ vanish on $P$ only $MD' + mM^2$, far smaller than the trivial $MD$.

Now what is the source of this magic? It comes from the following two differential equations, satisfied by $h(x)-\theta$ and $x^p - x$:

$$\bigl(h(x)-\theta\bigr)' = \frac{p-1}{r}\cdot\frac{h(x)\cdot f'(x)}{f(x)} \quad \text{and} \quad (x^p - x)' = -1 \tag{8.6}$$

We use them to differentiate (8.5). We get

$$\begin{aligned} F' &= u'(h-\theta) + u\cdot h' + v'(x^p - x) + v(x^p-x)' \\ &= \big(u'(h-\theta) + v'(x^p-x)\big) + \left(\frac{(p-1)\cdot u\cdot h\cdot f'}{r\cdot f} - v\right) \end{aligned}$$

Now the first summand vanishes on $P$ so that we must only make sure that the second one does too. We want

$$(p-1)\cdot u(\alpha)\cdot h(\alpha)\cdot f'(\alpha) - r\cdot v(\alpha)\cdot f(\alpha) = 0, \quad \forall\, \alpha\in P.$$

Since $h(\alpha)=\theta$ for each $\alpha\in P$ we want that

$$-\theta\cdot u(\alpha)\cdot f'(\alpha) - r\cdot v(\alpha)\cdot f(\alpha) = 0, \quad \forall\, \alpha\in P$$

The equation above suggests a natural choice of $u$ and $v$. Lets choose $u(x) = r\cdot f(x)$ and $v(x) = -\theta\cdot f'(x)$. This ensures that $-\theta\cdot u(z)f'(z) - r\cdot v(z)f(z)$ is identically zero(!) and therefore, the last equation holds for all $\alpha\in P$. Substituting these choices of $u$ and $v$ back into equation (8.5), we get

$$F = r\cdot f\cdot (h-\theta) - \theta\cdot f'\cdot (x^p - x)$$

so that $\deg(F) \le m + \max(\deg(h), p)$. We have thus ensured the vanishing of the first derivative by paying an additive cost of just $m$ to the degree of $F$.

At this point, a delicate but important issue crops up — we must ensure that the polynomial $F$ obtained in this way, is not identically zero. Often the most difficult part of this kind of arguments is to make sure that $F$ is nonzero. This is also the place where the absolute irreducibility of the polynomial $y^r - f(x)$ is crucially used. In our case, it suffices to ensure that the degrees of the two summands of $F$ are distinct in order to ensure that $F$ is nonzero. We have

$$\begin{aligned} \deg(u\cdot(h-\theta)) - \deg(v\cdot(x^p-x)) &= m + m\cdot\frac{p-1}{r} - ((m-1)+p) \\ &= \frac{p-1}{r}(m-r), \end{aligned}$$

which is nonzero. This shows that

$$|P| \le \frac{1}{2}\left(m + \max\left(p, m\cdot\frac{p-1}{r}\right)\right),$$

a tiny improvement on $\max(p, m \cdot \frac{p-1}{r})$. Of course, in order to derive the maximum benefit, we have to keep applying this reasoning for all derivatives, not just the first. Let us now see these ideas in action with full detail.

## 8.3 Formal proof of Weil's theorem

The polynomial $F(x)$ that we construct will be chosen from an appropriate linear space $L$ of polynomials in $\mathbb{F}_p[x]$. The vanishing of its first $M$ derivatives will be guaranteed by homogeneous linear constraints on $L$. Let the linear space $L(A, B, C)$ be defined as all polynomials in $\mathbb{F}_p[x, y, z]$ with degree at most $A$ in $x$, $B$ in $y$ and $C$ in $z$. We will pick a nonzero polynomial $G$ in $L$ and set

$$F(x) = f(x)^M G(x, h(x), x^p).$$

Using this equation, the chain rule and the differential equations (8.6), we see that

$$F'(x) = f(x)^{M-1} G_1(x, h(x), x^p),$$

where $G_1$ is in $L(A + m, B, C)$. Hence we can repeat the process of taking derivatives, getting that

$$F^{(k)} = f(x)^{M-k} G_k(x, h(x), x^p),$$

with $G_k$ in $L(A + mk, B, C)$, for all $k \leq M$. Moreover, the coefficients of $G_k$ are linear combinations of the coefficients of $G$!

Furthermore, if we have any $G^*$ in $L(A^*, B, C)$ and set

$$F^* = f^* \cdot G^*(x, h(x), x^p)$$

(for any polynomial $f^*$), we can ensure that $F^*(\alpha) = 0$ for *all* $\alpha \in P$ simultaneously by imposing only $A^* + C$ homogeneous linear constraints (for this we use the fact that for all $\alpha \in P$, $h(\alpha) = \theta$ and $\alpha^p = \alpha$). Doing so for all $G_k$, we need to impose $(A + C)M + mM^2$ constraints in total. To make sure the initial $G$ is nonzero we need that the initial space $L(A, B, C)$ has large enough dimension, namely

$$ABC > (A + C)M + mM^2 \tag{8.7}$$

The degree of $F$ is $Mm + A + \deg(h) \cdot B + pC$ which gives an upper bound on the size of the set $P$

$$|P| < \frac{1}{M} \cdot \big(mM + A + \deg(h) \cdot B + pC\big) \tag{8.8}$$

Finally, the fact that $G$ is nonzero does not imply that $F$ is nonzero! Indeed, often this is the hardest thing to arrange and usually leads to further conditions on the parameters $A$, $B$ and $C$. Once all are set, we optimize the value of $M$ to yield the best possible upper bound on $|P|$. Let us now instantiate this approach and specify the best choice of the various parameters.

**The Actual Construction and Choice of Parameters.** We choose the precise values of the parameter $M$ of Lemma 8.9, and the parameters $A$, $B$ and $C$ as follows:

$$\begin{aligned} M &= \sqrt{\frac{2p}{r}} - 3 && (8.9) \\ A &= \frac{p}{r} - m \\ B &= r - 1 \\ C &= \frac{M}{r} + \frac{m+1}{r} && (8.10) \end{aligned}$$

Thus the polynomial $G$ is of the form

$$G(x,y,z) = \left( \sum_{0 \le i \le A} \sum_{0 \le j \le B} \sum_{0 \le k \le C} a_{ijk} \cdot x^i \cdot y^j \cdot z^k \right),$$

so that the polynomial $F(x)$ takes the form

$$F(x) = f(x)^M \cdot \left( \sum_{0 \le i \le A} \sum_{0 \le j \le B} \sum_{0 \le k \le C} a_{ijk} \cdot x^i \cdot h(x)^j \cdot x^{pk} \right), \tag{8.11}$$

where the $a_{ijk}$'s are unknowns that we will ultimately determine by solving an appropriate system of homogeneous linear equations.

We now prove the three properties of $F$ from lemma 8.9.

**First Property.** Any nontrivial choice of $F(x)$ of the form given by (8.11) above is nonzero:

**Proposition 8.11.** If $G(x,y,z)$ is nonzero then $F(x)$ is nonzero.

*Proof.* A typical summand of $F(x)$ is of the form

$$H_{ijk}(x) = a_{ijk} \cdot x^i \cdot h(x)^j \cdot x^{pk}.$$

It suffices to show that the degrees of the nonzero summands are all distinct. If $r = 1$ then this fact would be trivial — for larger $r$ it will follow from the fact that $r$ and $m$ are relatively prime, as follows.

$$\deg(H_{ijk}(x)) = i + j \cdot \frac{p-1}{r} \cdot m + pk = \frac{p}{r}(rk + jm) + i - \frac{j}{r}m$$

whence by the choice of $A$ and $B$,

$$\frac{p}{r}(rk + jm) - m \;<\; \deg(H_{ijk}) \;\leq\; \frac{p}{r}(rk + jm) + \frac{p}{r} - m$$

Hence we need only verify that for pairs $(j,k) \neq (j',k')$, we have

$$(rk + jm) \neq (rk' + j'm).$$

So suppose

$$rk + jm = rk' + j'm$$

Then

$$mj \equiv mj' \pmod{r}$$

So since $r$ is a prime not dividing $m$ we have

$$j \equiv j' \pmod{r}$$

But $0 \leq j, j' \leq r - 1$, so $j = j'$ and $k = k'$. □

This shows that $F(x)$ satisfies the first property. We wish to make $F(x)$ vanish with high multiplicity at every point in $P$. For this we need to examine the derivatives of the summands. The following proposition carries out the (straightforward) computation of derivatives of $F(x)$ and shows that they are pretty much of the same form as $F(x)$ itself.

**Proposition 8.12. Computation of derivatives.** For $\ell \leq M$, the $\ell$-th derivative of $F(x)$ is of the form:

$$F^{(\ell)}(x) = f(x)^{M-\ell} G_\ell(x, h(x), x^p),$$

where $G_\ell(x, y, z) \in \mathbb{F}_p[x, y, z]$ is a polynomial in the linear space $L(A + \ell(m-1), B, C)$. The coefficients of $G_\ell(x, y, z)$ are linear combinations of the (unknown) coefficients of $G(x, y, z)$.

The proof goes by induction, with the inductive step involving some straightforward differentiation. We leave the details as an exercise for the reader.

**Second Property.** Let $\alpha \in P$ be an arbitrary element of $P$ and let us look at the $\ell$-th partial of $F(x)$ evaluated at $\alpha$. We have

$$\begin{aligned} F^{(\ell)}(\alpha) &= f(\alpha)^{M-\ell} \cdot G_\ell(\alpha, h(\alpha), \alpha^p) \\ &= f(\alpha)^{M-\ell} \cdot G_\ell(\alpha, \theta, \alpha) \qquad (\text{as } h(\alpha) = \theta \text{ and } \alpha^p = \alpha) \end{aligned}$$

Notice that the factor $G_\ell(\alpha, \theta, \alpha)$ is a relatively low-degree polynomial in $\alpha$. We will choose the $a_{ijk}$'s in such a way as to ensure that this sum is zero. The degree of $G_\ell(\alpha, \theta, \alpha)$, viewed as a polynomial in $\alpha$, is at most $C + A + \ell(m-1)$. This means that for each $\ell \in \{0, \ldots, M\}$, we are imposing $C + A + \ell(m-1)$ homogeneous linear constraints on the $a_{ijk}$'s. Thus, we have

$$\begin{aligned} \text{Number of Constraints} &= (C + A) \cdot M + (m-1) \cdot \frac{M \cdot (M+1)}{2} \\ \text{Available Coefficients} &= A \cdot (B+1) \cdot (C+1) \end{aligned}$$

The reader can now verify that our choice of the parameters $A, B, C$ and $M$ ensure that the number of available coefficients is more than the number of constraints so that a nontrivial solution to this system of homogeneous linear equations always exists. Thereby we get that there exist $a_{ijk}$'s, not all zero, such that $F(x)$ vanishes with multiplicity $M$ everywhere on $P$. This shows that $F(x)$ satisfies property (2).

**Third Property.** Lastly, we verify that our choice of the parameters also ensures that $F(x)$ satisfies the third property.

$$\begin{aligned}\deg(F(x)) &\le \deg\left(f(x)^M\right) + \left(A + B \cdot \frac{m \cdot (p-1)}{r} + C \cdot p\right) \\ &= m \cdot M + A + m \cdot \frac{p-1}{r} \cdot (r-1) + p \cdot C \\ &= \frac{p}{r} \cdot M + \Theta(p \cdot m)\end{aligned}$$

This completes the construction of $F$ and hence, the proof of Theorem 8.3 for the special case of polynomials of the form $y^r - f(x)$, with $r$ prime and $\gcd(r, \deg(f)) = 1$.

We then leave the proof of the following more general lemma as a practice problem for the interested readers.

**Exercise 8.13.** Let $\mathbb{F}$ be a field (of say characteristic zero) and $f_1(x)$, $f_2(x) \in \mathbb{F}[x]$ be two polynomials of degree $d_1$ and $d_2$, respectively. If $\gcd(d_1, d_2) = 1$, then for any $\theta_1, \theta_2 \in \mathbb{F}$, the number of common roots of $f_1(x)^n - \theta_1$ and $f_2(x)^n - \theta_2$ is at most $n + O(d_1 d_2 \sqrt{n})$.

## 8.4 The Heath-Brown and Mit'kin estimates

In independent work around 1993, Heath-Brown [HB96] and Mit'kin [Mit92] proved the following:

**Theorem 8.14.** Let $h(x) \in \mathbb{F}_p[x]$ be the following polynomial:

$$h(x) = x + \frac{x^2}{2} + \frac{x^3}{3} + \ldots + \frac{x^{p-1}}{p-1}.$$

Then for any $\theta \in \mathbb{F}_p$, the number of $\mathbb{F}_p$-roots of $h(x) - \theta$ is at most $O(p^{\frac{2}{3}})$.

Note that the given polynomial has highest possible degree $p-1$, but despite that it attains no $\mathbb{F}_p$-value more than $p^{2/3}$ times. The two papers had different motivations (which are detailed therein). For Mit'kin it was showing that Stepanov's method can be applied in completely different situations than the original, especially to "transcendental-looking" polynomials, like the logarithm and exponential. For Heath-

Brown this polynomial arose naturally as part of his proof for giving a nontrivial estimate on the Heilbronn exponential sum.

We use Stepanov's method described in the previous section. What guides the proof is that $h(x)$ looks like a discrete version of $\log(1-x)$. This gives us two insights. First, it supplies the following appropriate differential equation:

$$(x^2 - x)h' = (x^p - x). \tag{8.12}$$

Notice that as before, the $\mathbb{F}_p$-roots of $h(x)$ are precisely the common roots of $h(x)$ and $x^p - x$ and that is where the structure of the differential equation above helps us. Secondly, as $h$ "looks like" a transcendental equation, it should have no low degree algebraic relations. Specifically,

**Lemma 8.15.** Let $G(x,y) \in \mathbb{F}_p[x,y]$ be any polynomial of degree $A$ in $x$ and degree $B$ in $y$. If $AB < p$ then

$$G(x, h(x)) = 0 \pmod{x^p - x}$$

if and only if $G$ is identically zero.

This lemma (which also uses the differential equation) is the hardest part of the argument, and we skip the proof. It gives us the following: For any nonzero polynomial $G \in L(A,B,C)$, if $AB < p$ then

$$F(x) \stackrel{\text{def}}{=} (x^2 - x)^M \cdot G(x, h(x), x^p)$$

is nonzero as well.

**Choice of parameters.** We choose

$$\begin{aligned} A &= p^{\frac{2}{3}} \\ B &= p^{\frac{1}{3}} \\ C &= p^{\frac{1}{3}} \\ M &= \frac{p^{\frac{2}{3}}}{3} \end{aligned}$$

We see that the number of available coefficients, namely $ABC$ is larger than the number of imposed constraints which is at most $(A+C)\cdot M+$

$2M^2$ (as here the "cost" $m$ per differentiation is only $m = 2$, the degree of $x^2 - x$). On the other hand, the degree $D$ of $F$ is $2M + A + Bp + Cp$ so that the set of common zeroes has cardinality at most

$$\frac{\deg(F)}{M} = \frac{2M + A + Bp + Cp}{M} = O(p^{\frac{2}{3}}).$$

# Part II: Lower Bounds

## Overview

In this part, we will see how partial derivatives are useful in proving lower bounds on a variety of arithmetic models of computation.

In Chapter 9, we start with the most general model of arithmetic circuits. After some basic results, we present the only nontrivial lower bound for it so far: an $n \log d$ lower bound on circuits computing the sum of $d$-powers of $n$ variables. A central piece of the proof is a result showing that if a circuit of size $s$ computes a polynomial $f$, then with $O(s)$ size one can additionally compute all the first-order partial derivatives of $f$. So the use of partial derivatives here is indirect, reducing the proof of a lower bound on computing a single polynomial to that of a lower bound on computing many polynomials simultaneously.

In Chapter 10, we move to restricted models, and demonstrate the use of partial derivatives as a "progress measure" in a very special situation. We consider a restricted kind of depth-3 circuits which compute the sum of powers of linear functions. For this very restricted model, considering all partial derivatives provides exponential lower bounds, even when a monomial is to be computed.

In Chapter 11, we consider other restricted forms of depth-3 arithmetic circuits, for which the use of partial derivatives for lower bounds is more sophisticated (and the bounds are typically weaker).

In Chapter 12, we consider arithmetic formulae. We first recall that in the arithmetic setting formulae are not much weaker than circuits, and then move to derive some more lower bounds. In particular, for *multilinear* formulae, Raz proved an $n^{\Omega(\log n)}$ lower bound for both Determinant and Permanent, by combining partial derivatives with random restrictions. We demonstrate this approach by presenting a lower bound for iterated matrix multiplication.

In Chapter 13, we recall the Permanent vs. Determinant problem mentioned in Chapter 2, motivated by their completeness properties. We then show the power of the Hessian, the matrix of all second-order partial derivatives of a polynomial, in giving the best lower bound for projecting the Permanent to Determinant. In Part Three we will also see algorithmic uses of the Hessian.

# 9

# General arithmetic circuits

What is the smallest arithmetic circuit computing a given polynomial? Our current knowledge falls far short of giving satisfactory answers to questions of this type in general. In this chapter we first look at some well known (families of) polynomials and give some of the smallest known arithmetic circuits for computing these polynomials. We then give an exposition of the state of the art in terms of lower bounds for general arithmetic circuits and the role played by partial derivatives in this result.

We first review known upper bounds for some interesting polynomials. Then we prove the classical $\Omega(n \log d)$ lower bound of Strassen [Str73a] and Baur-Strassen [BS83] for $\mathsf{S}(\sum_{i=1}^{n} x_i^d)$.

We start with *matrix multiplication*. Let

$$\mathbf{X} = (x_{i,j})_{i,j\in[n]} \quad \text{and} \quad \mathbf{Y} = (y_{i,j})_{i,j\in[n]}$$

be two $n \times n$ matrices with $2n^2$ variables. We need to design an arithmetic circuit with $n^2$ output gates to compute $\mathbf{XY}$. Because

$$\left(\mathbf{XY}\right)_{i,j} = \sum_{k\in[n]} x_{i,k} \cdot y_{k,j},$$

we have $\mathsf{S}(\mathbf{XY}) = O(n^3)$. However, Strassen's matrix multiplication algorithm [Str69] implies that $\mathsf{S}(\mathbf{XY}) = O(n^{2.81})$. The best upper bound right now is $O(n^{2.376\ldots})$ by Coppersmith and Winograd [CW90].

**Open Problem 9.1.** What is $\mathsf{S}(\mathbf{XY})$? Is it $O(n^2)$?

We are also interested in the complexity of $\mathbf{Ax}$, where $\mathbf{A}$ is a fixed $n \times n$ matrix and $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is a vector of $n$ variables. Note that an arithmetic circuit for $\mathbf{Ax}$ should have $n$ output gates to compute $\sum_{j \in [n]} A_{i,j} \cdot x_j$, for every $i \in [n]$. It can be shown that for almost all matrices $\mathbf{A}$, $\mathsf{S}(\mathbf{Ax}) \geq c \cdot n^2$, for some constant $c > 0$. However, the following problem remains open.

**Open Problem 9.2.** Find an explicit matrix $\mathbf{A}$ such that $\mathsf{S}(\mathbf{Ax}) \gg n$. (Actually, even showing $\mathsf{S}(\mathbf{Ax}) > 3n$ would be interesting.)

