Technische Universität Berlin
Institut für Mathematik

Bachelorarbeit

Algorithmic testing of equivalence of polynomials to the determinant

Andre Thorsten Weltsch

Berlin 2016

Erstgutachter: Prof. Dr. Peter Bürgisser
Zweitgutachter: Prof. Dr. Stefan Felsner

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Die selbständige und eigenständige Anfertigung versichert an Eides statt:

Berlin, den

Andre Thorsten Weltsch

Contents

1 Introduction

1.1 Context

In algebraic complexity theory one wants to investigate computational problems in an arithmetic model, e.g., common arithmetic operations $(+, -, \cdot, /)$ count as one operation. For example one can ask how many arithmetic operations it takes to compute a polynomial $f \in \mathbb{F}[X_1, \ldots, X_n]$. For a more formal definition of computation models see [BCS97]. VP and VNP (see [BCS97] or [Bür00]) are well known complexity classes of polynomial families in algebraic complexity theory. An analog to Cook's famous $P \neq NP$ hypothesis is Valiant's hypothesis that states $VP \neq VNP$. Similar to reductions in standard complexity theory the concept of p-projections (also [BCS97] or [Bür00]) is used to find relations among these complexity classes.
What can be seen as some form of generalization of p-projections are what we will call affine projections. A polynomial $f \in \mathbb{F}[X_1, \ldots, X_n]$ is said to be an affine projection of a polynomial $g \in \mathbb{F}[X_1, \ldots, X_m]$ if there is a matrix $A \in \mathbb{F}^{m \times n}$ and a vector $b \in \mathbb{F}^m$ s.t. $f(x) = g(Ax + b)$. Affine projections introduce a geometric notion for the problem of computability of related polynomials. In general, deciding if a polynomial is an affine projection of another polynomial is NP-hard [Kay12]. In his paper [Kay12] Neeraj Kayal shows that certain instances of the affine projection problem (equivalence of polynomials) for the determinant ($g = \det$) polynomial can be solved in probabilistic polynomial time. In this thesis we will deal with his approach to the problem.

1.2 The main result

The main problem that we will deal with is a special case of finding affine projections. In compliance with [Kay12] we define the polynomial equivalence problem PolyEquiv .

Name: PolyEquiv
Input: Polynomials $f(X_1, \ldots, X_n), g(X_1, \ldots, X_n) \in \mathbb{C}[X_1, \ldots, X_n]$
Output: An invertible matrix $A \in \mathrm{GL}_n$ s.t. $f(x) = g(Ax)$, if such an A exists. Else output `'No such equivalence exists'`.

What Kayal proved in [Kay12] is that there is a randomized algorithm that can solve PolyEquiv for the determinant polynomial in polynomial time. Our concern here is to develop the necessary means to state and understand the algorithm and to prove its correctness. When we talk about the time that an algorithm takes, we count the operations that it does when executed. We count additon, subtraction, multiplication, division and evaluation of a blackbox polynomial at a chosen point as one operation each.
In some cases the algorithm returns matrix A s.t. $f(X) \neq g(AX)$, but we can use the Schwart-Zippel-Lemma as a (probabilistic) check if the computed output is correct.

Theorem 1.1 ([Kay12], Theorem 4). *Under the assumption that there exists a routine that can diagonalize a matrix with pairwise distinct eigenvalues in polynomial time, there exists a randomized algorithm, that given an integer n and blackbox access to an n^2-variate polynomial f of degree n determines whether there exists a matrix $A \in GL_{n^2}$, s.t.*

$$f(X) = \det(A \cdot X)$$

The running time of the algorithm is $n^{O(1)}$ [(polynomial) in the number of operations].

1.3 The approach

Kayal investigates the group of symmetries $\mathcal{G}_f = \{A \in \mathrm{GL}_n(\mathbb{C}) \mid f(AX) = f(X)\}$ of a polynomial. The group of symmetries is a so called Lie Group and we can therefore look at its corresponding Lie algebra $\mathfrak{g}_f$.
A central observation is that if $f(X) = g(AX)$, then their Lie algebras are conjugate, $\mathfrak{g}_f = A^{-1}\mathfrak{g}_g A$. In the case of the determinant its group of symmetries has been studied and is explicitly known ([MM59]).
The algorithm consists of 3 computational steps, of which each reduces GL_n-equivalence of f and det to certain subgroups of GL_n. The first step is rather involved and uses special properties of $\mathfrak{g}_{\det}$, but it achieves to reduce GL_n-equivalence to equivalence with matrices that permute and scale the input variables (so called PS_n-equivalence).
The permutation part (the discrete part) of PS_n can be resolved by examining the Hessian of f, whereas the remaining scaling can easily be resolved by evaluating f at some carefully chosen permuation matrices.
A fourth step is added to check the correctness of the output. Unfortunately it is still only a probabilistic test, but there is considerable effort to derandomize identity tests for polynomials ([Sax09]).

1.4 Notation

$$\begin{aligned}
\mathrm{GL}_n &:= \mathrm{GL}(n, \mathbb{C}) \\
\mathrm{SL}_n &:= \mathrm{SL}(n, \mathbb{C}) \\
\mathrm{SC}_n &:= \mathrm{SC}(n, \mathbb{C}) := \{A \in \mathrm{GL}_n \mid A \text{ is diagonal}\} \\
\mathrm{PM}_n &:= \mathrm{PM}(n, \mathbb{C}) := \{A \in \mathrm{GL}_n \mid A \text{ is a permutation matrix}\} \\
\mathrm{PS}_n &:= \mathrm{PS}(n, \mathbb{C}) := \langle \mathrm{PM}_n, \mathrm{SC}_n \rangle = \{A_1 \cdot \ldots \cdot A_k \mid k \in \mathbb{N} \wedge A_i \in \mathrm{PM}_n \cup \mathrm{SC}_n\}
\end{aligned}$$

2 Basics on polynomials

This chapter will introduce a few advanced concepts related to polynomials, and I will give a thorough definition of what an affine projection of a polynomial is. [Bro89] is a good reference for algebraic geometric concepts.