Next we list some known size upper bounds concerning the *permanent* $\textsc{Per}_n$, *determinant* $\textsc{Det}_n$, and the (elementary) *symmetric polynomial* $\mathrm{SYM}_n^d$.

**Definition 9.3.** Let $\mathbf{X}_n = (x_{i,j})_{i,j \in [n]}$ be an $n \times n$ matrix with $n^2$ variables. We define the permanent and determinant of $\mathbf{X}_n$ as

$$\textsc{Per}_n(\mathbf{X}_n) = \sum_{\sigma} \prod_{i \in [n]} x_{i,\sigma(i)}$$

and

$$\textsc{Det}_n(\mathbf{X}_n) = \sum_{\sigma} \mathrm{sgn}(\sigma) \prod_{i \in [n]} x_{i,\sigma(i)}.$$

The summation is over the set of permutations $\sigma$ from $[n]$ to itself.

For $d \leq n$, we define the $d$-th elementary symmetric polynomial, in $n$ variables $x_1, \ldots, x_n$, $\mathrm{SYM}_n^d \in \mathbb{F}[x_1, \ldots, x_n]$ as

$$\mathrm{SYM}_n^d = \sum_{S \subseteq [n],\, |S| = d} \prod_{i \in S} x_i.$$

One way to compute $\text{DET}_n$ is to use the Gaussian elimination. However, one cannot implement it directly using an arithmetic circuit, as division is not allowed. Strassen [Str73b] gives a division-free arithmetic circuit of size $O(n^3)$ for computing $\text{DET}_n$ and thus, we have

**Lemma 9.4.** $\mathsf{S}(\text{DET}_n) = O(n^3)$.

The computation of the permanent, in contrast, is believed to be much harder. We will come back to the comparison of permanent and determinant in Chapter 13.

Note that the number of monomials in $\text{PER}_n$ is $n!$. This gives us a trivial upper bound $n \cdot n!$ for $\mathsf{S}(\text{PER}_n)$ by computing these monomials one by one. Indeed there is a much more efficient way to expand $\text{PER}_n$ called Ryser's formula, which we present below. The formula is based on the inclusion-exclusion principle, which gives an upper bound much better than $n \cdot n!$.

Let $S_{k,n}$ denote the set of $k$-tuples $\pi = (i_1, \ldots, i_k)$, with $1 \leq i_1 < i_2 < \ldots < i_k \leq n$. Given $\pi \in S_{k,n}$ and an $n \times n$ matrix $\mathbf{X}$, we let $\mathbf{X}_\pi$ denote the new matrix obtained by setting every entry of $\mathbf{X}$ in columns $i_1, i_2, \ldots, i_k$ to be 0. We also use $\mathsf{w}(\mathbf{X})$ to denote $\prod_{i=1}^n (\sum_{j=1}^n x_{i,j})$.

Then, the permanent of $\mathbf{X}$ is equal to

$$\mathsf{w}(\mathbf{X}) + \sum_{k=1}^{n} \sum_{\pi \in S_{k,n}} (-1)^k \cdot \mathsf{w}(\mathbf{X}_\pi).$$

The proof follows directly from the inclusion-exclusion principle.

**Corollary 9.5.** $\mathsf{S}(\text{PER}_n) = O(n^2 \cdot 2^n)$.

In the rest of this chapter, we will focus on the following two problems: the computation of $(x_1^d, x_2^d, \ldots, x_n^d)$ (in which we need to output $n$ polynomials) and $\sum_{i=1}^n x_i^d$, where we assume the field $\mathbb{F}$ to be $\mathbb{C}$. It is easy to see that both

$$\mathsf{S}\big(x_1^d, x_2^d, \ldots, x_n^d\big) \quad \text{and} \quad \mathsf{S}\big(\textstyle\sum_{i=1}^d x_i^d\big)$$

are $O(n \log d)$. Intuitively, one might guess that $n \log d$ is also a lower bound for $\mathsf{S}(x_1^d, \ldots, x_n^d)$, since $\mathsf{S}(x^d) = \Theta(\log d)$, and the $n$ monomials

$x_i^d$ are in distinct variables. In other words, it might seem that evaluating a polynomial $f$ on $n$ independent inputs cannot be combined, and the computational effort must increase $n$-fold when compared to evaluating $f$ on a single input. This hypothesis, which is often called a *direct-sum conjecture*, is natural to make for this and other computational models, but is often wrong, as shown by the following example.

**Example 9.6.** As mentioned earlier, for almost all $n \times n$ matrices $\mathbf{A}$, $\mathsf{S}(\mathbf{Ax}) = \Omega(n^2)$. Let $\mathbf{A}$ be such a matrix. Now we need to compute $n$ polynomials $\mathbf{Ax}_1, \ldots, \mathbf{Ax}_n$, where $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,n})$. Even though the $n$ variables $\mathbf{x}_i$ in $\mathbf{Ax}_i$ do not appear in any other linear forms and intuitively, the computation of $\mathbf{Ax}_i$ should be independent from the computation of other linear forms, it follows from $\mathsf{S}(\mathbf{XY})$ that

$$\mathsf{S}\big(\mathbf{Ax}_1, \ldots, \mathbf{Ax}_n\big) = O(n^{2.376\ldots}) \ll n \cdot \mathsf{S}(\mathbf{Ax}).$$

As we will see, even for these two seemingly simple problems, proving tight lower bounds is very challenging and counterintuitive. We first present Strassen's $\Omega(n \log d)$ lower bound for $\mathsf{S}(x_1^d, x_2^d, \ldots, x_n^d)$ [Str73a]. The same bound is later extended to $\mathsf{S}(\sum_{i=1}^n x_i^d)$ by Strassen and Baur [BS83] for which we present a simplified proof of Morgenstern [Mor85].

We start with the $\Omega(n \log d)$ lower bound for $\mathsf{S}(x_1^d, \ldots, x_n^d)$. It uses several concepts from Algebraic Geometry, and in particular, Bezout's Theorem [Har77, I.7, Thm.7.7]. We need the following definitions.

**Definition 9.7.** Let $f_1, \ldots, f_k$ be a set of polynomials in $\mathbb{C}[\mathbf{X}]$. We define their *variety* as

$$\mathsf{V}\big(f_1, \ldots, f_k\big) = \big\{\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{C}^n \;\big|\; f_1(\mathbf{a}) = \ldots = f_k(\mathbf{a}) = 0\big\}.$$

The *degree* of their variety, denoted by $\deg(\mathsf{V}(f_1, \ldots, f_k))$, is the maximum *finite* number achievable by $|\,\mathsf{L} \cap \mathsf{V}(f_1, \ldots, f_k)\,|$, where $\mathsf{L}$ is any affine subspace of $\mathbb{C}^n$ defined by a collection of affine forms $\{\ell_1, \ldots, \ell_m\}$:

$$\mathsf{L} = \big\{\mathbf{a} \in \mathbb{C}^n \;\big|\; \ell_1(\mathbf{a}) = \ldots = \ell_m(\mathbf{a}) = 0\big\} \subseteq \mathbb{C}^n.$$

We use the following example to explain these two concepts.

**Example 9.8.** Let $f(x,y) = y - x^d \in \mathbb{C}[x,y]$, then we show that

$$\deg\big(\mathsf{V}(f)\big) = d.$$

First, $\deg(\mathsf{V}(f)) \geq d$ because after restricting $y = 1$, we get $1 - x^d$ which has $d$ roots in $\mathbb{C}$. Second, we have $\deg(\mathsf{V}(f)) \leq d$ because

(1) If we do not add any affine linear constraint, then $\mathsf{L} = \mathbb{C}^n$ and $|\mathsf{L} \cap \mathsf{V}(f)|$ is clearly infinite;

(2) If we add two affine linear constraints, then $|\mathsf{L} \cap \mathsf{V}(f)| \leq 1$;

(3) If we add one affine linear constraint, then the degree of the resulting univariate polynomial is at most $d$ and thus, the number of roots is at most $d$.

As a result, the degree of $\mathsf{V}(f)$ is exactly $d$.

Now we prove $\mathsf{S}(x_1^d, \ldots, x_n^d) = \Omega(n \log d)$. Let

$$\mathcal{C} = (x_1, \ldots, x_n, g_{n+1}, \ldots, g_s, g_{s+1}, \ldots, g_{s+n})$$

be an arithmetic circuit that computes $(x_1^d, \ldots, x_n^d)$. In particular $g_{s+i}$ outputs $x_i^d$ for every $i \in [n]$. We also assume each gate $g_i$ has bounded fan-in 2, since any arithmetic circuit of size $t$ can be easily converted to a bounded fan-in circuit of size $O(t)$. Because we assumed that the fan-in of $\mathcal{C}$ is bounded, we have $\mathsf{S}(\mathcal{C}) = \Theta(s)$.

Next we use $\mathcal{C}$ to define a collection of polynomials. For every $i \in [n+1 : s+n]$, we define a polynomial $f_i \in \mathbb{C}[x_1, \ldots, x_n, \ldots, x_{s+n}]$:

(1) $f_i = x_i - \alpha x_j - \beta x_k$ if $g_i = \alpha g_j + \beta g_k$ in $\mathcal{C}$; or

(2) $f_i = x_i - \alpha x_j x_k$ if $g_i = \alpha g_j g_k$ in $\mathcal{C}$.

We also let $h_i = x_{s+i} - x_i^d$ be a polynomial in

$$\mathbb{C}[x_1, \ldots, x_n, x_{s+1}, \ldots, x_{s+n}],$$

for every $i \in [n]$.

Then the lower bound on $s$ follows from the following inequalities:

$$\begin{aligned} d^n &\leq \deg\big(\mathsf{V}(h_1,\ldots,h_n)\big) && (9.1)\\ &\leq \deg\big(\mathsf{V}(f_{n+1},f_{n+2},\ldots,f_{n+s})\big) && (9.2)\\ &\leq \prod_{i\in[s]} \deg\big(\mathsf{V}(f_{n+i})\big) && (9.3)\\ &\leq 2^s. && (9.4)\end{aligned}$$

(9.1) follows from the fact that if we restrict $x_{s+i}=1$ for all $i\in[n]$ then there are $d^n$ roots to $\{x_i^d=1 : i\in[n]\}$.

(9.2) holds because

$$(x_1,\ldots,x_n,x_{n+1},\ldots,x_{s+n}) \mapsto (x_1,\ldots,x_n,x_{s+1},\ldots,x_{s+n})$$

is clearly a one-to-one correspondence between

$$\mathsf{V}(f_{n+1},\ldots,f_{n+s}) \quad\text{and}\quad \mathsf{V}(h_1,\ldots,h_n),$$

because we assumed that $(x_1^d,\ldots,x_n^d)$ is correctly computed by $\mathcal{C}$. As a result, any subspace $\mathsf{L}$ of $\mathbb{C}^{2n}$ yields a subspace $\mathsf{L}'$ of $\mathbb{C}^{s+n}$ (defined by the same set of affine forms) such that

$$\big|\mathsf{L}'\cap\mathsf{V}(f_{n+1},\ldots,f_{n+s})\big| = \big|\mathsf{L}\cap\mathsf{V}(h_1,\ldots,h_n)\big|.$$

(9.3) is an application of Bezout's Theorem [Har77]. The theorem states that, under certain conditions (which are satisfied here),

$$\deg\big(\mathsf{V}(f_1,\ldots,f_k)\big) \leq \prod_{i\in[k]} \deg(\mathsf{V}(f_i)).$$

Finally, (9.4) follows from $\deg(\mathsf{V}(f_i))\leq 2$ for all $n+1\leq i\leq n+s$.

It then follows that $\mathsf{S}(\mathcal{C})=\Theta(s)=\Omega(n\log d)$. Besides, it should be noted that other than (9.1), the proof above actually works for any set of polynomials $(f_1,\ldots,f_k)$. As a result, it gives us the following more general lower bound for $\mathsf{S}(f_1,\ldots,f_k)$ [Str73a]. Indeed this lower bound applies even if we only count multiplication gates!

**Theorem 9.9.** $\mathsf{S}(f_1,\ldots,f_k)\geq\log\deg\big(\mathsf{V}(h_1,\ldots,h_k)\big)$, where

$$h_i := y_i - f_i(x_1,\ldots,x_n), \qquad \text{for every } i\in[k].$$

Next we extend the same lower bound to $\mathsf{S}(\sum_{i=1}^n x_i^d)$. It is a direct corollary of the following theorem of Baur and Strassen [BS83]:

**Theorem 9.10.** For any field $\mathbb{F}$ and $f \in \mathbb{F}[x_1, \ldots, x_n]$, we have

$$\mathsf{S}\big(\partial_1(f), \ldots, \partial_n(f)\big) \leq 5\mathsf{S}(f).$$

**Corollary 9.11.** $\mathsf{S}\big(\sum_{i=1}^n x_i^d\big) = \Omega(n \log d)$.

*Proof.* By Theorem 9.10, we have

$$5\mathsf{S}\big(\textstyle\sum_{i=1}^n x_i^d\big) \geq \mathsf{S}\big(x_1^{d-1}, \ldots, x_n^{d-1}\big) = \Omega\big(n \log(d-1)\big). \qquad \square$$

We present the following simplified proof of Morgenstern [Mor85].

*Proof.* (of Theorem 9.10) Roughly speaking, the idea is to compute from the given circuit $\mathcal{C}$ for $f$, inductively backwards from its output, the partial derivatives of $f$ with respect to *every gate* of $\mathcal{C}$. The chain rule will imply that each of these partial derivatives can be computed by adding a few new gates and edges into $\mathcal{C}$. Moreover, the added gates can be naturally arranged in a "mirror image" of $\mathcal{C}$, as shown in Figure 9.1 for a simple example. Note that in particular, we have computed the partial derivatives of $f$ with respect to all its variables, and these become the outputs of the new circuit. Implementing this idea formally requires some care.

Suppose the following arithmetic circuit

$$\mathcal{C} = (x_1, \ldots, x_n, g_1, \ldots, g_s)$$

computes $f \in \mathbb{F}[x_1, \ldots, x_n]$.

First we view all the $x_i$'s and $g_i$'s in the circuit $\mathcal{C}$ as variables. Then we inductively define a sequence of polynomials $f_s, f_{s-1}, \ldots, f_0$, where $f_i \in \mathbb{F}[x_1, \ldots, x_n, g_1, \ldots, g_i]$ for all $i : 0 \leq i \leq s$. The polynomial $f_s$ is simply $g_s$. Now suppose $f_i \in \mathbb{F}[x_1, \ldots, x_n, g_1, \ldots, g_i]$ has already been defined, then $f_{i-1}$ is the polynomial obtained by replacing the variable

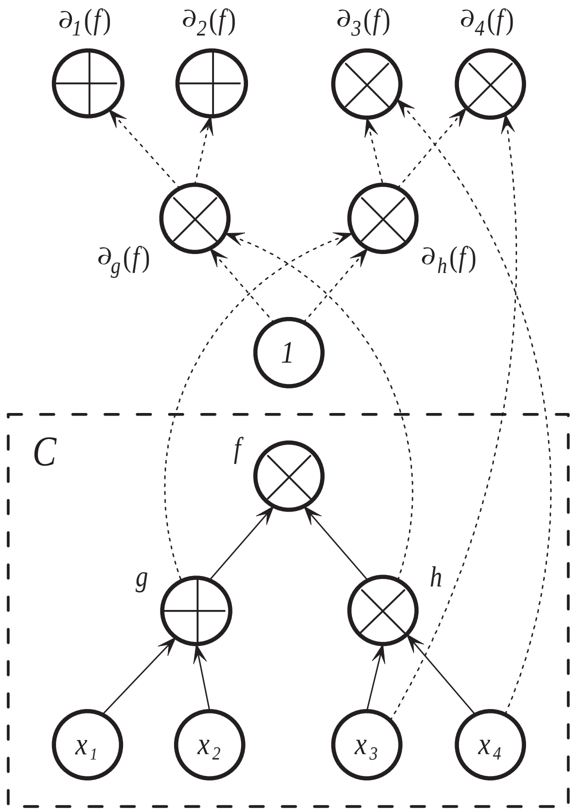

Fig. 9.1 Adding new gates and edges to compute the partial derivatives $\partial_i(f)$ of $f$.

$g_i$ in $f_i$ with either $\alpha g_j + \beta g_k$ or $g_j g_k$, depending on the operation of $\mathcal{C}$ at $g_i$. It is easy to see that $f_0$ is exactly $f \in \mathbb{F}[x_1, \ldots, x_n]$ since $\mathcal{C}$ is assumed to compute $f$.

Next, we start to add new gates into the original circuit $\mathcal{C}$ to compute a sequence of partial derivatives:

$$\partial_{g_i}(f_i)\big|_{g_j=\mathcal{C}_j,\, j\in[i]} \in \mathbb{F}[x_1, \ldots, x_n]$$

from $i = s$ to 1. $\partial_{g_i}(f_i)$ is a polynomial in $\mathbb{F}[x_1, \ldots, x_n, g_1, \ldots, g_i]$ and $\partial_{g_i}(f_i)|_{g_j=\mathcal{C}_j,\, j\in[i]}$ is obtained by replacing every $g_j$ with $\mathcal{C}_j \in \mathbb{F}[x_1, \ldots, x_n]$, the polynomial computed at $g_j$ in $\mathcal{C}$. For convenience, we simply denote $\partial_{g_i}(f_i)|_{g_j=\mathcal{C}_j,\, j\in[i]}$ by $\partial_{g_i}(f_i)$. After obtaining this sequence

$$\partial_{g_s}(f_s), \ldots, \partial_{g_1}(f_1),$$

we will follow the same idea to compute $\partial_{x_i}(f)$.

The basis is trivial: $\partial_{g_s}(f_s) = 1$. For the induction step, we need to show that if

$$\partial_{g_s}(f_s), \ldots, \partial_{g_{k+1}}(f_{k+1})$$

have already been computed in the circuit, then one only need to add a small number of gates to get $\partial_{g_k}(f_k)$. Since $\partial_{g_k}(f_k)$ can be written as

$$\partial_{g_k}(f_k) = \sum_{i=k}^{s-1} \big(\partial_{g_k}(f_i) - \partial_{g_k}(f_{i+1})\big) + \partial_{g_k}(f_s) = \sum_{i=k}^{s-1} \big(\partial_{g_k}(f_i) - \partial_{g_k}(f_{i+1})\big),$$

where we use $\partial_{g_k}(f_i)$, $i \geq k$, to denote

$$\partial_{g_k}(f_i)\Big|_{g_j=\mathcal{C}_j, j\in[i]} \in \mathbb{F}[x_1, \ldots, x_n],$$

we take a close look at $\partial_{g_k}(f_{\ell+1}) - \partial_{g_k}(f_\ell)$, $\ell \geq k$. There are three cases.

First, if gate $g_k$ is not a predecessor of $g_{\ell+1}$ in $\mathcal{C}$, then we have

$$\partial_{g_k}(f_{\ell+1}) - \partial_{g_k}(f_\ell) = 0,$$

since the only difference between $f_{\ell+1}$ and $f_\ell$ is that $g_{\ell+1}$ is replaced by other $g$'s, and this does not affect the derivative with respect to $g_k$.