2.1 Algebraic geometric concepts

Our first goal is to define the notion of probability that we are refering to throughout the thesis. Most sets that we look at are somehow defined by zero sets of polynomials (or their complements). To put it into an algebraic geometric perspective we introduce some of those concepts, and give them their names that are consistently used in the literature.

Definition 2.1. *Let $F \in \mathbb{C}[X_1, \dots, X_n]$ be an ideal of polynomials. We call the set*

$$V(F) := \{x \in \mathbb{C}^n \mid \forall f \in F : f(x) = 0\}$$

the zero set of F*. For single polynomials $f \in \mathbb{C}[X_1, \dots, X_n]$, we write*

$$V(f) := V(\{f\}).$$

Definition 2.2. *A subset $M \subseteq \mathbb{C}^n$ is called* algebraic *if*

$$M = V(J)$$

for some ideal $J \subseteq \mathbb{C}[X_1, \dots, X_n]$.

Definition 2.3. *A subset $M \subseteq \mathbb{C}^n$ is called* Zariski-closed *if it is algebraic. A subset $N \subseteq \mathbb{C}^n$ is called* Zariski-open *if its complement $\mathbb{C}^n \setminus N$ is Zariski-closed.*

Remark 2.4. *The set of Zariski-open sets on $\mathbb{C}^n$ is a topology.*

Proposition 2.5 ([BC13, Corollary A.35])**.** *Any Zariski-closed set $M \subsetneq \mathbb{C}^n$ properly contained in $\mathbb{C}^n$ has measure zero in $\mathbb{C}^n$.*

Based on proposition 2.5 we introduce a naming that is familiar from probability theory about sets of measure zero. This is the basis for the conecpt of probability that we use.

Definition 2.6. *A property P holds for* Zariski-almost-all *elements of $\mathbb{C}^n$ if the set*

$$\{x \in \mathbb{C}^n \mid P(x)\}$$

is Zariski-open.

Also related to probability is the follwoing proposition, the so called Schwartz-Zippel-Lemma. It yields an importatnt corollary that we can use to estimate the probability of a polynomial being identical to the zero polynomial.

Proposition 2.7 ([vG13, Lemma 6.44] (DeMillo-Lipton-Schwartz-Zippel-Lemma)). *Let $S \subseteq \mathbb{C}$ be a finite set with $s = |S|$ elements, and $r \in \mathbb{C}[X_1, \ldots, X_n]$ a polynomial of total degree at most $d \in \mathbb{N}$.*

1. *If $r \neq 0$, then r has at most ds^{n-1} zeroes in S^n.*
2. *If $s > d$ and r vanishes on S^n, then $r = 0$.*

We get the following corollary:

Corollary 2.8. *Let $S \subseteq \mathbb{C}$ be a finite set wit $s = |S|$ elements, and $r \in \mathbb{C}[X_1, \ldots, X_n]$ be a polynomial of total degree at most $d \in \mathbb{N}$. If we choose $a \in S^n$ uniformly random distributed, the probability of $r(a) = 0$ is*

$$prob(\{r(a) = 0 \mid a \in S^n\}) \leq \frac{d}{s}.$$

The Schwartz-Zippel can be used for efficient probabilistic identity tests on polynomials. Given $f, g \in \mathbb{C}[X_1, \ldots, X_n]$ we can check if $f = g$ by plugging in uniformly distributed random values from a finite set $S \subseteq \mathbb{C}$ for $X_1, \ldots, X_n$. The probability that $f - g = 0$ can be bounded by the previous corollary 2.8.
As a first application of this new concept of the Zariski-topology we will prove that almost surely a random matrix $A \in \mathbb{C}^{n \times n}$ has distinct eigenvalues. To prove this we have to examine a rather prominent polynomial first, the discriminant.

Definition 2.9 ([Bos13]). *Let $f, g \in \mathbb{C}[X]$ be polynomials*

$$\begin{aligned} f &= a_0 X^m + a_1 X^{m-1} + \ldots + a_m \\ g &= b_0 X^n + b_1 X^{n-1} + \ldots + b_n \end{aligned}$$

The resultant *of f and g is defined as*

$$res(f, g) := \det \begin{pmatrix} a_0 & a_1 & \ldots & a_m & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & a_0 & a_1 & \ldots & a_m \\ b_0 & b_1 & \ldots & b_n & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & b_0 & b_1 & \ldots & b_n \end{pmatrix} \in \mathbb{C}^{(n+m)\times(n+m)}.$$

The discriminant *of f is defined as*

$$disc(f) := (-1)^{\frac{m(m-1)}{2}} res(f, f').$$

The discriminant is a mighty tool to check if a polynomial has multiple zeros without having to factor the polynomial first.

Proposition 2.10 ([Bos13, Chapter 4.4, Bemerkung 3])**.** *Let*

$$f = \prod_{i=1}^{n}(X - \alpha_i) \in \mathbb{C}[X]$$

be a polynomial. Then disc $(f) = \prod_{i<j}(\alpha_i - \alpha_j)^2$. *In particular disc* $(f) = 0$ *if and only if* f *has multiple zeros.*

As a consequence we can show that Zariski-almost-all matrices are diagonalizable.

Lemma 2.11. *Zariski-almost-all matrices* $A \in \mathbb{C}^{n\times n}$ *have pairwise distinct eigenvalues.*

Proof. Let

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \in \mathbb{C}^{n\times n}.$$

We know that the eigenvalues of a matrix are the zeros of its characteristic polynomial

$$\chi(A) = \det(A - XI_n).$$

We now want to look at $\chi(A)$ as a polynomial in the entries of A and in X, i.e. $\chi(A) \in \mathbb{C}[a_{11}, \dots, a_{nn}][X]$. We know that the discriminant is zero iff $\chi(A)$ has zeros with multiplicities greater than 1, i.e. A does not have distinct eigenvalues. Thus the set of matrices with pairwise distinct eigenvalues is nonempty and Zariski-open. $\square$

Given polynomials $f_1, \dots, f_m$ we want to find a linear dependence among them, i.e. find $a_i \in \mathbb{C}$ s.t.

$$\sum a_i f_i = 0$$

The next lemma and corollary will show that we can achieve this by evaluating the f_i at random points and solving a related system of linear equations. We will need this later to calculate the basis of Lie algebras of polynomials. The proof is a transcription from [Kay11].

Lemma 2.12 ([Kay11, Claim 7])**.** *Let* $f_1, \dots, f_m \in \mathbb{C}[X_1, \dots, X_n]$. *Let*

$$t = \dim \left\{ b \in \mathbb{C}^m \mid \sum_{i\in[m]} f_i b_i = 0 \right\}.$$

Consider the additional variables

$$Y_{11}, \dots, Y_{1n}, \dots, Y_{m1}, \dots, Y_{mn}$$

and the field of rational functions $\mathbb{C}(Y_{11}, \dots, Y_{mn})$. *Let* $P(Y_1, \dots, Y_m)$ *be the matrix*

$$P(Y_1, \dots, Y_m) := \begin{pmatrix} f_1(Y_{11}, \dots, Y_{1n}) & \dots & f_m(Y_{11}, \dots, Y_{1n}) \\ \vdots & \ddots & \vdots \\ f_1(Y_{m1}, \dots, Y_{mn}) & \dots & f_m(Y_{m1}, \dots, Y_{mn}) \end{pmatrix}.$$

Then P *has rank* $(m - t)$.

Proof. Without loss of generality we assume that the first $(m-t)$ polynomials are $\mathbb{C}$-linearly independent, i.e. $f_1, \dots, f_{m-t}$ are linearly independent. It suffices to show that the quadratic submatrix

$$P := \begin{pmatrix} f_1(Y_{11}, \dots, Y_{1n}) & \dots & f_m(Y_{11}, \dots, Y_{1n}) \\ \vdots & \ddots & \vdots \\ f_{m-t}(Y_{(m-t)1}, \dots, Y_{(m-t)n}) & \dots & f_{m-t}(Y_{(m-t)1}, \dots, Y_{(m-t)n}) \end{pmatrix}$$

has rank full rank $(m-t)$. This is equivalent to showing that $\det(P) \neq 0$.
We show this by induction on $(m-t)$:
$(m-t) = 1$: $P = (f(Y_{11}, \dots, Y_{1n}))$ is a nonzero polynomial, hence $\det(P) \neq 0$.
$(m-t) - 1 \to (m-t)$: We use Laplace expansion along the first row of P and get

$$\det P = \sum_{i=1}^{(m-t)} (-1)^{i+1} f_i(Y_{11}, \dots, Y_{1n}) \det(P_{1i}(Y_{21}, \dots, Y_{2n}, \dots, Y_{(m-t)1}, \dots, Y_{(m-t)n}))$$

where P_{ij} is the matrix that we get by removing row i and column j from P (commonly referred to as minor). Removing the j-th column removes one polynomial from the set that we examine and as every subset of $f_1, \dots, f_{m-t}$ is $\mathbb{C}$-linearly independent, we can use the induction hypothesis. Therefore $\det(P_{1i}) \neq 0$ for all i.
If we assume $\det(P) = 0$, we could then find a $\mathbb{C}$-linear dependence of

$$f_1(Y_{11}, \dots, Y_{1n}), \dots, f_{m-t}(Y_{11}, \dots, Y_{1n})$$

by plugging in random values in $\mathbb{C}$ for $Y_{21}, \dots, Y_{(m-t)n}$. This yields a contradiction ☐

Corollary 2.13. *For the same assumptions as in the previous lemma 2.12 for Zariski-almost-all $A = (a_1, \dots, a_m) \in (\mathbb{C}^n)^m$ the matrix*

$$P(a_1, \dots, a_m) = \begin{pmatrix} f_1(a_1) & \dots & f_m(a_1) \\ \vdots & \ddots & \vdots \\ f_1(a_m) & \dots & f_m(a_m) \end{pmatrix}.$$

has rank $(m-t)$.

2.2 Polynomial equivalence

This section contains the very definitions that underlie the PolyEquiv problem that we are trying to solve here.

Definition 2.14. *Let $f \in \mathbb{C}[X_1, \dots, X_n]$ and $g \in \mathbb{C}[Y_1, \dots, Y_m]$. f is called an* affine projection *of g if there exists a matrix $A \in \mathbb{C}^{m \times n}$ and a vector $b \in \mathbb{C}^m$, s.t.*

$$\forall x \in \mathbb{C}^n : f(x) = g(Ax + b)$$

If $m = n$, $A \in GL_n$ and $b = 0$ we say that f and g are equivalent.
*Further if $G \leq GL_n$ is a subgroup of GL_n, then we we say f, g are G-*equivalent *if there exists a matrix $A \in G$, s.t. $f(x) = g(Ax)$.*