Second, if gate $g_{\ell+1} = \alpha g_k + \beta g_m$, for some other $m : m \neq k$ and $m \leq \ell$, then using the chain rule we have

$$\partial_{g_k}(f_\ell) = \partial_{g_k}(f_{\ell+1}) + \partial_{g_k}(g_{\ell+1})\partial_{g_{\ell+1}}(f_{\ell+1}) = \partial_{g_k}(f_{\ell+1}) + \alpha \cdot \partial_{g_{\ell+1}}(f_{\ell+1}).$$

Finally, if gate $g_{\ell+1} = \alpha g_k g_m$ then we have the following two cases. If $m \neq k$, then using the chain rule, we have

$$\partial_{g_k}(f_\ell) = \partial_{g_k}(f_{\ell+1}) + \alpha\mathcal{C}_m \cdot \partial_{g_{\ell+1}}(f_{\ell+1}).$$

If $m = k$ then

$$\partial_{g_k}(f_\ell) = \partial_{g_k}(f_{\ell+1}) + 2\alpha\mathcal{C}_k \cdot \partial_{g_{\ell+1}}(f_{\ell+1}).$$

Let $N_k$ denote the set of successors of $g_k$ in $\mathcal{C}$, we have

$$\partial_{g_k}(f_k) = \sum_{i=k}^{s-1} \big(\partial_{g_k}(f_i) - \partial_{g_k}(f_{i+1})\big) = \sum_{\ell\in N_k} \partial_{g_k}(g_\ell) \cdot \partial_{g_\ell}(f_\ell).$$

Note that we assumed all the $\partial_{g_\ell}(f_\ell)$'s, $\ell > k$, have already been computed in the circuit. Also note that all the $\mathcal{C}_m$'s, $m \in [s]$, have already been computed in the circuit. As a result, we can compute $\partial_{g_k}(f_k)$ by adding at most $2|N_k| - 1$ new gates and $4|N_k| - 2$ new edges.

Now we have computed the sequence $\partial_{g_s}(f_s), \ldots, \partial_{g_1}(f_1)$. By using the same argument, one can show that $\partial_{x_i}(f_0) = \partial_{x_i}(f)$ can be computed by adding at most $2|M_i| - 1$ new gates and $4|M_i| - 2$ new edges where $M_i$ denotes the set of successors of $x_i$ in $\mathcal{C}$.

Since $\mathsf{S}(\mathcal{C}) = \sum |M_i| + \sum |N_i|$, the total number of edges inserted is at most $4\mathsf{S}(\mathcal{C})$ and thus, the size of the new circuit is $\leq 5\mathsf{S}(\mathcal{C})$. □

**Exercise 9.12.** Prove that given a matrix $\mathbf{A} \in \mathbb{C}^{n\times n}$, computing the matrix vector product $\mathbf{Ax}$ (with $n$ outputs) is at most a constant factor harder than computing the (single-output) bilinear form $\mathbf{yAx}$.

**Open Problem 9.13.** Let $f(x_1, \ldots, x_n, y_1, \ldots, y_m)$ be a function in $n + m$ variables. Is it true that

$$\mathsf{S}\left(\frac{\partial^2 f}{\partial x_i \partial x_j}\right) \leq O\big(\mathsf{S}(f) + n^2\big)?$$

**Exercise 9.14.** If the statement above is true, then matrix multiplication has a $O(n^2)$ algorithm.

**Open Problem 9.15.** The lower bounds above turn out to be the strongest ones we have right now for general arithmetic circuits. Can we prove stronger lower bounds for polynomials of constant degree?

# 10

# Sums of powers of linear forms

In chapter 2 we saw that every $n$-variate polynomial $f$ of degree $d$ can be written as

$$f = \ell_1^d + \ell_2^d + \ldots + \ell_s^d,$$

where the $\ell_i$'s are affine forms. For a given polynomial $f$, what is the smallest $s$ such that $f$ has an expression of the above form? In this chapter we will see how partial derivatives can be used to derive an exponential lower bound for the smallest $s$ required for computing a single monomial.

The *degree approach* described in Chapter 9 gives tight lower bounds for $\mathsf{S}(x_1^d, \ldots, x_n^d)$ and $\mathsf{S}(\sum_{i=1}^n x_i^d)$. So far, they are still the strongest size lower bounds we have under the general arithmetic circuit model. Proving lower bounds in this general setting has been known as a very challenging problem, and right now we do not have techniques powerful enough to deal with it. One of the directions is to make extra restrictions on the arithmetic circuits considered, and prove (size or depth) lower bounds for these restricted (but still interesting and nontrivial) classes of arithmetic circuits.

In this chapter we begin our exposition of lower bounds for restricted models of computation with a small warmup. We present a lower bound for representing a polynomial as a sum of powers of linear forms. [1] It complements the lemma by Ellison (Lemma 2.4) which shows that every polynomial can be represented in this fashion. It shows that representing a polynomial as a sum of powers of linear forms is an exceedingly restricted and weak model of computation for polynomials. We begin with a definition:

**Definition 10.1.** Let

$$\mathbf{f}(\mathbf{X}) \stackrel{\text{def}}{=} \big(f_1(\mathbf{X}), f_2(\mathbf{X}), \ldots, f_m(\mathbf{X})\big) \in \big(\mathbb{F}[\mathbf{X}]\big)^m$$

be a set of $m$ polynomials over a field $\mathbb{F}$. The $f_i$'s are said to be $\mathbb{F}$-linearly dependent if there exist constants $a_1, a_2, \ldots, a_m \in \mathbb{F}$, not all zero such that

$$a_1 f_1 + a_2 f_2 + \ldots + a_m f_m = 0.$$

The lower bound we present here is based on the following observation: a polynomial which is computed by a small sum-of-power circuit has the property that only a few of its partial derivatives are $\mathbb{F}$-linearly independent.

**Lemma 10.2.** Let $f_n = x_1 x_2 \cdot \ldots \cdot x_n$. If

$$f_n = \sum_{i=1}^{s} \ell_i^d,$$

where the $\ell_i$'s are affine forms, then $s(d+1) \geq 2^n$.

*Proof.* Given a polynomial $f$, we let $\partial^*(f)$ denote the set of all partial derivatives of $f$ of all possible orders. For example, for the polynomial $f_n = x_1 x_2 \cdot \ldots \cdot x_n$ considered here, we have

$$\partial^*(f_n) = \left\{ \prod_{i \in S} x_i : S \subseteq [n] \right\}.$$

[1] The main result of this chapter, Theorem 10.4, is originally due to Saxena [Sax08]. The proof we present here is based on [Kay10].

The polynomials in $\partial^*(f_n)$ are all distinct monomials and are therefore linearly independent. We thus have

$$\dim\left(\partial^*(f_n)\right) = 2^n,$$

where $\dim\left(\partial^*(f_n)\right)$ denotes the number of $\mathbb{F}$-linearly independent polynomials in $\partial^*(f_n)$.

Now if $\ell$ is an affine form and $d \geq 0$ is an integer, then the partial derivatives of $\ell^d$ are all scalar multiples of $\ell^i$ for some $i \in [0 : d]$. Thus $\dim(\partial^*(\ell^d)) \leq (d+1)$. By linearity of derivatives, we have

$$\dim\left(\partial^*\left(\sum_{i\in[s]} \ell_i^d\right)\right) \leq s(d+1).$$

From the two equations above we get that $s(d+1) \geq 2^n$. □

The above lemma implies that if $d = \text{poly}(n)$ then $s$ must be exponential. But what if we allowed exponential $d$? It can be shown that to cancel the monomials of degree higher than $n$, $d$ cannot be too much larger than $s$.

**Exercise 10.3.** Let $\mathbb{F}$ be a field of characteristic zero. Suppose that $\ell_i$'s are coprime affine forms over $\mathbb{F}$ such that $\sum_{i=1}^{s} \ell_i^d$ is a polynomial of degree $m$, then $s \geq d - m$.

Hint:

(1) If two multilinear polynomials $f(x_1, \ldots, x_n)$ and $g(x_1, \ldots, x_n)$ are coprime then they are also coprime under a random substitution of the form $x_i := a_i \cdot t + b_i$, i.e. with high probability over a random choice of the $a_i$'s and $b_i$'s, the polynomials $f(a_1t + b_1, \ldots, a_nt + b_n)$ and $g(a_1t + b_1, \ldots, a_nt + b_n)$ are coprime as well.

(2) If $(a_1t + b_1), (a_2t + b_2), \ldots, (a_nt + b_n)$ are mutually coprime polynomials, then use the invertibilty of the Vandermonde matrix to deduce that

$$\sum_{i=1}^{n} (a_it + b_i)^d \neq 0 \quad \text{for } d > n.$$

Combining Lemma 10.2 with the above exercise we get the following:

**Theorem 10.4.** If we have

$$x_1 x_2 \cdot \ldots \cdot x_n = \sum_{i=1}^{s} \ell_i^d,$$

where the $\ell_i$'s are affine forms, then $s$ must be $2^{\Omega(n)}$.

Fischer [Fis94] gave an explicit set of $2^{n-1}$ linear forms such that $\prod x_i$ is the sum of their $n$-th powers.

### Arithmetic circuits with addition and powering gates

Now consider arithmetic circuits which have the usual addition gates of arbitrary fanin (as usual, addition gates can compute an arbitrary linear combination of its inputs) and *powering gates*. A powering gate takes as input $(f(\mathbf{X}), d)$ with $f(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ and $d \in \mathbb{Z}_{\geq 1}$, and outputs $f(\mathbf{X})^d$. Clearly such circuits are restricted versions of the usual arithmetic circuits. In the other direction, using the identity

$$x \cdot y = \left(\frac{x+y}{2}\right)^2 - \left(\frac{x-y}{2}\right)^2,$$

we see that they can simulate general arithmetic circuits. However, the computational power of *constant-depth* circuits with addition gates (of arbitrary fanin) and powering gates is not clear. We pose a problem:

**Open Problem 10.5.** Can polynomial-sized constant-depth circuits with addition and powering gates and polynomially-bounded degree compute the polynomial $x_1 x_2 \cdot \ldots \cdot x_n$?

### Depth-2 symmetric arithmetic circuits

Following Shpilka [Shp02]: a depth-2 symmetric arithmetic circuit has a symmetric gate computing the symmetric polynomial $\mathrm{SYM}_m^d$ at the top and addition gates at the bottom. So the polynomial computed by such a circuit is of the form:

$$\mathrm{SYM}_m^d(\ell_1, \ell_2, \ldots, \ell_m),$$

where $\ell_1, \ell_2, \ldots, \ell_m$ are linear functions. The relation of this model to depth-3 circuits is studied in Shpilka [Shp02]. However, even for this restricted model, we do not have a superlinear lower bound.

**Open Problem 10.6.** Can we prove a superlinear lower bound for depth-2 symmetric arithmetic circuits?

# 11

## Depth-3 arithmetic circuits

In this chapter, we look at one of the most restricted classes of arithmetic circuits for which we do not have superpolynomial lower bounds — circuits of depth three where each addition/multiplication gate can have arbitary fanin. We will see that if we further impose the additional (but apparently mild) restriction of homogenity on depth three circuits then partial derivatives help us prove exponential lower bounds.

In this chapter, we focus on a class of restricted arithmetic circuits — circuits of depth 3. More exactly, we consider circuits whose gates are divided into three levels with a plus (output) gate at the top, product gates in the middle, and plus gates at the bottom. We allow the fan-in to be unbounded (otherwise it would be too weak). An example is shown in Figure 1.1. We call them $\Sigma\Pi\Sigma$-circuits. We also use $\mathsf{S}^{\Sigma\Pi\Sigma}(f)$ to denote the minimum size of a $\Sigma\Pi\Sigma$-circuit that computes $f$.

Of course, one can also define $\Pi\Sigma$- and $\Sigma\Pi$-circuits similarly, but they are so weak that both $\mathsf{S}^{\Pi\Sigma}(f)$ and $\mathsf{S}^{\Sigma\Pi}(f)$ are very easy to determine. The other class of depth-3 circuits is the $\Pi\Sigma\Pi$-circuits. But when the $f$ to be computed is irreducible, they are as powerful as the $\Sigma\Pi$-circuits. To summarize, $\Sigma\Pi\Sigma$ is the simplest non-trivial class of arithmetic circuits. But even for this class of circuits we do not have strong lower bounds for many interesting polynomials.

**Open Problem 11.1.** Find an *explicit* polynomial which cannot be computed by a $\Sigma\Pi\Sigma$-circuit of polynomial size.

One of the best lower bounds for $\Sigma\Pi\Sigma$-circuits is the $\Omega(n^2)$ lower bound by Shpilka and Wigderson [SW01] for the symmetric polynomials $\mathrm{SYM}_n^d$. It matches the beautiful $O(n^2)$ upper bound by Ben-Or. In an earlier paper by Nisan and Wigderson [NW97], it is shown that, if there is a circuit computing $\mathrm{SYM}_n^d$, which is not only $\Sigma\Pi\Sigma$ but also *homogeneous* (see the definition below) then its size must be exponential.

**Definition 11.2.** We say a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ is homogeneous if all of its monomials are of the same degree.

We say an arithmetic circuit $\mathcal{C}$ is homogeneous if for every vertex $v$ in $\mathcal{C}$, $\mathcal{C}_v$ is a homogeneous polynomial.

Both of the proofs use partial derivatives as the main tool. In this chapter, we first present Ben-Or's construction and then prove the exponential lower bound for homogeneous $\Sigma\Pi\Sigma$-circuits, which is easier to understand. Besides, as the only interesting depth-3 circuits are of type $\Sigma\Pi\Sigma$, we will use $\mathsf{S}_3$ to denote $\mathsf{S}^{\Sigma\Pi\Sigma}$ and use $\mathsf{S}_3^H$ for homogeneous $\Sigma\Pi\Sigma$-circuits.

For now assume the underlying field to be $\mathbb{R}$. Ben-Or's construction is based on the following observation. Let

$$g(t) = \prod_{i=1}^{n} (x_i + t),$$

then one can write it as

$$g(t) = \sum_{k=0}^{n} \mathrm{SYM}_n^k \cdot t^{n-k}.$$

If we set $t$ to be a constant, say 1, then $g(1)$ is a linear combination of the $n+1$ symmetric polynomials $\mathrm{SYM}_n^k$. The good thing about $g(t)$ is that, given any $t$, $g(t)$ can be computed by a $\Pi\Sigma$-circuit of size $O(n)$. Therefore, we can compute $g(t)$ for every $t \in [0:n]$ with $O(n^2)$ gates.

Now if we can find a set of "magic" constants $\alpha_0, \ldots, \alpha_n$ such that in the summation $\sum_{i=0}^{n} \alpha_i \cdot g(i)$, the coefficient of every $\mathrm{SYM}_n^k$, where $k \neq d$, is 0 and the coefficient of $\mathrm{SYM}_n^d$ is 1, then we are done. Such a set of constants always exists because

$$\begin{pmatrix} 1 & 0^1 & 0^2 & \cdots & 0^n \\ 1 & 1^1 & 1^2 & \cdots & 1^n \\ 1 & 2^1 & 2^2 & \cdots & 2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & n^1 & n^2 & \cdots & n^n \end{pmatrix}$$

is a Vandermonde matrix and thus, has full rank. It is easy to extend the construction to large fields with at least $n$ nonzero elements.

Note that Ben-Or's circuit is not homogeneous. We now use partial derivatives to prove an exponential lower bound for $\mathsf{S}_3^H(\mathrm{SYM}_n^d)$ which shows that the use of non-homogeneous circuits in Ben-Or's construction is indeed necessary.

Given $f \in \mathbb{F}[\mathbf{X}]$, recall that $\partial^*(f)$ denotes the set of all partial derivatives of $f$ of all possible orders. We also use $\dim(\partial^*(f))$ to denote the dimension of the linear space spanned by $\partial^*(f)$. For example,

$$\dim\big(\partial^*(x_i)\big) = 2 \quad \text{and} \quad \dim\left(\partial^*\left(\prod_{i=1}^{n} x_i\right)\right) = 2^n.$$

It is easy to show that

**Property 11.3.** For $f, f_1, \ldots, f_k \in \mathbb{F}[x_1, \ldots, x_n]$ and $\alpha \in \mathbb{F}$, $\alpha \neq 0$,

(1) $\dim\big(\partial^*(\alpha f)\big) = \dim\big(\partial^*(f)\big)$;

(2) $\dim\big(\partial^*(\sum_{i=1}^{k} f_i)\big) \leq \sum_{i=1}^{k} \dim\big(\partial^*(f_i)\big)$; and

(3) $\dim\big(\partial^*(\prod_{i=1}^{k} f_i)\big) \leq \prod_{i=1}^{k} \dim\big(\partial^*(f_i)\big)$.

Now suppose there is a homogeneous $\Sigma\Pi\Sigma$-circuit $\mathcal{C}$ that computes $\mathrm{SYM}_n^{2d}$. Since $\mathcal{C}$ is homogeneous, we may assume that every product gate $v \in \mathcal{C}$ has $2d$ predecessors and $\mathcal{C}_v$ is of degree $2d$ (otherwise we can remove all the product gates $v \in \mathcal{C}$ whose polynomial $\mathcal{C}_v$ has degree not equal to $2d$, and the new and smaller circuit still computes $\mathrm{SYM}_n^{2d}$). Every predecessor of $v$ computes a *linear* combination of $x_i$'s.

Assume the fan-in of the output gate of $\mathcal{C}$ is $s$. Since

$$\dim\left(\partial^*(\textstyle\sum_{i=1}^n \alpha_i x_i)\right) \leq 2,$$

we have

$$\dim\left(\partial^*(\mathcal{C}_{\mathsf{out}})\right) \leq s \cdot 2^{2d},$$

using Proposition 11.3. A lower bound for $s$ immediately follows from Lemma 11.4 below, which states that $\mathrm{SYM}_n^{2d}$ has a very large derivative space.

**Lemma 11.4.** $\dim\left(\partial^*(\mathrm{SYM}_n^{2d})\right) \geq \binom{n}{d}$.

*Proof.* Let $\mathcal{S}$ denote the set of all $U \subset \{x_1, \ldots, x_n\}$ with $|U| = d$. We only need to show the following derivatives are linearly independent:

$$\left\{\partial_U(\mathrm{SYM}_n^{2d}) : U \in \mathcal{S}\right\}.$$

For any $V \in \mathcal{S}$, we let $x^V$ denote $\prod_{x_i \in V} x_i$. Then it is easy to see that

$$\partial_U(\mathrm{SYM}_n^{2d}) = \textstyle\sum_{V \in \mathcal{S}, U \cap V = \emptyset} x^V.$$

The set of monomials $x^V$ is clearly linearly independent. Therefore, to prove these $\binom{n}{d}$ derivatives are linearly independent, we only need to show that the following $\binom{n}{d} \times \binom{n}{d}$ matrix $\mathbf{M}$ has full rank: Every row (or column) of $\mathbf{M}$ corresponds to a set $U \in \mathcal{S}$ (or $V \in \mathcal{S}$); $M_{U,V} = 1$ if $U \cap V = \emptyset$; and $M_{U,V} = 0$ otherwise. A proof that $\mathbf{M}$ is nonsingular can be found in the text by Kushilevitz and Nisan[KN97, pp.22–23]. □

**Corollary 11.5.** $\mathsf{S}_3^H(\mathrm{SYM}_n^{2d}) \geq 2^{-2d}\binom{n}{d} \geq \left(n/(4d)\right)^d$.

# 12

## Arithmetic formulae

We turn our attention to *multilinear arithmetic formulae*, i.e. arithmetic circuits in which every gate computes a multilinear polynomial and has fanout exactly one. In this chapter we present a lower bound for constant depth *set-multilinear* formulae, a result which was a prelude to an elegant recent result of Raz on lower bounds for multilinear formulae.

A formula is an arithmetic circuit whose underlying graph is a tree. The root of the tree is then the output of the formula. In other words, each gate can be used as an input for at most once. Similarly, given a polynomial $f$, we let $\mathsf{L}(f)$ denote the size (that is, the number of edges of the underlying graph) of the smallest formula computing $f$.

Since formulae are much more restricted than general circuits, it is expected that for many polynomials $f$, $\mathsf{L}(f)$ should be much greater than $\mathsf{S}(f)$. For example while $\mathsf{S}(\textsc{Det}_n) = O(n^3)$, we do not even know whether it has a polynomial-size formula or not. The smallest formula known for $\textsc{Det}_n$ is of size $n^{O(\log n)}$. For *multilinear* formulae, Raz gave a beautiful $n^{\Omega(\log n)}$ lower bound for both $\textsc{Det}_n$ and $\textsc{Per}_n$ [Raz09].

**Open Problem 12.1.** Is $\mathsf{L}(\textsc{Det}_n)$ polynomial in $n$?

Other than its size, another important parameter of an arithmetic circuit is its depth since a circuit of low depth for $f$ indicates a highly parallel way to compute $f$ (note that we come back to general arithmetic circuits again). When talking about depth, we only consider circuits with bounded fanin.

The best depth upper bound for $\textsc{Det}_n$ so far is $O(\log^2 n)$ given by Csansky [Csa76] and Berkowitz [Ber84]. One particular way to understand Berkowitz's construction is to use the concept of *clow* (closed walk) *sequences* (see Soltys [Sol02] for the definition).

However, it turns out later that this $O(\log^2 n)$ depth upper bound for $\textsc{Det}_n$ is not accidental, but is a corollary of a much more general theorem. If we take the class of $f$ that has a polynomial $\mathsf{S}(f)$ as an analog of class $\mathsf{P}$ and the class of $f$ that has a $(\log n)^{O(1)}$ depth circuit as an analog of $\mathsf{NC}$, then we can ask the same "$\mathsf{P}$ vs $\mathsf{NC}$" question in the general arithmetic circuit setting. Very surprisingly, it was shown by Valiant, Skyum, Berkowitz, and Rackoff [VSBR83] that "$\mathsf{P} = \mathsf{NC}$" in this arithmetic setting:

**Theorem 12.2.** If $\mathsf{S}(f) = s$, then there exists a bounded fan-in circuit for $f$ that has size $s^{O(1)}$ and depth $O((\log s) \cdot (\log \deg(f)))$.

As a corollary, $\textsc{Det}_n$ can be computed by a $O(\log^2 n)$ depth circuit as $\mathsf{S}(\textsc{Det}_n) = O(n^3)$ and $\deg(\textsc{Det}_n) = n$. It also implies the following important relationship between $\mathsf{S}(f)$ and $\mathsf{L}(f)$:

**Theorem 12.3.** $\mathsf{L}(f) \leq \mathsf{S}(f)^{O(\log \deg(f))}$.

We will not give their construction here but leave the following as an exercise:

**Exercise 12.4.** Every $f$ has a formula of depth $O(\log \mathsf{L}(f))$.

In the rest of this chapter, we first present a framework of proving lower bounds for formulae. Then we combine it with partial derivatives to derive the lower bound of Nisan and Wigderson [NW97] concerning *set-multilinear* formulae that compute $\mathsf{IMM}^2_\mathsf{d}$, the iterated multiplication of $d$ $2 \times 2$ matrices.

## 12.1 Cover sets and measure functions

Let $\mathcal{C}$ be a formula over variables $\mathbf{X}$, and $T$ be the underlying tree of $\mathcal{C}$. Every vertex $v$ in $T$ computes a polynomial $\mathcal{C}_v \in \mathbb{F}[\mathbf{X}]$. For every $v$ in $T$, we use $S_v$ to denote the set of vertices in the subtree rooted at $v$ (including $v$). We now define *cover sets* of $v$ inductively as follows:

**Definition 12.5.** Let $v$ be a vertex in $T$, and $V \subseteq S_v$. If $v$ is a leaf of $T$, then $V$ *covers* $v$ if $V = \{v\}$. Otherwise, let $v_1, \ldots, v_k$ denote the predecessors of $v$, then $V$ *covers* $v$ if

(1) $v \in V$; or

(2) $v$ is a plus gate, and one can decompose $V$ into $V_1 \cup \ldots \cup V_k$ such that $V_i$ covers $v_i$ for all $i \in [k]$; or

(3) $v$ is a product gate, and there exists a subset of $V$ that covers one of the $v_i$'s, $i \in [k]$.

We say $V$ is a cover set of $\mathcal{C}$ if it covers $\mathsf{out}_{\mathcal{C}}$, the root of $T$. Cover sets are useful when they are combined with a *measure function* $\rho$. Here $\rho$ is a function from $\mathbb{F}[\mathbf{X}]$ (or a subset of $\mathbb{F}[\mathbf{X}]$, e.g., the set of multilinear polynomials) to $[0,1]$ that satisfies the following two properties:

$$\begin{aligned}&\text{Multiplicity: } \rho(fg) \leq \rho(f) \cdot \rho(g); \quad \text{and}\\ &\text{Additivity: } \rho(f+g) \leq \rho(f) + \rho(g), \quad \text{for all } f, g \in \mathbb{F}[\mathbf{X}].\end{aligned}$$

The following lemma shows that if one can find a cover set $V$ of $\mathcal{C}$ such that every polynomial $\mathcal{C}_v$ computed at $v$, $v \in V$, has a small $\rho(\mathcal{C}_v)$ then $\rho(\mathcal{C}_{\mathsf{out}})$ cannot be too large.

**Lemma 12.6.** If $V$ is a cover set of $\mathcal{C}$, then we have

$$\rho(\mathcal{C}_{\mathsf{out}}) \leq \textstyle\sum_{v \in V} \rho(\mathcal{C}_v). \tag{12.1}$$

*Proof.* We use induction on the depth of $T$. Let $v_1, \ldots, v_k$ denote the predecessors of $\mathsf{out}$. We now prove (12.1), assuming that if $V'$ covers $v_i$, for some $i \in [k]$, then

$$\rho(\mathcal{C}_{v_i}) \leq \textstyle\sum_{v\in V'} \rho(\mathcal{C}_v).$$

If $\mathsf{out}$ is a plus gate, then because $V$ covers $\mathsf{out}$, we can decompose the set $V$ into $V = V_1 \cup \ldots \cup V_k$ and $v_i$ is covered by $V_i$ for every $i \in [k]$. By induction,

$$\rho(\mathcal{C}_{\mathsf{out}}) \leq \sum_{i=1}^{k} \rho(\mathcal{C}_{v_i}) \leq \sum_{i=1}^{k} \sum_{v\in V_i} \rho(\mathcal{C}_v) = \sum_{v\in V} \rho(\mathcal{C}_v).$$

If $\mathsf{out}$ is a product gate, then there exists a subset $V' \subseteq V$ and an $\ell \in [k]$ such that $V'$ covers $v_\ell$. By induction, we have

$$\rho(\mathcal{C}_{\mathsf{out}}) \leq \prod_{i=1}^{k} \rho(\mathcal{C}_{v_i}) \leq \rho(\mathcal{C}_{v_\ell}) \leq \sum_{v\in V'} \rho(\mathcal{C}_v) \leq \sum_{v\in V} \rho(\mathcal{C}_v).$$

Here $\prod_{i=1}^{k} \rho(\mathcal{C}_{v_i}) \leq \rho(\mathcal{C}_{v_\ell})$ is because $\rho(\mathcal{C}_{v_i}) \in [0,1]$ for all $i$. □

Now let $f \in \mathbb{F}[\mathbf{X}]$ be the polynomial considered. Suppose there is a formula $\mathcal{C}$ that claims to compute $f$ and has size $s$, and we want to show that this cannot be true. A naive way to use this cover-measure approach is as follows: find a measure function $\rho$ together with a cover $V$ of $\mathcal{C}$ such that $\rho(f)$ is large (e.g., 1), but $\rho(\mathcal{C}_v)$ is very small (e.g., $< 1/s$) for all $v \in V$. Since $|V| \leq s$, we conclude that

$$\rho(\mathcal{C}_{\mathsf{out}}) \leq \textstyle\sum_{v\in V} \rho(\mathcal{C}_v) < 1 = \rho(f)$$

and $\mathcal{C}_{\mathsf{out}} \neq f$. However, this approach does not work well since given a general formula $\mathcal{C}$, one has no idea what polynomial $\mathcal{C}_v$ is computed at an intermediate vertex $v$ and thus, it could be very hard to prove an upper bound for $\rho(\mathcal{C}_v)$. One of the solutions is to combine this cover-measure approach with *random restriction*, which will be demonstrated in the next section.

## 12.2 A constant-depth lower bound

In [NW97], Nisan and Wigderson proved a lower bound for *set-multilinear* formulae of depth $h$ that compute $\mathsf{IMM}_d^2$ (the iterated multiplication of $d$ $2 \times 2$ matrices):

**Definition 12.7.** Let $\mathbf{X}^t = (x^t_{i,j})_{i,j\in[n]}$, where $t \in [d]$, denote $d$ $n \times n$ matrices with $dn^2$ variables.[1] We use

$$\mathsf{IMM}^n_d \in \mathbb{F}[\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^d]$$

to denote the $(1,1)$-th entry of the $n \times n$ matrix $\mathbf{X}^1\mathbf{X}^2 \cdots \mathbf{X}^d$.

**Definition 12.8.** Given a subset $S = \{t_1, \dots, t_s\} \subseteq [d]$ with $s = |S|$, an $S$-monomial is a product of $s$ variables, one from each $\mathbf{X}^{t_i}$:

$$x^{t_1}_{i_1,j_1} \cdot \ldots \cdot x^{t_s}_{i_s,j_s}, \qquad \text{where } i_1, j_1, \dots, i_s, j_s \in [n].$$

We use $P_S$ to denote the set of all $S$-monomials.

A polynomial $f$ is said to be *set-multilinear* if there exists a subset $S \subseteq [d]$ such that $f$ is a linear combination of monomials in $P_S$.

A formula $\mathcal{C}$ over $\mathbb{F}[\mathbf{X}^1 \dots \mathbf{X}^d]$ is said to be *set-multilinear* if the polynomial $\mathcal{C}_v$ computed at any vertex $v$ of $\mathcal{C}$ is set-multilinear.

**Theorem 12.9.** Every depth-$h$ set-multilinear formula that computes $\mathsf{IMM}^2_d$ has size $2^{\Omega(d^{1/h})}$.

Theorem 12.9 implies that if $h$ is a constant, then the size of any depth-$h$ set-multilinear formula that computes $\mathsf{IMM}^2_d$ must be exponential in $d$. It also implies that any polynomial-size set-multilinear formula for $\mathsf{IMM}^2_d$ must have depth $\Omega(\log d/ \log\log d)$. We note that these lower bound results were strengthened to hold even for the more general model of multilinear formulae by Raz and Yehudayoff [RY08]. We start the proof of Theorem 12.9 by introducing a measure function $\rho$, from set-multilinear polynomials to $[0,1]$. By definition, if $\mathcal{C}$ is set-multilinear, then one can assign a set $S_v \subseteq [d]$ to each vertex $v$ such that $\mathcal{C}_v$ is $S_v$-multilinear.

[1] In this chapter superscripts will denote indices of the different sets of variables (and not powers). As we deal here with multilinear polynomials, no confusion should arise.

Now let $f$ be any $S$-multilinear polynomial, and $W = \{w_1, \ldots, w_k\}$ be a subset of $S$. We let $\partial_W(f)$ denote the set of partial derivatives of $f$ with respect to all possible monomials in $P_W$:

$$\partial_W(f) \stackrel{\text{def}}{=} \left\{ \partial_{x_{i_1,j_1}^{w_1} \ldots x_{i_k,j_k}^{w_k}}(f) : x_{i_1,j_1}^{w_1} \ldots x_{i_k,j_k}^{w_k} \in P_W \right\}.$$

We use $\dim_W(f)$ to denote the dimension of the linear space spanned by the polynomials in $\partial_W(f)$. The following upper bound for $\partial_W(f)$ is easy to prove:

**Lemma 12.10.** Let $f$ be an $S$-multilinear polynomial, and $W \subseteq S$. Then we have

$$\dim_W(f) \le \min\left\{n^{2|W|}, n^{2|S-W|}\right\} \le n^{2\lfloor |S|/2 \rfloor}.$$

Let $\dim(f) = \max_{W \subseteq S}\left[\dim_W(f)\right]$, then we have:

**Lemma 12.11.** For $S_1, S_2 \subseteq [d]$ with $S_1 \cap S_2 = \emptyset$, let $f, g$ be two $S_1$-multilinear polynomials and $h$ be an $S_2$-multilinear polynomial. Then

(1) $\dim(f) \le n^{2\lfloor |S_1|/2 \rfloor}$;
(2) $\dim(\alpha f) = \dim(f)$ for all $\alpha \in \mathbb{F}$;
(3) $\dim(f + g) \le \dim(f) + \dim(g)$; and
(4) $\dim(f \cdot h) \le \dim(f) \cdot \dim(h)$.

Now we can define $\rho$ by normalizing $\dim(f)$: given an $S$-multilinear $f$,

$$\rho(f) \stackrel{\text{def}}{=} \frac{\dim(f)}{n^{2\lfloor |S|/2 \rfloor}}.$$

Clearly $\rho$ is a measure function because it satisfies both additivity and multiplicity. Moreover, one can show that $\rho(\mathsf{IMM}_d^n)$ is very close to 1:

**Lemma 12.12.** For odd $d$, $\rho(\mathsf{IMM}_d^n) = 1$; for even $d$, $\rho(\mathsf{IMM}_d^n) = 1/n$.

Furthermore, $\rho$ satisfies the following important property. We will use it to argue that after certain random restriction, with high probability, there exists a cover set of $\mathcal{C}$, in which every vertex $v$ has a very small $\rho$ value.

**Property 12.13.** Let $S_1, \ldots, S_k$ be pairwise disjoint subsets of $[d]$, and $f_i$ be an $S_i$-multilinear polynomial, $i \in [k]$. If $m$ of these $k$ subsets $S_1, \ldots, S_k$ are of odd size, then

$$\rho\left(\prod_i f_i\right) \leq n^{-2\lfloor m/2\rfloor} \prod_i \rho(f_i) \leq n^{-2\lfloor m/2\rfloor} \min_i \rho(f_i).$$

*Proof.* Using $\dim(f \cdot g) \leq \dim(f) \cdot \dim(g)$, we have

$$\rho\left(\prod_i f_i\right) \leq \frac{\prod_i \dim(f_i)}{n^{2\lfloor(\sum_i |S_i|)/2\rfloor}} \leq \frac{\prod_i n^{2\lfloor |S_i|/2\rfloor} \cdot \prod_i \rho(f_i)}{n^{2\lfloor(\sum_i |S_i|)/2\rfloor}} = \frac{\prod_i \rho(f_i)}{n^{2\lfloor m/2\rfloor}}.$$

□

It implies that if $f = \prod_i f_i$, and many of the $f_i$'s are odd-multilinear, then $\rho(f)$ would be very small (regardless of what these $f_i$'s are). Now suppose $\mathcal{C}$ is a depth-$h$ formula for $\mathsf{IMM}_d^2$ ($n = 2$) and has size $s = 2^{\epsilon r}$, where $r = d^{1/h}$ and $\epsilon > 0$ is a small enough constant. We then apply the following random restriction to $\mathcal{C}$:

> Let $z_1, \ldots, z_d$ be $d$ independent, unbiased $\{0, 1\}$ random variables. For each $i$, set $\mathbf{X}^i$ to be the identity matrix if $z_i = 0$.

We use $\mathcal{C}'$ to denote the formula we get from $\mathcal{C}$, after this random restriction. First, as $\mathcal{C}$ is assumed to compute $\mathsf{IMM}_d^2$, $\mathcal{C}'$ must compute $\mathsf{IMM}_{d'}^2$ (where $d'$ is close to $d/2$ w.h.p.) and thus, $\rho(\mathcal{C}'_{\mathsf{out}}) \geq 1/2$. So to get a contradiction, we only need to show the existence of a cover set $V$ in $\mathcal{C}'$ such that for every $v \in V$, $\rho(\mathcal{C}'_v)$ is very small.

The intuition is that, since $\mathcal{C}$ is of depth $h$, by using a degree argument, there must be a lot of product gates with large fanin ($\geq r$). So one can hope that there exists a cover $V$ of $\mathcal{C}$ (note that if $V$ is a cover of $\mathcal{C}$, then it is also a cover of $\mathcal{C}'$) that consists of product gates with large fanin only.

On the other hand, if a product gate has large fanin ($\geq r$) in $\mathcal{C}$ then after the random restriction, w.h.p. ($\geq 1 - 2^{-r/10}$), the number of its predecessors that are odd-multilinear is large ($\geq r/3$) since every predecessor becomes odd-multilinear *independently* with probability $1/2$.

As a result, by Property 12.13, the values of $\rho$ at these gates must be very small ($\leq 2^{-2\lfloor r/6 \rfloor}$).

Note that $|V| \leq s = 2^{\epsilon r}$. When $\epsilon$ is a small enough constant, one can apply union bound to show that with positive probability, all the product gates in $V$ have $\rho$ value at most $2^{-2\lfloor r/6 \rfloor}$. So by Lemma 12.6

$$\rho(\mathcal{C}'_{\mathsf{out}}) \leq s \cdot 2^{-2\lfloor r/6 \rfloor} = 2^{\epsilon r} \cdot 2^{-2\lfloor r/6 \rfloor} \ll 1/2,$$

and we get a contradiction.

Below are the two technical lemmas we need to finish the proof of Theorem 12.9. First, Lemma 12.14 shows the existence of a cover set $V$ in any depth-$h$ formula $\mathcal{C}$ for $\mathsf{IMM}_d^2$, which consists of large fan-in product gates only. Then Lemma 12.15 shows that if a product gate $v$ has large fan-in then after the random restriction, $\rho(\mathcal{C}'_v)$ is small with high probability. Lemma 12.15 follows from the Chernoff bound.

**Lemma 12.14.** There exists a cover set $V$ of $\mathcal{C}$ such that every $v$ in $V$ is a product gate and has fan-in at least $r = d^{1/h}$.

*Proof.* Let $u_1 \ldots u_k$ be a path in $\mathcal{C}$ where $u_1 = \mathsf{out}$ and $u_k$ is a leaf. We say $u_1 \ldots u_k$ is a *good* path, if for every product gate $u_i$ on the path, $u_{i+1}$ has the largest $|S_{u_{i+1}}|$ among all $u_i$'s predecessors. Recall that $S_{u_i}$ is a subset of $[d]$ associated with $u_i$ such that $\mathcal{C}_{u_i}$ is $S_{u_i}$-multilinear.