Definition 2.15. *Let $f \in \mathbb{C}[X_1, \dots, X_n]$ be a polynomial. We call the set*

$$\mathcal{G}_f := \{A \in GL_n \mid f(Ax) = f(x)\}$$

the group of symmetries *of f.*

3 Lie groups and Lie algebras

This chapter will give some basic definitions and theorems about Lie groups and Lie algebras that are relevant for the algorithm.
Lie groups are geometric objects by nature, a Lie group G is usually defined as a smooth manifold equipped with a group structure s.t. the multiplication and inverse are continuous functions on G. Historically Lie algebras have been studied as tangent spaces to Lie groups at the identity element. Fortunately the Lie groups that are of interest for us can be described in a much more basic manner, s.t. we can limit the geometric prerequisites to a minimum.
What we will learn is that the symmetries $\mathcal{G}_f$ of a polynomial are indeed a Lie group and we can therefore examine its associated Lie algebra $\mathfrak{g}_f$. In particular $\mathcal{G}_{\det}$ is a Lie group and we can study its Lie algebra $\mathfrak{g}_{\det}$, what we will learn is that it is closely related to $\mathfrak{sl}_n$ the Lie algebra of SL_n.
Central to this chapter will be lemma 4.5: the observation that if a polynomial f is equivalent to a polynomial g via $A \in \mathrm{GL}_n$, i.e. $f(X) = g(AX)$, then their Lie algebras are conjugate:

$$\mathfrak{g}_f = A^{-1}\mathfrak{g}_g A.$$

This is the foundation for the algorithm that will be described later.
[Hal03] serves as a great reference for basic matrix Lie groups, most of the proofs in this chapter are transcriptions of proofs from this book.

3.1 General notions of Lie algebras

The most general definition of a Lie algebra is independent of any Lie groups, but we will soon have a look at Lie algebras as tangent space to some Lie group. In this section we simply introduce common definitions related to Lie algebras.

Definition 3.1 ([Hal03, Definition 2.36])**.** *A finite-dimensional complex* Lie algebra *is a finite-dimensional complex vector space* $\mathfrak{g}$*, together with a map*

$$[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$$

with the following properties:

1. $[\cdot,\cdot]$ *is bilinear*
2. $[X, Y] = -[Y, X]$ *for all* $X, Y \in \mathfrak{g}$
3. $[X, [Y, Z]] + [Y, [Z, X]] + [Z, [X, Y]] = 0$ *for all* $X, Y, Z \in \mathfrak{g}$

Example 3.2. *The space of complex matrices* $\mathbb{C}^{n\times n}$ *equipped with the bracket operation* $[X, Y] = XY - YX$ *for all* $X, Y \in \mathbb{C}^{n\times n}$ *is a Lie algebra. We denote it by* $\mathfrak{gl}_n$*.*

Definition 3.3. *Let $\mathfrak{g}$ be a Lie algebra and $\mathfrak{h} \subseteq \mathfrak{g}$ be a subspace. We call $\mathfrak{h}$ a* subalgebra *of $\mathfrak{g}$ if*

$$[H_1, H_2] \in \mathfrak{h}$$

for all $H_1, H_2 \in \mathfrak{h}$.

Definition 3.4. *Let $\mathfrak{g}$ and $\mathfrak{h}$ be Lie algebras. A linear map $\phi : \mathfrak{g} \to \mathfrak{h}$ is called a* Lie algebra homomorphism *if*

$$[\phi(X), \phi(Y)] = \phi([X, Y])$$

for all $X, Y \in \mathfrak{g}$.

Definition 3.5. *Let $\mathfrak{g}$ be a Lie algebra and let $A \in \mathfrak{g}$. The* centraliser *of A, denoted by $Cent_{\mathfrak{g}}(A)$, is defined as*

$$Cent_{\mathfrak{g}}(A) := \{X \in \mathfrak{g} \mid [A, X] = 0\}$$

If the bracket operation is defined as in example 3.2 the centraliser of an element is the set of its commuting matrices.
To understand the composite structure of $\mathfrak{g}_{\det}$ and its relation to $\mathfrak{sl}_n$ we want to examine the structure of Lie algebras of products of Lie groups.

Definition 3.6. *If $\mathfrak{g}_1$ and $\mathfrak{g}_2$ are Lie algebras we define the* direct sum *of $\mathfrak{g}_1$ and $\mathfrak{g}_2$ as follows:*

$$\mathfrak{g}_1 \oplus \mathfrak{g}_2 := \mathfrak{g}_1 \times \mathfrak{g}_2$$

with the following bracket operation:

$$[(X_1, X_2), (Y_1, Y_2)] = ([X_1, Y_1], [X_2, Y_2]) \text{ for } X_1, Y_1 \in \mathfrak{g}_1, X_2, Y_2 \in \mathfrak{g}_2$$

It is easy to see that the composite bracket operation defines a valid Lie algebra structure on the product.

3.2 Matrix Lie groups and the matrix exponential

What we will learn shortly is that the symmetries of $\det_n$ (a Lie group) is closely related to SL_n. We want to study its Lie algebra $\mathfrak{sl}_n$ and a special form of Lie groups, the so called matrix Lie groups, that will help us to understand $\mathfrak{sl}_n$.

Definition 3.7. *A* matrix Lie group *is any subgroup G of GL_n s.t. for any sequence $(A_m)_m \in G$ it holds that if $(A_m)_m$ converges to a matrix $A \in \mathbb{C}^{n\times n}$, then either $A \in G$ or or $A \notin GL_n$.*

This means that matrix Lie groups are closed (in the standard topology) subgroups of GL_n.

Example 3.8. *SL_n is a matrix Lie group.*

Definition 3.9. *For $X \in \mathbb{C}^{n \times n}$, we define the* matrix exponential *of X as*

$$e^X := \sum_{m=0}^{\infty} \frac{X^m}{m!}$$

The key to understanding the Lie algebra of a matrix Lie group lies in the matrix exponential. Thus we will check some properties of it.

Remark 3.10 ([Hal03], Proposition 2.3 Theorem 2.11,)**.** *Let $X, Y \in \mathbb{C}^{n \times n}$ and $C \in GL_n$, then*

1. *e^X converges*
2. *If $XY = YX$, then $e^{X+X} = e^X e^Y = e^Y e^X$*
3. *$(e^X)^{-1} = e^{-X}$*
4. *$\frac{d}{dt} e^{tX} = X e^{tX}$*
5. *$e^{CXC^{-1}} = C e^X C^{-1}$*
6. *$\det(e^X) = e^{trace(X)}$*

Proof. Let $X \in \mathbb{C}^{n \times n}$ and $C \in \mathrm{GL}_n$.

1. Let $\|\cdot\|$ be a matrix norm, then

$$\left\| e^X \right\| = \left\| \sum_{m=0}^{\infty} \frac{X^m}{m!} \right\| \leq \sum_{m=0}^{\infty} \frac{\|X^m\|}{m!} \leq \sum_{m=0}^{\infty} \frac{\|X\|^m}{m!} = e^{\|X\|}.$$

So this series converges absolutely and therefore it converges.

2. Let $XY = YX$. We get

$$(X+Y)^m = \sum_{k=0}^{m} \frac{m!}{k!(m-k)!} X^k Y^{m-k}.$$

Now we look at $e^X e^Y$:

$$\begin{aligned} e^X e^Y &= \sum_{m=0}^{\infty} \sum_{k=0}^{m} \frac{X^k}{k!} \frac{Y^{m-k}}{(m-k)!} \\ &= \sum_{m=0}^{\infty} \frac{1}{m!} \sum_{k=0}^{m} \frac{m!}{k!(m-k)!} X^k Y^{m-k} \\ &= \sum_{m=0}^{\infty} \frac{(X+Y)^m}{m!} = e^{X+Y}. \end{aligned}$$

3. X and $-X$ commute, thus by the previous point
$$e^X e^{-X} = e^{X-X} = e^0.$$

4. By its definition every entry of e^{tX} is a convergent power series and we can differentiate e^{tX} in every entry term by term.
$$\begin{aligned}\frac{d}{dt}e^{tX} &= \frac{d}{dt}\sum_{m=0}^{\infty}\frac{t^m X^m}{m!}\\ &= \sum_{m=0}^{\infty}\frac{d}{dt}\frac{t^m X^m}{m!}\\ &= \sum_{m=0}^{\infty} m\cdot\frac{t^{m-1}X^m}{m!}\\ &= \sum_{m=1}^{\infty} X\frac{t^{m-1}X^{m-1}}{(m-1)!}\\ &= Xe^{tX}\end{aligned}$$

5. We note that
$$(CXC^{-1})^m = CX^mC^{-1}.$$
Therefore
$$e^{CXC^{-1}} = \sum_{m=0}^{\infty}\frac{(CXC^{-1})^m}{m!} = \sum_{m=0}^{\infty}\frac{CX^mC^{-1}}{m!} = Ce^XC^{-1}.$$

6. Let X be diagonalizable, then we find $A \in \mathrm{GL}_n$ s.t. $AXA^{-1} = D = \mathrm{diag}\,(d_1, \ldots, d_n)$ is diagonal. We get
$$\begin{aligned}\det(e^X) &= \det(e^{A^{-1}DA}) = \det(A^{-1}e^DA)\\ &= \det(A^{-1})\det(A)\det(e^D) = \det\left(\mathrm{diag}\left(e^{d_1}, \ldots, e^{d_2}\right)\right)\\ &= \prod_{i=1}^{n} e^{d_i} = e^{\sum_{i=1}^n d_i} = e^{\mathrm{trace}(D)} = e^{\mathrm{trace}(X)}.\end{aligned}$$
The set of diagonalizable matrices is dense in $\mathbb{C}^{n\times n}$, therefore by continouity this holds for all $X \in \mathbb{C}^{n\times n}$. □

A little deeper than the previously mentioned propertious of the matrix exponential is given here, a proof will not be given.

Proposition 3.11 ([Hal03], Theorem 2.10 (Lie Product Formula))**.** *Let $X, Y \in \mathbb{C}^{n\times n}$, then*
$$e^{X+Y} = \lim_{m\to\infty}(e^{\frac{X}{m}}e^{\frac{Y}{m}})^m$$

Proof. See [Hal03]. $\square$

Definition 3.12. *Let G be a matrix Lie group. The Lie algebra of G, denoted $\mathfrak{g}$ is defined as*

$$\mathfrak{g} := \left\{X \in \mathbb{C}^{n\times n} \mid \forall t \in \mathbb{R} : e^{tX} \in G\right\}$$

As a side note it is nice to know, that this definition of Lie algebra actually defines the tangent space to the matrix Lie group at the identity. To see this just note that e^{tA} is a smooth curve for each A and for each invertible matrix X you can actually find an A s.t. $X = e^A$ (we will not prove this here).
What remains to show is that our new definition of Lie algebra matches with the general one from the previous chapter.