It is easy to show that on every good path $u_1 \ldots u_k$ of $\mathcal{C}$, there must exist a product gate $u_i$ with fanin at least $d^{1/h}$ (since $|S_{\mathsf{out}}| = d$ and $k \leq h$). Now for every good path of $\mathcal{C}$, we arbitrarily pick a product gate $u_i$ with fanin at least $d^{1/h}$. One can show that the set formed by these product gates is indeed a cover set of $\mathcal{C}$. □

**Lemma 12.15.** Let $z_1, \ldots, z_d$ be $d$ independent and unbiased $\{0, 1\}$ random variables. Given any $S \subseteq [d]$, we let $z(S) = \sum_{i \in S} z_i \pmod 2$. Then for any nonempty pairwise disjoint subsets $S_1, \ldots, S_r$ of $[d]$,

$$\mathbf{Pr}\left[\sum_i z(S_i) < r/3\right] \leq 2^{-r/10}.$$

# 13

## Projections of Determinant to Permanent

The arithmetic analog of the celebrated P versus NP problem is the following: what is the smallest integer $m$ such that the permanent polynomial can be expressed as the determinant of an $m \times m$ matrix whose entries are affine forms in the relevant set of variables. It is known as the Permanent versus Determinant problem in arithmetic complexity. In this chapter, we use the Hessian matrix consisting of second-order partial derivatives introduced in chapter 2 to derive a quadratic lower bound on $m$.

Determinant and permanent are two of the most well-studied polynomials in theoretical computer science. In Valiant's theory of arithmetic complexity [Val79b], they are two of the central objects. The complexity of computing the permanent characterizes the class VNP, while the complexity of computing the determinant (almost) characterizes the class VP. Since VNP and VP are analogs of NP and P in the arithmetic world, the following problem, called "Permanent versus Determinant", has received great attention.

**Definition 13.1.** Let $\mathbf{X}$ be an $n \times n$ matrix with $n^2$ variables: $(x_{i,j})$

$i, j \in [n]$. We say $\mathbf{A}$ is an affine projection from $\textsc{Per}_n$ to $\textsc{Det}_m$ over $\mathbb{F}$ if $\mathbf{A} = \{A_{k,\ell} : k, \ell \in [m]\}$ is a collection of affine forms over $\mathbf{X}$:

$$A_{k,\ell} = \textstyle\sum_{i,j\in[n]} \alpha_{i,j} \cdot x_{i,j} + \alpha, \qquad \text{for some } \alpha_{i,j}, \alpha \in \mathbb{F},$$

such that

$$\textsc{Per}_n(\mathbf{X}) \equiv \textsc{Det}_m\big(\mathbf{A}(\mathbf{X})\big)$$

over the polynomial ring $\mathbb{F}[\mathbf{X}]$.

We let $\mathsf{dc}(\textsc{Per}_n)$ denote the smallest integer $m$ such that an *affine* projection from $\textsc{Per}_n$ to $\textsc{Det}_m$ exists. We can similarly define $\mathsf{dc}(f)$, for any $f \in \mathbb{F}[\mathbf{X}]$. It is called the *determinantal complexity* of $f$.

**Exercise 13.2.** Show that $\mathsf{dc}(f) \le \mathsf{L}(f) + 1$.
(Hint: Use induction. Make sure that all the matrices constructed inductively are "almost" upper triangular, and have the following form:

$$\begin{pmatrix} \mathbf{u} & 0 \\ \mathbf{B} & \mathbf{v} \end{pmatrix},$$

where $\mathbf{u}$ is a row vector, $\mathbf{v}$ is a column vector and $\mathbf{B}$ is upper triangular with 1's on its diagonal.)

Using Ryser's formula [Rys63] together with Exercise 13.2, we have

$$\mathsf{dc}(\textsc{Per}_n) = O(n^2 \cdot 2^n).$$

On the other hand, it was first noticed by Pólya [Pól71] that

$$\textsc{Per}_2 \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \textsc{Det}_2 \begin{pmatrix} a & -b \\ c & d \end{pmatrix}.$$

He asked whether there are other similar equations. It was answered in the negative by Szegö [Sze92], which implies that $\mathsf{dc}(\textsc{Per}_n) \ge n + 1$ for all $n > 2$. The first non-trivial lower bound ($\sqrt{8/7}n$) for $\mathsf{dc}(\textsc{Per}_n)$ is due to von zur Gathen [vzG87]. It was then improved by Babai and Seress, by Cai [Cai90], and by Meshulam [Mes89], to $\sqrt{2}n$.

An approach to this problem, which is now referred to as the Geometric Complexity Theory, uses some exceptional properties of determinant and permanent to translate the question into a problem in the representation theory of groups. Interested readers are referred to the survey articles [Mul09b, Mul09a, BLMW09] and to the series of papers beginning with [Mul99, MS01, MS08].

In [MR04], Mignon and Ressayre proved a quadratic lower bound for $\mathsf{dc}(\textsc{Per}_n)$, which is the best known bound for this problem. The main idea is to study the Hessian matrices of $\textsc{Det}$ and $\textsc{Per}$.

Given $\mathbf{X} = (x_{i,j})_{i,j\in[n]}$ of $n^2$ variables, we use $\textsc{Det}_n(\mathbf{X})$ to denote the determinant and $\textsc{Per}_n(\mathbf{X})$ to denote the permanent of $\mathbf{X}$, respectively. Both of them are polynomials of degree $n$ in $\mathbb{F}[\mathbf{X}]$. We use

$$\mathbf{H}^{\textsc{Per}_n}(\mathbf{X}) = \big(H_{ij,k\ell}\big)_{i,j,k,\ell\in[n]}$$

to denote the Hessian matrix of $\textsc{Per}_n(\mathbf{X})$:

$$H_{ij,k\ell} = \frac{\partial^2 \textsc{Per}_n(\mathbf{X})}{\partial x_{i,j}\,\partial x_{k,\ell}} \in \mathbb{F}[\mathbf{X}], \qquad \text{for all } i,j,k,\ell \in [n].$$

Similarly, we can define the Hessian matrix $\mathbf{H}^{\textsc{Det}_m}$ of $\textsc{Det}_m$.

Suppose there exists a collection $\mathbf{A}$ of $m^2$ affine maps, where

$$\mathbf{A} = \big\{A_{k,\ell}(x_{1,1}, x_{1,2}, \ldots, x_{n,n}) : k,\ell \in [m]\big\}$$

such that in the polynomial ring $\mathbb{F}[\mathbf{X}]$,

$$\textsc{Per}_n(\mathbf{X}) = \textsc{Det}_m\left(\big(A_{k,\ell}(\mathbf{X})\big)_{k,\ell\in[m]}\right). \tag{13.1}$$

Now consider a scalar matrix $\mathbf{M} \in \mathbb{F}^{n\times n}$, satisfying $\textsc{Per}_n(\mathbf{M}) = 0$. The first step of the proof is to transform $\mathbf{A}$ into a new form, while maintaining the property that $\textsc{Per}_n(\mathbf{X}) = \textsc{Det}_m(\mathbf{A}(\mathbf{X}))$, such that $\mathbf{A}(\mathbf{M})$ is a diagonal matrix in $\mathbb{F}^{m\times m}$.

To this end, we expand every $A_{k,\ell}(\mathbf{X})$ at $\mathbf{M}$ and write $\mathbf{A}$ as

$$\mathbf{A}(\mathbf{X}) = \big(A_{k,\ell}(\mathbf{X})\big) = \big(L_{k,\ell}(\mathbf{X} - \mathbf{M})\big) + \mathbf{N}, \tag{13.2}$$

where $L_{k,\ell}$'s are *linear* functions and $\mathbf{N} \in \mathbb{F}^{m\times m}$. It follows that

$$\textsc{Det}_m(\mathbf{N}) = \textsc{Per}_n(\mathbf{M}) = 0.$$

As a result, we can find two non-singular matrices $\mathbf{C}$ and $\mathbf{D}$, such that $\textsc{Det}_m(\mathbf{C}) = \textsc{Det}_m(\mathbf{D}) = 1$ and $\mathbf{CND}$ is a diagonal matrix

$$\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_s \end{pmatrix}, \quad \text{where } \mathbf{J}_s \text{ is an } s\times s \text{ diagonal matrix.}$$

Since $\textsc{Det}_m(\mathbf{N}) = 0$ there is at least one 0 on the diagonal and $s < m$.

It then follows that, by (multiplying $\mathbf{C}$ and $\mathbf{D}$ to the left and right of (13.2) and) renaming $L_{k,\ell}$, we may assume (13.1) takes the form

$$\textsc{Per}_n(\mathbf{X}) = \textsc{Det}_m \left( \big( L_{k,\ell} \left( \mathbf{X} - \mathbf{M} \right) \big)_{k,\ell \in [m]} + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_s \end{pmatrix} \right).$$

Next, we take second-order derivatives, and evaluate $\mathbf{H}^{\textsc{Per}_n}$ at $\mathbf{M}$. By the chain rule (Lemma 2.7), we have

$$\mathbf{H}^{\textsc{Per}_n}(\mathbf{M}) = \mathbf{L} \cdot \mathbf{H}^{\textsc{Det}_m} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_s \end{pmatrix} \cdot \mathbf{L}^T,$$

where $\mathbf{L}$ is an $n^2 \times m^2$ scalar matrix over $\mathbb{F}$. It follows directly that

$$\operatorname{rank}\big(\mathbf{H}^{\textsc{Per}_n}(\mathbf{M})\big) \leq \operatorname{rank}\left( \mathbf{H}^{\textsc{Det}_m} \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_s \end{pmatrix} \right). \tag{13.3}$$

It is easy to obtain an upper bound for the latter, using the fact that the matrix is diagonal. Let us first assume $s = m - 1$. Note that when one takes a partial derivative $\partial_{x_{i,j}}$ on the determinant (as well as on the permanent), one simply gets the minor after striking out row $i$ and column $j$. Second-order derivative $\partial_{x_{i,j}x_{k,\ell}}$ simply strikes out row $\{i, k\}$ and column $\{j, \ell\}$. For the diagonal matrix, if the $\{ij, k\ell\}^{th}$ entry of $\mathbf{H}^{\textsc{Det}_m}$ is nonzero then it must be that $1 \in \{i, k\}$ and $1 \in \{j, \ell\}$. In fact the only nonzero entries are

$$(ij, k\ell) = (11, tt), (tt, 11), (1t, t1) \text{ or } (t1, 1t), \quad \text{for } t \in [2 : m].$$

This gives an upper bound of $2m$ for the right-hand side of (13.3).

Moreover, if $s < m - 1$ then it would be even more difficult to get a nonzero entry in $\mathbf{H}^{\textsc{Det}_m}$. For example, if $s = m - 2$, then there could be at most $O(1)$ many nonzero entries; if $s < m - 2$, then there are no nonzero entries. As a result, we obtain the following lemma:

**Lemma 13.3.** Let $\mathbf{M}$ be a matrix in $\mathbb{F}^{n \times n}$ with $\textsc{Per}_n(\mathbf{M}) = 0$, then

$$\mathsf{dc}(\textsc{Per}_n) \geq \operatorname{rank}\big(\mathbf{H}^{\textsc{Per}_n}(\mathbf{M})\big)/2.$$

Now it is clear that the only thing left is to find a matrix $\mathbf{M}$ such that $\textsc{Per}_n(\mathbf{M}) = 0$ and $\mathbf{H}^{\textsc{Per}_n}(\mathbf{M})$ has high rank. When $\mathbb{F}$ is of characteristic 0, we let $\mathbf{M}_n = (M_{i,j})$ denote the $n \times n$ matrix where $M_{1,1} = 1 - n$ and $M_{i,j} = 1$ otherwise. While $\textsc{Per}_n(\mathbf{M}_n) = 0$ is trivial, we leave the computation of the rank of $\mathbf{H}^{\textsc{Per}_n}(\mathbf{M}_n)$ as an exercise for the reader.

**Lemma 13.4.** (Mignon and Ressayre [MR04]) For any field $\mathbb{F}$ of characteristic 0,

$$\textsc{Per}_n(\mathbf{M}_n) = 0 \quad \text{and} \quad \operatorname{rank}\big(\mathbf{H}^{\textsc{Per}_n}(\mathbf{M}_n)\big) = n^2.$$

It immediately follows from Lemma 13.3 that

**Corollary 13.5.** For any $\mathbb{F}$ of characteristic 0, $\mathsf{dc}(\textsc{Per}_n) \geq n^2/2$.

However, the matrix $\mathbf{M}_n$ above does not work for fields $\mathbb{F}$ of small characteristics, e.g., 3. All entries of $\mathbf{H}^{\textsc{Per}_n}(\mathbf{M}_n)$ are divisible by large factorials and thus, divisible by char $\mathbb{F}$. As a result, $\mathbf{H}^{\textsc{Per}_n}(\mathbf{M}_n)$ is the zero matrix of rank 0. In [CCL08] a different family of matrices is used to extend Mignon and Ressayre's quadratic lower bound to all finite fields.

At first sight, it might seem very hopeful to try and extend this approach using higher-order derivatives, and to prove stronger lower bounds for $\mathsf{dc}(\textsc{Per}_n)$. However, note that in terms of the number of variables in $\textsc{Per}_n$, the lower bound is "only" linear. Such bounds on determinantal complexity can be obtained for much simpler functions than the permanent, e.g. the 2nd symmetric polynomial, as the following exercise shows. This exercise, in some sense, implies that the quadratic lower bound mainly follows from the limitation of the determinant, but does not take much advantage of the hardness of the permanent.

**Exercise 13.6.** Show that $\mathsf{dc}\big(\sum_{1 \leq i \neq j \leq n} x_i x_j\big) = \Omega(n)$.

**Open Problem 13.7.** Prove a super-linear lower bound on $\mathsf{dc}(f)$ for any explicit polynomial $f$.

# Part III: Algorithms

## Overview

Suppose that we are given an arithmetic circuit $\mathcal{C}$ whose inputs are $x_1$, $\ldots, x_n$. We use $\mathcal{C}(x_1,\ldots,x_n)$, abbreviated as $\mathcal{C}(\mathbf{X})$, to denote the polynomial computed by the circuit $\mathcal{C}$. In this part, we will see how partial derivatives help us in designing efficient algorithms to understand some basic algebraic properties of $\mathcal{C}(\mathbf{X})$.

In Chapter 14 we will define the "Identity Testing" problem of testing if the given polynomial $\mathcal{C}(\mathbf{X})$ is identically zero. This is a central problem in arithmetic complexity, and in a strong sense "complete" for derandomization. Here we will use partial derivatives to give a deterministic algorithm for a certain special case of it.

In Chapter 15 we turn to another basic problem: checking if $\mathcal{C}(\mathbf{X})$ is absolutely irreducible, namely that $\mathcal{C}(\mathbf{X})$ cannot be factored over the algebraic closure of its field of coefficients. Partial derivatives will lead to an efficient deterministic algorithm in the bivariate case and a probabilistic algorithm (via Hilbert's irreducibility theorem) in the general case.

In Chapter 16 we deal with the polynomial equivalence (or "isomorphism") problem, namely testing if the two polynomials computed by two given circuits can be made identical by an invertible linear transformation of the variables. Again, partial derivatives lead to deterministic algorithms in special cases.

We shall henceforth assume that the underlying field $\mathbb{F}$ has characteristic zero. This is mostly for ease of presentation. All results and their proofs remain unchanged when the characteristic is larger than say $d^2$, where $d$ is the degree of the input polynomial.

# 14

## Identity testing

Given an arithmetic circuit, how efficiently can we determine (deterministically) whether the polynomial computed is the zero polynomial or not? This is the famous polynomial identity testing problem. In this chapter, we will see how partial derivatives (combined with substitutions) help us devise efficient identity testing algorithms for a certain restricted class of arithmetic circuits.

Given a circuit $\mathcal{C}(\mathbf{X})$, a basic question is:

**Question 14.1.** Is $\mathcal{C}(\mathbf{X}) \equiv 0$?

The computational version of this question is commonly known as *Identity Testing*. It admits an efficient randomized algorithm [Sch80, Zip90]. Despite much effort, no deterministic polynomial-time algorithm is yet known for identity testing. [CK00, LV98, KS01, AB03, RS04, DS05, KS06, KS07, SV08, BHLV09, SS09, KS09] is a partial list of references on this problem. The problem is also connected to arithmetic circuit lower bounds (cf. [KI04, Agr05]). Because of the difficulty of the general problem, research has focussed on special cases. We will present a deterministic algorithm for a special family of polynomials — polynomials of the form

$$\sum_{i=1}^{m} \big(g_{i1}(x_1) + g_{i2}(x_2) + \ldots + g_{in}(x_n)\big)^{D_i}, \tag{14.1}$$

where each $g_{ij}(x_j)$ is a univariate polynomial of degree at most $d$. The running time will be $\mathrm{poly}(nmdD)$, where $D = \max_i D_i$. Our presentation is based on [Kay10].

We start this chapter by defining a problem called POLYDEP and then giving its relation to the problem of polynomial identity testing.

### POLYDEP and its connection to identity testing

We formulate a computational problem motivated by a concept (definition 10.1) which had featured in some of the lower bound proofs in previous chapters.

**Definition 14.2.** Let

$$\mathbf{f}(\mathbf{X}) \stackrel{\text{def}}{=} \big(f_1(\mathbf{X}), f_2(\mathbf{X}), \ldots, f_m(\mathbf{X})\big) \in \big(\mathbb{F}[\mathbf{X}]\big)^m$$

be a vector of $m$ polynomials over a field $\mathbb{F}$. The set of $\mathbb{F}$-*linear dependencies* in $\mathbf{f}$, denoted $\mathbf{f}^{\perp}$, is the set of all vectors $\mathbf{v} \in \mathbb{F}^m$ whose inner product with $\mathbf{f}$ is the zero polynomial, i.e.,

$$\mathbf{f}^{\perp} \stackrel{\text{def}}{=} \big\{(a_1, \ldots, a_m) \in \mathbb{F}^m : a_1 f_1(\mathbf{X}) + \ldots + a_m f_m(\mathbf{X}) = 0\big\}.$$

Notice that if $\mathbf{f}^{\perp}$ contains a nonzero vector, then the $f_i$'s are said to be $\mathbb{F}$-linearly dependent. The set $\mathbf{f}^{\perp}$ is clearly a linear subspace of $\mathbb{F}^m$. In many of our applications, we will want to efficiently compute a basis of $\mathbf{f}^{\perp}$ for a given tuple $\mathbf{f} = (f_1(\mathbf{X}), \ldots, f_m(\mathbf{X}))$ of polynomials. Let us capture this as a computational problem.

**Definition 14.3.** The problem of computing linear dependencies between polynomials, denoted POLYDEP, is defined to be the following computational problem: given as input $m$ arithmetic circuits computing polynomials $f_1(\mathbf{X}), \ldots, f_m(\mathbf{X})$ respectively, output a basis for the subspace $(f_1(\mathbf{X}), \ldots, f_m(\mathbf{X}))^{\perp} \subseteq \mathbb{F}^m$.

We state without proof the following lemma from [Kay10] which shows the connection between POLYDEP and identity testing.

**Lemma 14.4.** For $n$-variate arithmetic circuits of degree bounded by $\mathrm{poly}(n)$, identity testing is deterministic polynomial-time *equivalent* to POLYDEP. In particular, POLYDEP admits a randomized algorithm with running time polynomial in the arithmetic circuit size of the input polynomials.

### Basic properties of POLYDEP

Even though POLYDEP is equivalent to identity testing, it is a somewhat more structured problem which makes it more amenable to attack. We state without proof the following two properties of the POLYDEP problem. Note that the second property gives a basic form of reduction between two POLYDEP problems.

**Proposition 14.5.** Let $\mathbf{f} = (f_1(\mathbf{X}), \ldots, f_m(\mathbf{X})) \in \mathbb{F}[\mathbf{X}]^m$ and $\mathbf{h} = (h_1(\mathbf{X}), \ldots, h_t(\mathbf{X})) \in \mathbb{F}[\mathbf{X}]^t$ be two tuples of polynomials. Then:

(1) Testing zeroness of

$$\sum_{i \in [m]} a_i \cdot f_i(\mathbf{X})$$

reduces to solving $\mathrm{POLYDEP}(f_1, \ldots, f_m)$.