Remark 3.13 ([Hal03, Theorem 2.18])**.** *A Lie algebra (in the sense of definition 3.12) $\mathfrak{g}$ of a matrix Lie group G is a Lie algebra (in the sense of definition 3.1) equipped with the bracket operation $[X,Y] = XY - YX$ for all $X, Y \in \mathfrak{g}$.*

Proof. First we want to show, that $\mathfrak{g}$ is a linear subspace of $\mathbb{C}^{n\times n}$. This implies that it is closed.
$0 \in \mathfrak{g}$ because $e^0 = I_n \in G$.
Let $s \in \mathbb{C}$ be an arbitrary scalar, and $X \in \mathfrak{g}$, we know that for all $t \in \mathbb{R}$ $e^{s(tX)} = e^{(st)X} \in \mathfrak{g}$.
Let $X, Y \in \mathfrak{g}$, then

$$e^{t(X+Y)} \underbrace{=}_{3.11} \lim_{m\to\infty} (e^{t\frac{X}{m}} e^{\frac{tY}{m}})^m$$

G is a group thus for all $m \in \mathbb{N}$ $e^{\frac{tX}{m}} e^{\frac{tY}{m}} \in G$. Also $e^{t(X+Y)}$ is invertible by 3.10. So by definition of a matrix Lie group we get $\lim_{m\to\infty}(e^{\frac{tX}{m}} e^{\frac{tY}{m}})^m \in G$.
To show $XY - YX \in \mathfrak{g}$, we look at the derivative

$$\frac{d}{dt} e^{tX} Y e^{-tX}|_{t=0} = \big((XY)e^{-tX} + (e^{tX}Y)(-X)\big)|_{t=0} = XY - YX.$$

Using matrix exponential properties we get

$$e^{e^{tX}Ye^{-tX}} = e^{tX}e^{Y}e^{-tX} \in G.$$

So by the definition of the derivative

$$XY - YX = \lim_{h\to 0} \frac{e^{hX}Ye^{-hX} - Y}{h} \in \mathfrak{g}$$

because $\mathfrak{g}$ is closed. $\square$

Finally we can determine $\mathfrak{sl}_n$ by using the properties of the matrix exponential.

Corollary 3.14. *The Lie algebra of SL_n denoted by $\mathfrak{sl}_n$ is given by*

$$\mathfrak{sl}_n := \left\{A \in \mathbb{C}^{n\times n} \mid \mathit{trace}(A) = 0\right\}$$

Proof. By definition of a Lie algebra $A \in \mathfrak{sl}_n$ if for all $t \in \mathbb{R}$ we have $e^{tA} \in \mathrm{SL}_n$, thus with remark 3.10

$$\det\left(e^{tA}\right) = e^{\mathrm{trace}(tA)} = e^{t \cdot \mathrm{trace}(A)} \overset{!}{=} 1.$$

So $A \in \mathfrak{sl}_n$ if and only if $\forall t \in \mathbb{R} : t \cdot \mathrm{trace}\,(A) = 0 \iff \mathrm{trace}\,(A) = 0$. □

We have already defined a bracket operation on the product Lie algebras, thus we can check that this fulfills indeed all properties of definition 3.1. As a consequence the Lie algebra of a direct product of matrix Lie groups is just the direct sum of the Lie algebras of the factors.

Remark 3.15. *Let $G_1 \subseteq GL_{n_1}$ and $G_2 \subseteq GL_{n_2}$ be matrix Lie groups with corresponding Lie algebras $\mathfrak{g}_1$, $\mathfrak{g}_2$. Then $G_1 \times G_2 \subseteq GL_{n_1+n_2}$ is a matrix Lie group and its Lie algebra is $\mathfrak{g}_1 \oplus \mathfrak{g}_2$.*

Theorem 3.16 ([Hal03, Theorem 2.21])**.** *Let G and H be matrix Lie groups, with Lie algebras $\mathfrak{g}$ and $\mathfrak{h}$, respectively. Let $\Phi : G \to H$ be a smooth group homomorphism. Then, its derivative $\phi(X) := \frac{d}{dt}\Phi(e^t X)$ is the unique Lie algebra homomorphism $\phi : \mathfrak{g} \to \mathfrak{h}$ s.t.*

$$\Phi(e^X) = e^{\phi(X)}$$

for all $X \in \mathfrak{g}$. ϕ has the special property:

$$\phi(AXA^{-1}) = \Phi(A)\phi(X)\Phi(A)^{-1} \textit{ for all } X \in \mathfrak{g}, A \in G$$

Proof. See [Hal03]. □

4 Lie algebras of polynomials

After a rather short introduction to matrix Lie group we are ready to examine the objects of our actual interests. It is important to notice that the symmetry groups of polynomials are matrix Lie groups.
This section is meant to ground Kayal's observations about the group of symmetries of the determinant directly on the fundament of matrix Lie groups, that we just developed.
Let us reconsider our problem for an instant:
Given $f, g \in \mathbb{C}[X_1, \ldots, X_n]$ we want to find $A \in \mathrm{GL}_n$ s.t. $f(X) = g(AX)$.
There are two important lemmas in this section that will help us on our quest of finding A.
The first lemma (4.5) states that

$$\mathfrak{g}_f = A^{-1}\mathfrak{g}_g A$$

which is the basis to understanding why we consider the Lie algebras $\mathfrak{g}_f$ and $\mathfrak{g}_g$ to find A.
The second lemma (4.12) is more specific to the determinant polynomial, and has a more subtle consequence that we will only understand later. Nonetheless it is important to remember for the later proofs:
For Zariski-almost-all elements A in its Lie algebra $\mathfrak{g}_{\det}$, we can find an element S in its *group of symmetries* $\mathcal{G}_{\det}$ s.t. SAS^{-1} is a diagonal matrix.

4.1 Symmetries of a polynomial as matrix Lie groups

The symmetries of a polynomial (defintion 2.15) are a subgroup of GL_n, what we will show now, is that these symmetries are closed in the standard topology and hence are matrix Lie groups.