(2) If the $f_i$'s are known $\mathbb{F}$-linear combinations of the $h_j$'s then $\mathrm{POLYDEP}(\mathbf{f})$ reduces to $\mathrm{POLYDEP}(\mathbf{h})$. In other words, if we know a set of vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m \in \mathbb{F}^t$ such that

$$f_i(\mathbf{X}) = \mathbf{a}_i \cdot \mathbf{h} \quad \text{for each } 1 \leq i \leq m$$

then $\mathrm{POLYDEP}(\mathbf{f})$ reduces to $\mathrm{POLYDEP}(\mathbf{h})$.

We now focus our efforts on POLYDEP and devise an efficient deterministic algorithm for $\mathrm{POLYDEP}(f_1(\mathbf{X}), \ldots, f_m(\mathbf{X}))$ when each polynomial $f_i(\mathbf{X})$ is a *power of a sum of univariate polynomials*, i.e., for every $i \in [m]$:

$$f_i(\mathbf{X}) = \big(g_{i1}(x_1) + g_{i2}(x_2) + \ldots + g_{in}(x_n)\big)^{D_i},$$

and each $g_{ij}(x_j)$ is a univariate polynomial of degree at most $d$. For a set of polynomials of this form the brute force algorithm that expands every polynomial takes time about $m \cdot n^{d+D}$. Below we will present the poly$(nmdD)$-time algorithm given in [Kay10].

A key idea is that partial derivatives can be used to reduce a given instance of the POLYDEP problem into a large number of instances of the POLYDEP problem, each of which has one variable less. Specifically, to compute the dependencies of a set of multivariate polynomials, it suffices to compute these for the set of their coefficients of $x_1^j$ (which are polynomials in the remaining $n-1$ variables), for each power $j$.

**Proposition 14.6.** Let $\mathbf{f} = (f_1(\mathbf{X}), \ldots, f_m(\mathbf{X}))$. Suppose that the degree of each $f_i(\mathbf{X})$ is at most $d$. Then we have

$$\mathbf{f}^{\perp} = \bigcap_{j=0}^{d} V_j,$$

where for each $j \in \{0, 1, \ldots, d\}$, $V_j$ is defined to be

$$V_j \stackrel{\text{def}}{=} \left(\sigma_1 \partial_1^j f_1, \ldots, \sigma_1 \partial_1^j f_m\right)^{\perp}.$$

The proof is a straightforward generalization of Lemma 1.11 — we omit the details.

### The algorithm

Proposition 14.6 above suggests a natural strategy for POLYDEP — recursively compute the dependencies among

$$\left(\sigma_1 \partial_1^j f_1, \ldots, \sigma_1 \partial_1^j f_m\right)$$

for $j \in \{0, \ldots, d\}$, and then take the intersection of all the subspaces so obtained. This naïve strategy in general only gives an exponential algorithm that is little better than brute force expansion. However, for the case when $f_i$'s are the powers of sums of univariate polynomials, one can show that all the polynomials in

$$\left\{\sigma_1 \partial_1^j f_k : 0 \leq j \leq d, 1 \leq k \leq m\right\}$$

are efficiently expressible as linear combinations of a small number of polynomials $\{h_1, h_2, \ldots, h_t\} \subseteq \mathbb{F}[x_2, \ldots, x_n]$ where $h_i$'s are also powers of sums of univariate polynomials. Then *using just one recursive call* POLYDEP$(h_1, \ldots, h_t)$ and using the second property in Proposition 14.5, we can compute each $V_j$ for $0 \leq j \leq d$. Thereafter, taking the intersection of the $V_j$'s allows us to compute the linear dependencies between the original polynomials. The algorithm is efficient because in the inductive step we make just one recursive call to POLYDEP rather than $(d+1)$ calls.

**Lemma 14.7.** Let $f(x_1, \ldots, x_n) = (g_1(x_1) + \ldots + g_n(x_n))^D$, where the $g_i$'s are univariate polynomials of degree at most $d$. Let $G = g_1(x_1) + g_2(x_2) + \ldots + g_n(x_n)$. Then

$$\sigma_1 \partial_1^j f = \sum_{k=0}^{j} a_{jk} \cdot \sigma_1 G^{D-k}, \qquad \text{for some } a_{jk} \in \mathbb{F}.$$

Furthermore, the $a_{jk}$'s occurring in the above expression can be computed in time poly$(dD)$.

*Proof.* It suffices to prove that

$$\partial_1^j f = \sum_{k=0}^{j} \sum_{\ell=0}^{(d-1)j} b_{k\ell} \cdot G^{D-k} \cdot x_1^{\ell}, \tag{14.2}$$

where the $b_{k\ell}$'s are computed efficiently. We prove it by induction on $j$ with the base case of $j = 0$ being trivial. Now assume equation (14.2) holds then differentiating both sides of equation (14.2) with respect to $x_1$, we get an expression for $\partial_1^{j+1} f$. By linearity of derivatives it suffices to examine just one summand which is of the form $\partial_1 G^{D-k} x_1^{\ell}$. We have

$$\begin{aligned}\partial_1 G^{D-k} x_1^{\ell} &= \ell \cdot G^{D-k} x_1^{\ell-1} + (D-k) \cdot G^{D-k-1} x_1^{\ell} \cdot \partial_1 G \\ &= \ell \cdot G^{D-k} x_1^{\ell-1} + (D-k) \cdot G^{D-k-1} x_1^{\ell} \cdot \partial_1 (g_1(x_1)) \\ &= \ell \cdot G^{D-k} x_1^{\ell-1} + \sum_{i=0}^{d-1} a_i (D-k) \cdot G^{D-k-1} x_1^{\ell+i},\end{aligned}$$

where $\partial_1 g_1 = \sum_{i=0}^{d-1} a_i x_1^i$. This completes the proof of the lemma. □

We are now ready to prove our first result on POLYDEP and consequently on identity testing.

---

**Theorem 14.8.** For $i \in [m]$ and $j \in \{0, \ldots, D\}$, let $f_{ij}(\mathbf{X}) = G_i(\mathbf{X})^j$ where each $G_i$ is a sum of univariate polynomials of degree at most $d$, i.e., each $G_i$ is of the form

$$\big(g_{i1}(x_1) + \ldots + g_{in}(x_n)\big),$$

the $g_{ik}$'s being univariate polynomials of degree at most $d$. There is a deterministic algorithm for

$$\text{POLYDEP}\big(f_{ij} : i \in [m], j \in \{0, \ldots, D\}\big),$$

whose running time is bounded by poly$(nmdD)$.

---

*Proof.* The algorithm is as follows.

**Step 1**: If $n = 0$, then the $f_{ij}$'s are all field elements and thus, the computation of their dependencies is trivial.

**Step 2**: If $n \geq 1$, then by making a recursive call to

$$\text{POLYDEP}\big(\sigma_1(G_i)^j : i \in [m], j \in \{0, \ldots, D\}\big),$$

we get a basis for

$$\big(\sigma_1(G_i)^j : i \in [m], j \in \{0, \ldots, D\}\big)^{\perp}.$$

**Step 3**: Use the algorithm of Lemma 14.7 to compute $a_{ijks}$'s such that

$$\sigma_1 \partial_1^k G_i^j = \sum_{s=0}^{D} a_{ijks} \cdot \sigma_1(G_i)^s.$$

**Step 4**: From the data above and using the second property of Proposition 14.5, compute a basis for

$$V_k \overset{\text{def}}{=} \left\{ \sigma_1 \partial_1^k G_i^j : i \in [m], j \in \{0, \ldots, D\} \right\}^{\perp}.$$

**Step 5**: Output

$$\bigcap_{k=0}^{dD} V_k.$$

The correctness follows from Proposition 14.6. If the time taken is denoted by $T(n,m,d,D)$, then the recurrence relation is

$$T(n,m,d,D) = T(n-1,m,d,D) + \text{poly}(mdD),$$

which solves out to give $T(n,m,d,D) = \text{poly}(nmdD)$. □

# 15

## Absolute irreducibility testing

In chapter 8 we saw the role of absolute irreducibility in determining the number of zeroes of a bivariate polynomial over a finite field. Let us consider the following algorithmic question: given a bivariate polynomial $f(x, y)$ how do we determine whether or not it is absolutely irreducible? In this chapter, we will see how partial derivatives help in converting this apparently highly nonlinear problem into a simple task in linear algebra. This then gives an efficient algorithm for determining absolute irreducibility.

Recall that a bivariate polynomial $f(x, y) \in \mathbb{F}[x, y]$ is said to be *absolutely irreducible* if it is irreducible over the algebraic closure $\overline{\mathbb{F}}$ of $\mathbb{F}$. In Chapter 8 we saw an application of absolute irreducibility in determining the number of rational points on a curve over a finite field. A natural and interesting computational question is the following:

**Question 15.1.** Is $\mathcal{C}(\mathbf{X})$ absolutely irreducible?

In this chapter we will show how partial derivatives can be used to determine the absolutely irreducibility of a polynomial. We will then go on to Hilbert's Irreducibility Theorem. We start with bivariate polynomials and then generalize to multivariate polynomials. Theorem 15.2 below is due to Ruppert [Rup99]. Parts of the proof have been adapted from Gao [Gao03].

### Notation

For a polynomial $f(x, y) \in \mathbb{F}[x, y]$, we say that $\deg(f) = (m, n)$ if

$$\deg_x(f) = m \quad \text{and} \quad \deg_y(f) = n.$$

In this chapter, we will denote $\frac{\partial f}{\partial x}$ simply by $f_x$ and $\frac{\partial f}{\partial y}$ simply by $f_y$.

**Theorem 15.2.** Let $\mathbb{F}$ be a field of characteristic zero, then $f(x, y) \in \mathbb{F}[x, y]$ of degree $(m, n)$ is absolutely irreducible iff the equation

$$P_y \cdot f - f_y \cdot P = Q_x \cdot f - f_x \cdot Q \tag{15.1}$$

has no nonzero solution $P, Q \in \overline{\mathbb{F}}[x, y]$ with

$$\deg(P) \leq (m - 1, n) \quad \text{and} \quad \deg(Q) \leq (m, n - 2). \tag{15.2}$$

Before proceeding to the proof, first note the magic. The nonlinear task of testing if $f$ has nontrivial factors is reduced to the task of solving a system of homogeneous linear equations! Moreover, since the coefficients of this system are in $\mathbb{F}$, a solution $P, Q$ (if it exists) will also have coefficients in $\mathbb{F}$. These observations will essentially give the algorithm.

*Proof.* **The 'if part'.** We would like the reader to observe that *for every polynomial* $f$, the polynomials $P = f_x$ and $Q = f_y$ satisfy equation (15.1) so a solution to (15.1) with degree constraints $(m - 1, n)$ and $(m, n - 1)$ respectively always exists. More interestingly, we will see that if $f$ is reducible then we can lower the degree of a satisfying $Q$ from the trivial $(m, n - 1)$ to the strictly smaller value $(m, n - 2)$. Suppose that $f$ factors over the algebraic closure $\overline{\mathbb{F}}$. Let $f = g \cdot h$. We have already said that

$$P = f_x = gh_x + g_x h \quad \text{and} \quad Q = f_y = gh_y + g_y h$$

are solutions to (15.1). More interestingly, we observe that for any $\alpha, \beta \in \overline{\mathbb{F}}$,

$$\begin{aligned} P &= \alpha \cdot g_x \cdot h + \beta \cdot g \cdot h_x, \quad \text{and} \\ Q &= \alpha \cdot g_y \cdot h + \beta \cdot g \cdot h_y. \end{aligned}$$

is also a solution to (15.1). We leave it to the reader to verify this. If we now choose $\alpha = \deg_y(h)$ and $\beta = -\deg_y(g)$, then we further decrease the degree of $Q$ from $(m, n-1)$ to $(m, n-2)$ so that the degree constraints of equation (15.2) are also satisfied.

**The 'only if' part.** Assume that $f$ is absolutely irreducible. We will show that (15.1) and (15.2) has no solution. Let us now view $f(x, y)$ as a univariate polynomial in $x$ over the function field $\mathbb{F}(y)$. $\overline{\mathbb{F}(y)}$ is the algebraic closure of the function field $\mathbb{F}(y)$. Let $\mathbb{K} \subset \overline{\mathbb{F}(y)}$ be the splitting field of $f$ over $\overline{\mathbb{F}}(y)$. Then (dividing $f$ by its leading coefficient if necessary) we have

$$f = \prod_{i=1}^{m} (x - c_i),$$

where the $c_i$'s are distinct algebraic functions of $y$ in $\mathbb{K} \subset \overline{\mathbb{F}(y)}$ and are algebraic conjugates of each other over the base field $\overline{\mathbb{F}}(y)$. Suppose for contradiction that $P, Q$ satisfy equation (15.1) together with the degree constraints (15.2). The crux would be to prove that this can only happen if $Q = \alpha f_y$ for some constant $\alpha \in \mathbb{F}$ (from this a contradiction would follow simply). To show this, it would be useful to note that, up to a scaling factor $f^2$, equation 15.1 is equivalent to the equation

$$\frac{\partial}{\partial y}\left(\frac{P}{f}\right) = \frac{\partial}{\partial x}\left(\frac{Q}{f}\right)$$

so we proceed to explore these functions.

Using $\deg_x(P) < \deg_x(f)$ we have the partial fraction decompositions

$$\frac{P}{f} = \sum_{i=1}^{m} \frac{a_i}{x - c_i}, \quad \frac{Q}{f} = \sum_{i=1}^{m} \frac{b_i}{x - c_i} + q,$$

where

$$a_i = \frac{P(c_i, y)}{f_x(c_i, y)} \in \mathbb{K}, \quad b_i \in \mathbb{K} \text{ and } q \in \mathbb{F}(y) \subseteq \mathbb{K}. \tag{15.3}$$

The elements of $\mathbb{K}$ are algebraic functions of $y$ so that we have

$$\frac{\partial a}{\partial x} = 0 \quad \text{and} \quad \frac{\partial a}{\partial y} \in \mathbb{K} \quad \text{for} \quad \text{all } a \in \mathbb{K}.$$

We thus have:

$$\begin{aligned} \frac{\partial}{\partial y}\left(\frac{P}{f}\right) &= \sum_{i=1}^{m}\left(\frac{1}{x-c_i}\cdot\frac{\partial a_i}{\partial y} + \frac{a_i}{(x-c_i)^2}\cdot\frac{\partial c_i}{\partial y}\right), \\ \frac{\partial}{\partial x}\left(\frac{Q}{f}\right) &= \sum_{i=1}^{m}\frac{-b_i}{(x-c_i)^2} \end{aligned}$$

Now applying equation 15.1 in the form above, and using the uniqueness of partial fraction expansion, we have $\frac{\partial a_i}{\partial y} = 0$. This means that $a_i \in \overline{\mathbb{F}} \subset \overline{\mathbb{F}(y)}$. Further since the $c_i$'s are all algebraic conjugates over $\overline{\mathbb{F}}(y)$ so are the $a_i$'s which means that in fact $a_i = a_j$ as they are in $\overline{\mathbb{F}}$. Let $a_1 = a_2 = \ldots = a_m = \alpha$. Then by (15.3) we have that $P(x,y) = \alpha \cdot f_x$. But equation (15.1) then implies

$$f_x \cdot (\alpha f_y + Q) = f \cdot (Q_x + \alpha P_y)$$

By the absolute irreducibility of $f$, we have that $f$ divides either $f_x$ or $(\alpha f_y + Q)$. $f$ cannot divide $f_x$ because the latter has smaller degree by assumption. Therefore $f$ must divide $(\alpha f_y + Q)$. This polynomial also has smaller degree than that of $f$ so that the only way this is possible is if $(\alpha f_y + Q)$ is zero. This means that $Q = -\alpha f_y$ so that $\deg(Q) = (m, n-1)$, a contradiction. Thus there is no $P$ and $Q$ simultaneously satisfying both (15.1) and (15.2). □

### Consequences

Notice that solving for $P$ and $Q$ satisfying (15.1) and (15.2) involves solving a system of homogeneous linear equations. Now a system of linear equations has a nonzero solution in the algebraic closure $\overline{\mathbb{F}}$ if and only if it has one in $\mathbb{F}$ itself. We thus have

**Corollary 15.3.** There is a deterministic algorithm that given a polynomial $f(x,y) \in \mathbb{Q}[x,y]$ determines whether $f$ is absolutely irreducible or not and has running time $\text{poly}(\deg(f)\cdot H)$, where $H$ is the maximum height (namely bit-size) of the coefficients of $f$.

We now state without proof the following theorem which roughly says that we can set all but 2 variables of any absolutely irreducible $n$-variate polynomial to randomly chosen values so that with high probability the resulting bivariate polynomial will also be absolutely irreducible.

**Theorem 15.4. (Hilbert's Irreducibility Theorem)**
Let $f \in \mathbb{F}[x, y_1, y_2, \ldots, y_n]$ be an absolutely irreducible $(n+1)$-variate polynomial. If $f_x \neq 0$ then for "most" choices of $\mathbf{a}, \mathbf{b} \in \mathbb{F}^n$, the following bivariate polynomial

$$f(x, a_1 t + b_1, a_2 t + b_2, \ldots, a_n t + b_n)$$

is also absolutely irreducible.

See for example Kaltofen [Kal89] for a proof. Hilbert's Irreducibility Theorem now immediately provides us with an algorithm to test absolute irreducibility in randomized polynomial time.

**Corollary 15.5.** Let $\mathbb{F}$ be a field of characteristic zero. Given a polynomial $f(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$, we can determine the absolute irreducibility of $f$ in randomized time $\text{poly}(d \cdot s)$ where $d$ is the degree of $f$ and $s$ is the size of the arithmetic circuit specifying $f$.

*Proof.* Apply a random invertible transformation of the variables — this ensures that if $f$ is a nonconstant polynomial then with high probability, $\frac{\partial f}{\partial x}$ is nonzero. Let

$$g(x, t) \stackrel{\text{def}}{=} f(x, a_1 t + b_1, a_2 t + b_2, \ldots, a_n t + b_n)$$

where the $a_i$'s and the $b_i$'s are chosen uniformly at random from an appropriately large set. Use the algorithm of Corollary 15.3 to test absolute irreducibility of $g$ and accept if and only if $g$ is irreducible. □

Hilbert's Irreducibility Theorem can be used to do much more — it is the key ingredient in the efficient randomized algorithm of Kaltofen [Kal89] for factoring multivariate polynomials given as arithmetic circuits. We shall use Kaltofen's algorithm in the subsequent section to devise more algorithms.

# 16

## Polynomial equivalence testing

An $n$-variate polynomial $f$ over a field $\mathbb{F}$ represents a function from the vector space $\mathbb{F}^n$ to $\mathbb{F}$ in the natural way:

$$(x_1, x_2, \ldots, x_n) \mapsto f(x_1, x_2, \ldots, x_n).$$

A basic question then is: given two polynomials, how do we determine if they are the same function *upto a change of basis*. In other words, given $n$-variate polynomials $f$ and $g$, does there exist an invertible matrix $A = (a_{ij})_{1 \le i,j \le n}$ such that

$$g(x_1, \ldots, x_n) = f(a_{11}x_1 + \ldots + a_{1n}x_n, \ldots, a_{n1}x_1 + \ldots + a_{nn}x_n)?$$

While this appears to be a difficult question in general, we will show how partial derivatives help solve some rather special cases.

Two polynomials $f(\mathbf{X})$ and $g(\mathbf{X})$ are said to be *equivalent* iff there is an invertible linear transformation of the variables which maps $f$ to $g$, i.e., if there exists an $A \in GL_n(\mathbb{F})$ such that

$$f(A \cdot \mathbf{X}) = g(\mathbf{X}).$$

Another way to look at this equivalence relation among polynomials is the following. An $n$-variate polynomial can be viewed as a function from the vector space $\mathbb{F}^n$ to $\mathbb{F}$. Two polynomials are equivalent if they are the same function *upto a change of basis* of the vector space $\mathbb{F}^n$. If two polynomials are equivalent then they are structurally or geometrically the same. In particular they have the same complexity in most natural models of computation. This chapter addresses the following problem: given two polynomials can we efficiently determine if they are equivalent?

**Historical Notes.** A topic of intense mathematical interest and attention during the nineteenth century was the study of the intrinsic or geometrical properties of polynomials. The task here was to find and describe all fundamental invariants of a polynomial, i.e. find a list of geometrical properties of a polynomial which describes the polynomial completely upto equivalence. The dramatic and unexpected solution to its most fundamental problem — the finitude of the number of fundamental invariants — propelled the young David Hilbert into the position of the most renowned mathematician of his time. Following subsequent decline this topic sank into obscurity during the twentieth century. Ironically, though, its indirect influence continued to be felt on many parts of abstract algebra, while in computational algebra/algebraic geometry the three most famous of Hilbert's theorems — the Basis Theorem, the Syzygy Theorem and the Nullstellensatz — were all born as lemmas (*Hilfsätze*) for proving "more important" results in invariant theory! For this fascinating account and more details of the topic, we refer the reader to the text by Peter Olver [Olv99].

**A Modern Perspective.** An algorithmic consequence of Hilbert's work is that testing whether two given polynomials are equivalent (the polynomial equivalence problem) is decidable — but we do not know if it is in $\mathsf{P}$ (Hilbert's work actually goes much deeper than this). In its full generality, polynomial equivalence appears to be a difficult question — it is at least as hard as graph isomorphism (see the footnote to Exercise 2.9) and probably much harder. Graph isomorphism admits heuristic algorithms which seem to work very well in practice but no such heuristic algorithms are known for polynomial equivalence. In this chapter we will use partial derivatives to solve certain special cases. These algorithms were first presented in [Kay10]. The algorithms show how the space of partial derivatives of up to second order of a given polynomial relates to its structure. Partial derivatives also play a crucial role in classical invariant theory and we refer the reader to the text by Hilbert himself [Hil93] and to the text by Olver [Olv99] for more on this topic.

Suppose that we are given a polynomial as an arithmetic circuit $\mathcal{C}(\mathbf{X})$. We will address the following questions.

**Question 16.1.** Is $\mathcal{C}(\mathbf{X})$ equivalent to a polynomial that depends on a fewer number of variables?

**Question 16.2.** Is $\mathcal{C}(\mathbf{X})$ equivalent to $x_1^d + \ldots + x_n^d$?

**Question 16.3.** Is $\mathcal{C}(\mathbf{X})$ equivalent to an elementary symmetric polynomial?

Notice that for each of these questions, if the phrase "equivalent to" is replaced by "equal to" then the questions are easily seen to reduce to the identity testing problem. We will see how partial derivatives are useful in devising efficient randomized algorithms for the simpler versions of polynomial equivalence that are captured in questions 16.1, 16.2 and 16.3. We refer the interested readers to [Kay10] for certain generalizations of these algorithms.

### Notation

We specify some more notation to be used for the rest of this chapter. Let $f(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$ be a polynomial.

- $\partial^k f$ shall denote the set of $k$-th order partial derivatives of $f$. Thus $\partial^1 f$, abbreviated as $\partial f$, shall equal

$$\left\{\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right\}, \quad \text{and}$$

$\partial^2 f$ is the set

$$\left\{\frac{\partial^2 f}{\partial x_i \cdot \partial x_j} : 1 \leq i \leq j \leq n\right\}.$$

- Also recall that we have

$$\partial_i f \overset{\text{def}}{=} \frac{\partial f}{\partial x_i} \quad \text{and} \quad \sigma_i f \overset{\text{def}}{=} f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n).$$

**Ring of differential operators.** Notice that both differentiation and substitution are $\mathbb{F}$-linear maps from $\mathbb{F}[\mathbf{X}]$ to itself. We will denote the linear combinations and compositions of these basic linear operators in the natural way. Thus $\partial_i \partial_j$ is a shorthand for the map that sends $f(\mathbf{X})$ to $(\partial_i(\partial_j f))(\mathbf{X})$, while $\partial_i + \partial_j$ is a shorthand for the map that sends $f(\mathbf{X})$ to $(\partial_i f + \partial_j f)(\mathbf{X})$. Continuing in this way, one can look at all polynomial expressions in $\partial_1, \ldots, \partial_n$. They form a commutative ring which we denote by $\mathbb{F}[\partial_1, \ldots, \partial_n]$. We call it the ring of differential operators.

## 16.1 Algorithm for minimizing the number of variables

We will now see how to eliminate redundant variables from a polynomial. We adopt the following terminology from Carlini [Car06]:

**Definition 16.4.** Let $f(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ be a polynomial. We say $f(\mathbf{X})$ is independent of a variable $x_i$ if no monomial of $f(\mathbf{X})$ contains $x_i$. We say *the number of essential variables* in $f(\mathbf{X})$ is $t$ if $t \geq 0$ is the smallest integer, such that, we can make an invertible linear $A \in GL_n(\mathbb{F})$ transformation on the variables $\mathbf{X}$ and $f(A \cdot \mathbf{X})$ depends only on $t$ variables $x_1, \ldots, x_t$. The remaining $(n-t)$ variables $x_{t+1}, \ldots, x_n$ are said to be *redundant* variables. We will say that $f(\mathbf{X})$ is a *regular* polynomial if it has no redundant variables.

**Example 16.5.** The number of essential variables in the quadratic polynomial $f(x_1, x_2, x_3) = x_1^2 + 2x_1x_2 + x_2^2 + x_3^2$ is just two, because notice that $f = (x_1 + x_2)^2 + x_3^2$ and thus, after making the invertible linear transformation

$$A \quad : \quad \begin{array}{l} x_1 + x_2 \mapsto x_1 \\ x_3 \mapsto x_2 \\ x_2 \mapsto x_3 \end{array}$$

we get that $f(A \cdot \mathbf{X}) = x_1^2 + x_2^2$ is a polynomial that only depends on two variables $x_1$ and $x_2$.

### The vanishing of partials

Let us examine the situation when the number of essential variables in $f(\mathbf{X})$ is $t < n$ and relate this to the partial derivatives of $f$. We state without proof the following lemma from Carlini [Car06].

**Lemma 16.6.** The number of redundant variables in a polynomial $f(\mathbf{X})$ equals the dimension of $\partial(f)^\perp$.

Furthermore, given a basis of $\partial(f)^\perp$, one can construct efficiently a linear transformation $A$ over the variables such that the polynomial $f(A \cdot \mathbf{X})$ only depends on the first $(n - \dim(\partial(f)^\perp))$ variables.

Lemma 16.6 combined with the randomized algorithm for POLYDEP given by lemma 14.4 and the fact that arithmetic circuits for the first order partial derivatives of $f$ can be easily computed from the circuit of $f$ (Exercise 1.10) imply:

**Theorem 16.7.** Given a polynomial $f(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ with $m$ essential variables, we can compute in randomized polynomial time an invertible linear transformation $A \in GL_n(\mathbb{F})$ such that $f(A \cdot \mathbf{X})$ depends on the first $m$ variables only.

Henceforth we will assume that the given polynomial is "*regular*", that is, no invertible linear transformation of the variables can reduce the number of variables.

## 16.2 Equivalence to a sum of powers

Consider the following problem: given a circuit computing a homogeneous polynomial $f(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ of degree $d$, does there exist a linear transformation $A \in GL_n(\mathbb{F})$ and constants $a_1, \ldots, a_n \in \mathbb{F}$ such that

$$f(A \cdot \mathbf{X}) = a_1 \cdot x_1^d + a_2 \cdot x_2^d + \ldots + a_n \cdot x_n^d.$$

Equivalently, the problem can be restated as follows: given a circuit computing a homogeneous polynomial $f(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ of degree $d$, determine $n$ independent linear forms $\ell_1, \ldots, \ell_n \in \mathbb{F}[\mathbf{X}]$ and $n$ constants $a_1, \ldots, a_n \in \mathbb{F}$ such that

$$f(\mathbf{X}) = a_1 \cdot \ell_1(\mathbf{X})^d + \ldots + a_n \cdot \ell_n(\mathbf{X})^d.$$

We will devise a randomized polynomial-time algorithm that given $f$, computes the constants and the set of linear forms $\ell_1, \ldots, \ell_n$, if they exist. The idea involved is a small variation of the idea used in characterizing the symmetries of power-symmetric polynomials that we presented in Chapter 2. Recall the definition of the Hessian matrix and its following property:

**Lemma 16.8.** Let $f(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ be an $n$-variate polynomial and $A \in \mathbb{F}^{n \times n}$ be a linear transformation. Let

$$F(\mathbf{X}) \stackrel{\text{def}}{=} f(A \cdot \mathbf{X}).$$

Then we have

$$H_F(\mathbf{X}) = A^T \cdot H_f(A \cdot \mathbf{X}) \cdot A.$$

In particular,

$$\textsc{Det}\big(H_F(\mathbf{X})\big) = \textsc{Det}(A)^2 \cdot \textsc{Det}\big(H_f(A \cdot \mathbf{X})\big).$$

Now consider a homogeneous polynomial $f(\mathbf{X})$ of degree $d \geq 3$, which has the property that there exists a linear transformation $A \in GL_n(\mathbb{F})$ of the variables such that

$$f(A \cdot \mathbf{X}) = x_1^d + x_2^d + \ldots + x_n^d.$$

Set $F(\mathbf{X}) \stackrel{\text{def}}{=} x_1^d + x_2^d + \ldots + x_n^d$. Observe that

$$\frac{\partial^2 F}{\partial x_i \cdot \partial x_j} = \begin{cases} 0 & \text{if } i \neq j, \\ d(d-1)x_i^{d-2} & \text{if } i = j. \end{cases}$$

Thus the matrix $H_F(\mathbf{X})$ is a diagonal matrix so that we have

$$\textsc{Det}\big(H_F(\mathbf{X})\big) = d^n (d-1)^n \cdot \prod_{i=1}^{n} x_i^{d-2}.$$

By Lemma 2.6, we get that

$$\textsc{Det}\big(H_f(A \cdot \mathbf{X})\big) = d^n (d-1)^n \cdot \textsc{Det}(A)^{-2} \cdot \prod_{i=1}^{n} x_i^{d-2}$$

and thus,

$$\textsc{Det}\big(H_f(\mathbf{X})\big) = d^n (d-1)^n \cdot \textsc{Det}(A)^{-2} \cdot \prod_{i=1}^{n} \ell_i(\mathbf{X})^{d-2},$$

where the $\ell_i(\mathbf{X})$'s are linear forms corresponding to the different rows of the matrix $A^{-1}$. Let us record this as a lemma.

**Lemma 16.9.** For a polynomial $f(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ of degree $d$, if

$$f(\mathbf{X}) = \sum_{i=1}^{n} a_i \cdot \ell_i(\mathbf{X})^d,$$

where $\ell_1(\mathbf{X}), \ldots, \ell_n(\mathbf{X})$ are independent linear forms, then

$$\textsc{Det}(H_f(\mathbf{X})) = c \cdot \prod_{i=1}^{n} \ell_i(\mathbf{X})^{d-2},$$

where $c \in \mathbb{F}$ is a nonzero constant.

Lemma 16.9 can now be used to devise a randomized polynomial-time algorithm for sum-of-powers equivalence testing as follows.

**Input.** An arithmetic circuit which computes an $n$-variate polynomial $f(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ of degree $d$.

**Output.** A set of independent linear forms $\ell_1(\mathbf{X}), \ldots, \ell_n(\mathbf{X})$ and constants $a_1, \ldots, a_n$ such that

$$f(\mathbf{X}) = a_1 \cdot \ell_1(\mathbf{X})^d + \ldots + a_n \cdot \ell_n(\mathbf{X})^d,$$

if such a set of $\ell_i$'s exists.

**The Algorithm.**

(a) Compute a new arithmetic circuit $\mathcal{C}'(\mathbf{X})$ from $\mathcal{C}(\mathbf{X})$, which computes $\textsc{Det}(H_f(\mathbf{X}))$.

(b) Use Kaltofen's factorization algorithm [Kal89] to factor $\mathcal{C}'(\mathbf{X})$ in randomized polynomial time. If it is not the case that

$$\mathcal{C}'(\mathbf{X}) = \prod_{i=1}^{n} \ell_i(\mathbf{X})^{d-2},$$

where each $\ell_i(\mathbf{X})$ is a linear form, then output No such forms. [1] Else (by solving a system of linear equations) compute constants $a_1, \ldots, a_n$ such that

$$f(\mathbf{X}) = \sum_{i=1}^{n} a_i \cdot \ell_i(\mathbf{X})^d.$$

Output $\big(\ell_1(\mathbf{X}), \ldots, \ell_n(\mathbf{X})\big)$ and $(a_1, \ldots, a_n)$.

This completes the description of the algorithm to determine whether a given polynomial is a sum of powers.

## 16.3 Equivalence to an elementary symmetric polynomial

The problem that we now tackle is the following — given an arithmetic circuit which computes an $n$-variate homogeneous polynomial $f(\mathbf{X}) \in$

[1] Kaltofen's algorithm as usually stated outputs circuits for each factor. If the factors are constant degree, as in our case, then we can also recover each factor explicitly in the usual sum of monomials representation by doing interpolation.

$\mathbb{F}[\mathbf{X}]$, is there an invertible linear transformation $A$ such that $f(A \cdot \mathbf{X})$ is the elementary symmetric polynomial of degree $d$? Recall that the elementary symmetric polynomial of degree $d$ [2] is

$$\mathrm{SYM}_n^d \overset{\mathrm{def}}{=} \sum_{S \subseteq [n], |S|=d} \prod_{i \in S} x_i.$$

Observe that $\mathrm{SYM}_n^d$ is a multilinear polynomial and therefore we have

$$\partial_i^2 \left(\mathrm{SYM}_n^d\right) = 0, \qquad \text{for all } i \in [n]. \tag{16.1}$$

More interestingly, these are essentially the only second-order partial derivatives of $\mathrm{SYM}_n^d$ which vanish. The following lemma follows from the proof of Lemma 11.4.

**Lemma 16.10.** For $d \geq 4$, we have $\dim \left(\partial^2(SYM_n^d)\right) = \binom{n}{2}$.

This means that if $f$ is equivalent to $\mathrm{SYM}_n^d$, then $\partial^2(f)$ has dimension $\binom{n}{2}$. Indeed our method shows that for any polynomial $f \in \mathbb{F}(\mathbf{X})$ which has the property that $\partial^2(f)$ has dimension $\binom{n}{2}$, we can efficiently determine whether $f$ is equivalent to a multilinear polynomial and if so, find an invertible matrix $A$ such that $f(A \cdot \mathbf{X})$ is multilinear. Now let

$$g(\mathbf{X}) \overset{\mathrm{def}}{=} f(A \cdot \mathbf{X})$$

be multilinear. It will also follow from our proof that this multilinear polynomial $g(\mathbf{X})$ is equivalent to an elementary symmetric polynomial if and only if there is a diagonal matrix $B$ such that

$$g(B \cdot \mathbf{X}) = \mathrm{SYM}_n^d.$$

It is then a relatively easy exercise to determine whether such a diagonal matrix $B$ exists or not.

[2] $\mathrm{SYM}_n^d$ also admits an alternative definition similar in spirit to the characterizations provided by exercises 2.2 and 2.3. $\mathrm{SYM}_n^d$ is the unique (upto scalar multiples) homogeneous *multilinear* polynomial of degree $d$ in $n$ variables, which is invariant under every permutation of the variables.

**Exercise 16.11.** Show that given a multilinear polynomial $g(\mathbf{X})$, one can efficiently determine whether there exist $\lambda_1, \ldots, \lambda_n \in \mathbb{F}$ such that

$$g(\lambda_1 x_1, \ldots, \lambda_n x_n) = \mathrm{SYM}_n^d(\mathbf{X})$$

In the rest of this section, we will assume that $f(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ is a polynomial that satisfies $\dim(\partial^2(f)) = \binom{n}{2}$. We will tackle the problem of finding an invertible matrix $A$ such that $f(A \cdot \mathbf{X})$ is multilinear, if such an $A$ exists. We will first observe that our problem boils down to finding a *good* basis for a given space of symmetric matrices consisting of *rank one matrices*. We then devise an efficient randomized algorithm for the latter problem.

### 16.3.1 Reduction to finding a good basis for a space of matrices

First we make the following observation. Let

$$g(\mathbf{X}) = f(A \cdot \mathbf{X}) = f\left(\sum_j a_{1j} x_j, \sum_j a_{2j} x_j, \ldots, \sum_j a_{nj} x_j\right).$$

Then $\partial_i^2 g = 0$ if and only if

$$\big(a_{1i}\partial_1 + a_{2i}\partial_2 + \ldots + a_{ni}\partial_n\big)^2 f = 0.$$

Therefore, if $g(\mathbf{X})$ is multilinear then every column vector of $A$ satisfies

$$\big(a_1\partial_1 + a_2\partial_2 + \ldots + a_n\partial_n\big)^2 f = 0,$$

and these $n$ vectors are linearly independent since $A$ is invertible.

Now given $f$, we first compute the set $\partial^2 f$ and then using the randomized algorithm for POLYDEP, we obtain a basis for the set of all quadratic differential operators $D(\partial_1, \ldots, \partial_n)$ such that $Df = 0$. Since $\dim(\partial^2(f)) = \binom{n}{2}$ we have $\dim(D(\partial_1, \ldots, \partial_n)) = n$. By the observation above our problem boils down to finding a basis for $D(\partial_1, \ldots, \partial_n)$ such that every quadratic operator in the basis has the following form:

$$(a_1\partial_1 + a_2\partial_2 + \ldots + a_n\partial_n)^2 f = 0.$$

Towards this end, we associate every $n$-variate quadratic operator $D$ with an $n \times n$ symmetric matrix $\hat{D}$ in the following natural way. Let $D \in \mathbb{F}[\partial_1, \ldots, \partial_n]$ be a quadratic polynomial, where

$$D = \sum_{i \in [n]} \alpha_i \partial_i^2 + \sum_{1 \le i < j \le n} \beta_{ij} \partial_i \partial_j.$$

The matrix $\hat{D}$ associated with this operator $D$ is the following:

$$\hat{D} \overset{\text{def}}{=} \begin{pmatrix} \alpha_1 & \frac{1}{2}\beta_{12} & \ldots & \frac{1}{2}\beta_{1n} \\ \frac{1}{2}\beta_{12} & \alpha_2 & \ldots & \frac{1}{2}\beta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2}\beta_{1n} & \frac{1}{2}\beta_{2n} & \ldots & \alpha_n \end{pmatrix}. \tag{16.2}$$

This way of associating a quadratic polynomial with a symmetric matrix has the following property.