Theorem 4.1. *Let $f \in \mathbb{C}[X_1, \ldots, X_n]$. Then its group of symmetries $\mathcal{G}_f \subseteq GL_n$ is a matrix Lie group. The corresponding Lie algebra is denoted by $\mathfrak{g}_f$.*

Proof. Let $f \in \mathbb{C}[X_1, \ldots, X_n]$ be a polynomial. Due to the fact that polynomials are continuous in the standard topology we can do the following:
Let $(A_m)_m \subseteq \mathcal{G}_f$ be a converging sequence in $\mathcal{G}_f$, and let $A = \lim_{m\to\infty} A_m \in \mathbb{C}^{n\times n}$. Then we know that

$$\begin{aligned} f(AX) &= f(\lim_{m\to\infty} A_m X) \\ &= \lim_{m\to\infty} f(A_m X) = f(X) \end{aligned}$$

Hence $A \in \mathcal{G}_f$. Thus $\mathcal{G}_f$ is a matrix Lie group. □

Lie algebras are vector spaces and therefore have a basis. The next theorem will give us a system of linear equations that describe the Lie algebra of the symmetries of a polynomial. Solving these equations will lead to an algorithm that can find a basis for the Lie algebra in random polynomial time.

Theorem 4.2 ([Kay12, Claim 59])**.** *Let $f \in \mathbb{C}[X_1, \dots, X_n]$, then $A = (a_{ij}) \in \mathfrak{g}_f$ if and only if*

$$\sum_{i,j\in[n]} a_{ij} X_j \frac{\partial f}{\partial X_i} = 0 \tag{1}$$

Proof. Let assume $A \in \mathbb{C}^{n\times n}$. Define $\hat{f}(t) = f(e^{tA}x) - f(x) \in \mathbb{C}[X_1, \dots, X_n][t]$ First we have a look at the derivative and note

$$\frac{d}{dt}\hat{f}(t) = \frac{d}{dt} f(e^{tA}x) = f'(e^{tA}x) \cdot A \cdot e^{tA} \cdot x, \tag{2}$$

where f' denotes the gradient $(\partial f/\partial X_1, \dots, \partial f/\partial X_n)$ of f. Now $A \in \mathfrak{g}_f$, i.e. for all $t \in \mathbb{R}$ $f(e^{tA}x) = f(x)$. Hence $\hat{f} = f(e^{tA}x) - f(x) = 0$ is the zero polynomial, so its derivative is also the zero polynomial.
On the other hand,

$$\begin{aligned} 0 = \frac{d}{dt}(f(e^{tA}x) - f(x))|_{t=0} &= (f'(e^{tA}x)Ae^{tA}x)|_{t=0} \\ &= f'(x)Ax \\ &= \sum_{i,j\in[n]} a_{ij} X_j \frac{\partial f}{\partial X_i}. \end{aligned}$$

For the reverse direction let $A \in \mathbb{C}^{n\times n}$ be s.t. equation (1) holds. We substitute $x \mapsto e^{tA}x$ and get

$$0 = \sum_{i,j\in[n]} a_{ij}(e^{tA}x)_j \frac{\partial f}{\partial X_i}(e^{tA}x) = f'(e^{tA})Ae^{tA}x \underbrace{=}_{(2)} \frac{d}{dt}\hat{f}$$

As a consequence $\hat{f}$ is constant, also $\hat{f}(0) = f(e^{0A}x) - f(x) = 0$, therefore $\hat{f}(t) = 0$ for all $t \in \mathbb{R}$, so $\forall t \in \mathbb{R} : e^{tA} \in \mathcal{G}_f$. By the definition of the Lie algebra we have $A \in \mathfrak{g}_f$. □

We will now present another equivalent description of $\mathfrak{g}_f$. [Kay12] uses this as definition, and we will relate this to our previous examination of $\mathfrak{g}_f$, we will also use this to prove a fact about Lie algebra conjugacy.

Lemma 4.3 ([Kay12, Lemma 22])**.** *Let $f \in \mathbb{C}[X_1, \dots, X_n]$ and $\varepsilon \neq 0$ be an artificial variable s.t. $\varepsilon^2 = 0$. Then $A \in \mathfrak{g}_f$ if and only if $f((I_n + \varepsilon A)X) = f(x)$.*

Proof. We have a look at $f((I_n + A)X) - f(X)$ and claim:

Claim 4.4 ([Kay12, Claim 59])**.** $f((I_n + A)X) - f(X) = \varepsilon(\sum_{i,j\in[n]} a_{ij} X_j \frac{\partial f}{\partial X_i})$

Proof of claim. It is sufficient to show this if f is a monomial, $f = \prod_{i=1} X_i^{k_i}$. Let δ_{ij}

denote the Kronecker-Delta.

$$\begin{aligned}
f\left(\left(I_n + \varepsilon A\right) X\right) &= f\left(\sum_{j=1}^{n} \left(\delta_{1j} + \varepsilon a_{1j}\right) X_j, \ldots, \sum_{j=1}^{n} \left(\delta_{nj} + \varepsilon a_{nj}\right) X_j\right) \\
&= \prod_{i=1}^{n} \left(\sum_{j=1}^{n} \left(\delta_{ij} + \varepsilon a_{ij}\right) X_j\right)^{k_i} \\
&= \prod_{i=1}^{n} \left(X_i + \varepsilon \left(\sum_{j=1}^{n} a_{ij} X_j\right)\right)^{k_i} \\
&= \prod_{i=1}^{n} \left(X_i^{k_i} + \varepsilon X_i^{k_i - 1} k_i \left(\sum_{j=1}^{n} a_{ij} X_j\right)\right) \\
&= \prod_{i=1}^{n} X_i^{k_i} + \varepsilon \sum_{i=1}^{n} \left(k_i \prod_{m=1}^{n} X_i^{k_i - \delta_{mj}} \left(\sum_{j=1}^{n} a_{ij} X_j\right)\right) \\
&= \prod_{i=1}^{n} X_i^{k_i} + \varepsilon \sum_{i=1}^{n} \left(\frac{\partial f}{\partial X_i} \left(\sum_{j=1}^{n} a_{ij} X_j\right)\right) \\
&= \prod_{i=1}^{n} X_i^{k_i} + \varepsilon \left(\sum_{i,j \in [n]} a_{ij} X_j \frac{\partial f}{\partial X_i}\right)
\end{aligned}$$

□

From the claim we conclude with theorem 4.2 $f((I_n + A)X) - f(X) = 0$ if and only if $A \in \mathfrak{g}_f$. □

Finally we arrive at the central lemma that is the fundament of the final algorithm. Lie algebras of equivalent polynomials are conjugate via their transformation matrix. The proof of this lemma is a transcription from [Kay12].