**Property 16.12.** Over an algebraically closed field $\mathbb{F}$ of characteristic different from 2, the quadratic polynomial $D$ is equivalent to a sum of $r$ squares if and only if the corresponding symmetric matrix $\hat{D}$ is of rank $r$. In particular, the polynomial $D$ is a perfect square if and only if $\hat{D}$ is of rank one.

Using this property, our problem is equivalent to finding a basis of a given space of symmetric matrices consisting of rank one symmetric matrices in the following way.

(1) Given an arithmetic circuit of size $s$ for the polynomial $f(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$, we use the naive method of computing derivatives to obtain a new circuit of size $O(sn^2)$, whose outputs are the second-order partial derivatives $\partial^2(f)$ of $f$.

(2) Using the randomized algorithm for POLYDEP, we obtain a basis for $(\partial^2(f))^\perp$. Each element in the basis of $(\partial^2(f))^\perp$ is a homogeneous quadratic polynomial in $\mathbb{F}[\partial_1, \ldots, \partial_n]$ in the natural way. Let this basis be

$$\{D_1, \ldots, D_n\} \subset \mathbb{F}[\partial_1, \ldots, \partial_n].$$

(3) From $D_1, \ldots, D_n$, we get the corresponding symmetric matrices $\hat{D}_1, \ldots, \hat{D}_n$. Using the randomized algorithm given below, we obtain another basis $\{\hat{E}_1, \ldots, \hat{E}_n\}$ of the vector space generated by $\{\hat{D}_1, \ldots, \hat{D}_n\}$ such that each $\hat{E}_i$ is a rank one symmetric matrix [3], if such a basis exists. Their corresponding quadratic polynomials $E_1, \ldots, E_n \subset \mathbb{F}[\partial_1, \ldots, \partial_n]$ are then perfect squares. Let

$$E_i = \left( \sum_{j \in [n]} a_{ij} \partial_j \right)^2 .$$

The matrix $A = (a_{ij})_{i,j \in [n]}$ is then the required linear transformation which makes $f$ multilinear.

We now present an efficient randomized algorithm that given $n$ linearly independent matrices of dimension $n \times n$, finds a basis consisting of rank-one matrices, if such a basis exists. Our proof will also show that such a basis, if it exists, is unique up to scalar multiples and permutations of the basis elements.

### 16.3.2 Randomized algorithm for finding a basis consisting of rank-one matrices

We are given $n$ symmetric matrices $\hat{D}_1, \ldots, \hat{D}_n$, and we want to find another basis $\hat{E}_1, \ldots, \hat{E}_n$ of the space generated by the given matrices such that each $\hat{E}_i$ is of rank one. A rank one symmetric matrix is the outer product of a vector with itself. So for each $i \in [n]$, let $\hat{E}_i = \mathbf{v}_i^T \mathbf{v}_i$ where $\mathbf{v}_i \in \mathbb{F}^n$.

**Lemma 16.13.** Suppose that $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{F}^n$ are vectors. Then

$$\textsc{Det}\big(z_1 \mathbf{v}_1^T \cdot \mathbf{v}_1 + \ldots + z_n \mathbf{v}_n^T \cdot \mathbf{v}_n\big) = z_1 z_2 \ldots z_n \cdot \big(\textsc{Det}(V)\big)^2, \quad (16.3)$$

where $V = [\mathbf{v}_1^T \ldots \mathbf{v}_n^T]$ is the matrix whose columns are the $\mathbf{v}_i$'s.

[3] Here we think of matrices as $n^2$-dimensional vectors.

*Proof.* Let $M(\mathbf{z}) \overset{\text{def}}{=} z_1\mathbf{v}_1^T \cdot \mathbf{v}_1 + \ldots + z_n\mathbf{v}_n^T \cdot \mathbf{v}_n$. Then $\textsc{Det}(M(\mathbf{z}))$ is a polynomial of degree $n$ in the formal variables $z_1, \ldots, z_n$. If $z_i = 0$ then for every setting of the remaining variables, the matrix $M$ is singular because its image is spanned by the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{i-1}, \mathbf{v}_{i+1}, \ldots, \mathbf{v}_n$, and is of rank at most $n-1$. Thus $z_i$ divides $\textsc{Det}(M(\mathbf{z}))$ for all $i \in [n]$. Using Chinese remaindering, we have that $\prod z_i$ divides $\textsc{Det}(M(\mathbf{z}))$. Because the degree of $\textsc{Det}(M(\mathbf{z}))$ is $n$, we have

$$\textsc{Det}\big(M(\mathbf{z})\big) = \lambda \prod_{i\in[n]} z_i,$$

for some scalar $\lambda \in \mathbb{F}$. Setting all the $z_i$'s to 1, we get

$$\lambda = \textsc{Det}\left(\sum_{i\in[n]} \mathbf{v}_i^T \cdot \mathbf{v}_i\right) = \textsc{Det}(V \cdot V^T) = \textsc{Det}(V)^2.$$

We thus have $\textsc{Det}(M(\mathbf{z})) = z_1z_2\ldots z_n \cdot (\textsc{Det}(V))^2$. □

---

**Corollary 16.14.** Let $\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_n \in \mathbb{F}^{n\times n}$ be symmetric matrices. Suppose that there exist vectors $\mathbf{v}_1, \ldots \mathbf{v}_n$ such that

$$\hat{D}_i = \sum_{j=1}^{n} \alpha_{ij}\mathbf{v}_j^T \cdot \mathbf{v}_j. \tag{16.4}$$

Then

$$\textsc{Det}(z_1\hat{D}_1 + \ldots z_n\hat{D}_n) = \text{constant} \cdot \ell_1\ell_2\ldots\ell_n,$$

where for all $j \in [n]$, $\ell_j = \sum_{i=1}^{n} \alpha_{ij}z_i$ is a linear form over $z_1, \ldots, z_n$.

---

Corollary 16.14 suggests an algorithm.

---

**Lemma 16.15.** There exists a randomized polynomial-time algorithm that, given $n$ symmetric matrices $\hat{D}_1, \ldots, \hat{D}_n \in \mathbb{F}^{n\times n}$, finds a basis for the space generated by them consisting of matrices of rank one, if such a basis exists.

---

*Proof.* We write down an arithmetic circuit for the polynomial

$$F(z_1, \ldots, z_n) \overset{\text{def}}{=} \textsc{Det}(z_1\hat{D}_1 + \ldots + z_n\hat{D}_n).$$

Then we use Kaltofen's algorithm [Kal89] to factor $F(z_1, z_2, \ldots, z_n)$ in randomized polynomial time. By Corollary 16.14 we can use the linear factors $\ell_1, \ell_2, \ldots, \ell_n$ of this polynomial, which are unique up to scalar multiples and permutations, to solve the equations (16.4), and get the rank one matrices as required. □

This completes the description of the algorithm.

# Acknowledgements

We would like to thank an anonymous referee for many suggestions. Our thanks also go to Jiri Matousek, Amir Shpilka, Madhu Sudan and Amir Yehudayoff for taking the time to read an earlier draft and for their comments and suggestions.

# References

[AB03] M. Agrawal and S. Biswas. Primality and identity testing via chinese remaindering. *Journal of the ACM*, 50(4), 2003.

[Agr05] M. Agrawal. Proving lower bounds via pseudo-random generators. In *Proceedings of the 25th Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 92–105, 2005.

[AS06] M. Agrawal and N. Saxena. Equivalence of F-algebras and cubic forms. In *Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science*, pages 115–126, 2006.

[AW09] S. Aaronson and A. Wigderson. Algebrization: A new barrier in complexity theory. *ACM Transactions on Computation Theory*, 1(1):1–54, 2009.

[BCS97] P. Bügisser, M. Clausen, and A. Shokrollahi. *Algebraic Complexity Theory*. Springer, 1997.

[Ber84] S. J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Information Processing Letters*, 18:147–150, 1984.

[BGS75] T. Baker, J. Gill, and R. Solovay. Relativizations of the P =? NP question. *SIAM Journal on Computing*, 4(4):431–442, 1975.

[BHLV09] M. Bläser, M. Hardt, R.J. Lipton, and N.K. Vishnoi. Deterministically testing sparse polynomial identities of unbounded degree. *Information Processing Letters*, 109(3):187–192, 2009.

[BLMW09] P. Bürgisser, J.M. Landsberg, L. Manivel, and J. Weyman. An overview of mathematical issues arising in the geometric complexity theory approach to VP v.s. VNP. *CoRR*, abs/0907.2850, 2009.

[BM75] A. Borodin and I. Munro. *The Computational Complexity of Algebraic and Numeric Problems*. American Elsevier, first edition, 1975.

[BS83] W. Baur and V. Strassen. The complexity of partial derivatives. *Theoretical Computer Science*, 22(3):317–330, 1983.

[Cai90] J.-Y. Cai. A note on the determinant and permanent problem. *Information and Computation*, 84(1):119–127, 1990.

[Car06] E. Carlini. Reducing the number of variables of a polynomial. In *Algebraic Geometry and Geometric Modelling*, pages 237–247. Springer, 2006.

[CCL08] J.-Y. Cai, X. Chen, and D. Li. A quadratic lower bound for the permanent and determinant problem over any characteristic $\neq 2$. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 491–498, 2008.

[Chi85] A.L. Chistov. Fast parallel calculation of the rank of matrices over a field of arbitary characteristic. In *Proceedings of the International Conference Foundations of Computation Theory*, pages 63–69, 1985.

[CK00] Z.-Z. Chen and M.-Y. Kao. Reducing randomness via irrational numbers. *SIAM Journal of Computing*, 29(4):1247–1256, 2000.

[Csa76] L. Csanky. Fast parallel inversion algorithm. *SIAM Journal on Computing*, 5:618–623, 1976.

[CW90] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251–280, 1990.

[DGW09] Z. Dvir, A. Gabizon, and A. Wigderson. Extractors and rank extractors for polynomial sources. *Computational Complexity*, 18(1):1–58, 2009.

[DS05] Z. Dvir and A. Shpilka. Locally decodable codes with 2 queries and polynomial identity testing for depth 3 circuits. In *Proceedings of the 37th annual ACM symposium on Theory of computing*, pages 592–601, 2005.

[Dvi09a] Z. Dvir. From randomness extraction to rotating needles. *SIGACT News*, 40(4):46–61, 2009.

[Dvi09b] Z. Dvir. On the size of Kakeya sets in finite fields. *Journal of the American Mathematical Society*, 22:1093–1097, 2009.

[Ell69] W. J. Ellison. A 'waring's problem' for homogeneous forms. *Proceedings of the Cambridge Philosophical Society*, 65:663–672, 1969.

[ER93] R. Ehrenborg and G.-C. Rota. Apolarity and canonical forms for homogeneous polynomials. *European Journal of Combinatorics*, 14(3):157–181, 1993.

[Fis94] I. Fischer. Sums of like powers of multivariate linear forms. *Mathematics Magazine*, 67(1):59–61, 1994.

[Gao03] S. Gao. Factoring multivariate polynomials via partial differential equations. *Mathematics of computation*, 72(242):801–822, 2003.

[GK08] L. Guth and N.H. Katz. Algebraic methods in discrete analogs of the Kakeya problem. arXiv:0812.1043, 2008.

[GS99] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45(6):1757–1767, 1999.

[Har77] R. Hartshorne. *Algebraic Geometry.* Graduate Texts in Mathematics, No. 52. Springer-Verlag, New York, 1977.

[HB96] D.R. Heath-Brown. An estimate for Heilbronn's exponential sum. In *Analytic number theory: Proceedings of a conference in honor of Heini Halberstam*, pages 451–463, 1996.

[Hil93] David Hilbert. *Theory of Algebraic Invariants.* Cambridge University Press, 1993.

[HY09] P. Hrubes and A. Yehudayoff. Arithmetic complexity in algebraic extensions. *Manuscript*, 2009.

[Kal89] E. Kaltofen. Factorization of polynomials given by straight-line programs. *Randomness and Computation*, 5:375–412, 1989.

[Kay09] N. Kayal. The complexity of the annihilating polynomial. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity*, pages 184–193, 2009.

[Kay10] N. Kayal. Algorithms for arithmetic circuits. Technical report, Electronic Colloquium on Computational Complexity, 2010.

[Kel39] O. Keller. Ganze cremona-transformationen. *Monatsh. Math. Phys.*, 47:299–306, 1939.

[KI04] V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1/2):1–46, 2004.

[KN97] E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press Cambridge, 1997.

[KS01] A. Klivans and D.A. Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the 33rd annual ACM symposium on Theory of computing*, pages 216–223, 2001.

[KS06] N. Kayal and N. Saxena. Polynomial identity testing for depth 3 circuits. In *Proceedings of the twenty-first Annual IEEE Conference on Computational Complexity*, 2006.

[KS07] Z.S. Karnin and A. Shpilka. Black box polynomial identity testing of depth-3 arithmetic circuits with bounded top fan-in. *Electronic Colloquium on Computational Complexity*, 14(042), 2007.

[KS09] N. Kayal and S. Saraf. Blackbox polynomial identity testing for depth 3 circuits. *Electronic Colloquium on Computational Complexity*, 16(032), 2009.

[KSS10] H. Kaplan, M. Sharir, and E. Shustin. On lines and joints. *Discrete and Computational Geometry*, to appear, 2010.

[KSY11] S. Kopparty, S. Saraf, and S. Yekhanin. High-rate codes with sublinear-time decoding. In *Proceedings of ACM Symposium on Theory of Computing*, 2011.

[Kud01] L.D. Kudryavtsev. Implicit function. *Encyclopaedia of Mathematics*, 2001.

[Lov11] Shachar Lovett. Computing polynomials with few multiplications. Technical Report 094, Electronic Colloquium on Computational Complexity, 2011.

[LV98] D. Lewin and S.P. Vadhan. Checking polynomial identities over any field: Towards a derandomization? In *Proceedings of the 30th annual ACM Symposium on Theory of Computing*, pages 438–447, 1998.

[Mes89] R. Meshulam. On two extremal matrix problems. *Linear Algebra and its Applications*, 114/115:261–271, 1989.

[Mit92] D.A. Mit'kin. Stepanov method of the estimation of the number of roots of some equations. *Matematicheskie Zametki*, 51(6):52–58, 1992. Translated as Mathematical Notes, 51, 565–570, 1992.

[Mor85] J. Morgenstern. How to compute fast a function and all its derivatives: a variation on the theorem of Baur-Strassen. *SIGACT News*, 16(4):60–62, 1985.

[MR04] T. Mignon and N. Ressayre. A quadratic bound for the determinant and permanent problem. *International Mathematics Research Notices*, pages 4241–4253, 2004.

[MS01] K. Mulmuley and M.A. Sohoni. Geometric complexity theory I: An approach to the P vs. NP and related problems. *SIAM Journal on Computing*, 31(2):496–526, 2001.

[MS08] K. Mulmuley and M.A. Sohoni. Geometric complexity theory II: Towards explicit obstructions for embeddings among class varieties. *SIAM Journal on Computing*, 38(3):1175–1206, 2008.

[Mul99] K. Mulmuley. Lower bounds in a parallel model without bit operations. *SIAM Journal on Computing*, 28(4):1460–1509, 1999.

[Mul09a] K. Mulmuley. On P versus NP, geometric complexity theory and the Riemann hypothesis. Technical report, The University of Chicago, August 2009.

[Mul09b] K. Mulmuley. On P versus NP, geometric complexity theory, explicit proofs and the complexity barrier. Technical report, The University of Chicago, August 2009.

[NW97] N. Nisan and A. Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1997.

[Olv99] Peter Olver. *Classical Invariant Theory.* London Mathematical Society, 1999.

[Oxl06] J.G. Oxley. *Matroid Theory.* Oxford Graduate Texts in Mathematics. Oxford University Press, 2006.

[Pól71] G. Pólya. Aufgabe 424. *Arch. Math. Phys.*, 20:1913, 271.

[Raz09] R. Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *Journal of the Association for Computing Machinery*, 56(2), 2009.

[RR94] A.A. Razborov and S. Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55:204–213, 1994.

[RS04] R. Raz and A. Shpilka. Deterministic polynomial identity testing in non-commutative models. In *IEEE Conference on Computational Complexity*, pages 215–222, 2004.

[Rup99] W.M. Ruppert. Reducibility of polynomials $f(x, y)$ modulo $p$. *Journal of Number Theory*, 77:62–70, 1999.

[RY08] R. Raz and A. Yehudayoff. Lower bounds and separations for constant depth mutilinear circuits. In *Proceedings of the 23rd IEEE Annual Conference on Computational Complexity*, pages 128–139, 2008.

[Rys63] H.J. Ryser. *Combinatorial Mathematics.* Mathematical Association of America, 1963.

[Sax08] N. Saxena. Diagonal circuit identity testing and lower bounds. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming*, pages 60–71, 2008.

[Sch80] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the Association for Computing Machinery*, 27(4):701–717, 1980.

[Sch04] W.M. Schmidt. *Equations over Finite Fields.* Kendrick Press, 2004.

[Shp02] A. Shpilka. Affine projections of symmetric polynomials. *Journal of Computer and System Sciences*, 65(4):639–659, 2002.

[Sol02] M. Soltys. Berkowitz's algorithm and clow sequences. *Electronic Journal of Linear Algebra*, 9:42–54, 2002.

[SS09] N. Saxena and C. Seshadhri. Proceedings of the 24th annual ieee conference on computational complexity. In *IEEE Conference on Computational Complexity.* IEEE Computer Society, 2009.

[Str69] V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13:354–356, 1969.

[Str73a] V. Strassen. Die berechnungskomplexität von elementarsymmetrischen funktionen und von interpolationskoeffizienten. *Numerische Mathematik*, 20(3):238–251, 1973.

[Str73b] V. Strassen. Vermeidung von divisionen. *J. Reine Angew. Math.*, 264:184–202, 1973.

[Sud97] M. Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *Journal of Complex*, 13(1):180–193, 1997.

[SV08] A. Shpilka and I. Volkovich. Read-once polynomial identity testing. In *Proceedings of the 40th annual ACM Symposium on Theory of Computing*, pages 507–516, 2008.

[SW01] A. Shpilka and A. Wigderson. Depth-3 arithmetic circuits over fields of characteristic zero. *Computational Complexity*, 10(1):1–27, 2001.

[Sze92] G. Szegö. Zu aufgabe 424. *Arch. Math. Phys.*, 21:1913, 291–292.

[Val79a] L.G. Valiant. Completeness classes in algebra. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing*, pages 249–261, 1979.

[Val79b] L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.

[vdW37] B.L. van der Waerden. *Moderne Algebra.* Berlin, Springer, 2nd edition, 1937.

[VSBR83] L. G. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff. Fast parallel computation of polynomials using few processors. *SIAM Journal on Computing*, 12(4):641–644, 1983.

[vzG87] J. von zur Gathen. Permanent and determinant. *Linear Algebra and its Applications*, 96:87–100, 1987.

[vzGG99] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 1999.

[Woo96] T.D. Wooley. A note on simultaneous congruences. *Journal of Number Theory*, 58(2):288–297, 1996.

[Wri81] D. Wright. On the Jacobian conjecture. *Illinois Journal of Mathematics*, 15(3):423–440, 1981.

[Yu95] J.T. Yu. On the Jacobian conjecture: reduction of coefficients. *Journal of Algebra*, 171:515–523, 1995.

[Zip90] R. Zippel. Interpolating polynomials from their values. *Journal of Symbolic Computation*, 9(3):375–403, 1990.