Lemma 4.5 ([Kay12, Proposition 58]). *Let $f, g \in \mathbb{C}[X_1, \ldots, X_n]$. If there exists a nonsingular matrix $A \in GL_n$ s.t. $f(X) = g(AX)$, then their Lie algebras are conjugate via A, i.e.*

$$\mathfrak{g}_f = A^{-1} \mathfrak{g}_g A \tag{3}$$

Proof. Let $B \in \mathfrak{g}_f$. We note that

$$B \in A^{-1} \mathfrak{g}_g A \iff ABA^{-1} \in \mathfrak{g}_g$$

Hence we look at

$$\begin{aligned}
g((I_n + \varepsilon ABA^{-1})x) &= g(A(A^{-1} + \varepsilon BA^{-1})x) \\
&\overset{f(x)=g(Ax)}{=} f((A^{-1} + \varepsilon BA^{-1})x) \\
&= f((I_n + \varepsilon B)A^{-1}x) \\
&\overset{B \in \mathfrak{g}_f}{=} f(A^{-1}x) = g(AA^{-1}x) = g(x)
\end{aligned}$$

This implies $ABA^{-1} \in \mathfrak{g}_g$. The reverse direction works in the same way. $\square$

From the previous lemma we can deduce that the centralisers of conjugate elements are conjugate.

Corollary 4.6. *Let $f, g \in \mathbb{C}[X_1, \dots, X_n]$. If $f(x) = g(AX)$ for some $A \in GL_n$, and let $B \in \mathfrak{g}_f$. Then there exists $C \in \mathfrak{g}_g$, s.t.*

$$Cent(B) = A^{-1} Cent(C)\, A$$

Proof. Let $B \in \mathfrak{g}_f$, then by the previous lemma 4.5 $C := ABA^{-1} \in \mathfrak{g}_f$. Let $D \in \mathrm{Cent}\,(C)$, then we have

$$[B, A^{-1}DA] = BA^{-1}DA - A^{-1}DAB = A^{-1}\underbrace{(ABA^{-1}D - DABA^{-1})}_{=0}A = 0.$$

This implies that $A^{-1}DA \in \mathrm{Cent}\,(B)$, hence we get $\mathrm{Cent}\,(B) \supseteq A^{-1}\mathrm{Cent}\,(C)\,A$.
The reverse direction works in the same way. $\square$

4.2 The Lie algebra of the determinant polynomial

This chapter is dedicated to examine the Lie group and Lie algebra of the determinant polynomial $\det_n \in \mathbb{C}[X_{11}, \dots, X_{nn}]$.
There have been several attempts to describe the symmetry group $\mathcal{G}_{\det}$, a rather short and basic overview is given in [MM59] a thorough examination can be found in [Rei16].

Theorem 4.7 ([MM59, Theorem 2])**.** *Let $T : \mathbb{F}^{n\times n} \to \mathbb{F}^{n\times n}$ be a linear transformation of matrices. If T preserves* $\det$*, i.e.*

$$\forall M \in \mathbb{F}^{n\times n} : \det M = \det T(M)$$

then there exist matrices $U, V \in SL_n$ s.t. for all $X \in \mathbb{F}^{n\times n}$ either

$$T(X) = UXV$$

or

$$T(X) = UX^TV$$

Theorem 4.7 will gives us the opportunity to embed the structure into GL_{n^2}. Thanks to our previous investigation of Lie algebras in relation to Lie groups we can easily deduce the structure of $\mathfrak{g}_{\det}$.

Corollary 4.8. *The Lie algebra of the determinant $\mathfrak{g}_{\det}$ is isomorphic to $\mathfrak{sl}_n \oplus \mathfrak{sl}_n$ via the isomorphism*

$$\begin{aligned}\phi : \mathfrak{sl}_n \oplus \mathfrak{sl}_n &\to \mathfrak{g}_{\det}\\ (A, B) &\mapsto (X \mapsto AX + XB^T)\end{aligned}$$

Proof. We define the group homomorphism

$$\begin{aligned}\Phi : \mathrm{SL}_n \times \mathrm{SL}_n &\to \mathcal{G}_{\det} \\ (A, B) &\mapsto (X \mapsto AXB^T).\end{aligned}$$

Hint: We identify $\mathrm{End}\left(\mathbb{C}^{n\times n}\right) \cong \mathbb{C}^{n^2\times n^2}$.
By theorem 4.7 Φ maps $\mathrm{SL}_n \times \mathrm{SL}_n$ to $\mathcal{G}_{\det}$. Theorem 3.16 tells us that

$$\begin{aligned}\phi : \mathfrak{sl}_n \otimes \mathfrak{sl}_n &\to \mathfrak{g}_{\det} \\ (A, B) &\mapsto \frac{d}{dt}\Phi(e^{tX})|_{t=0} = \frac{d}{dt}(X \mapsto e^{tA}Xe^{tB^T})|_{t=0} \\ &= (X \mapsto AX + XB^T)\end{aligned}$$

is a Lie algebra homomorphism. Now we show that ϕ is injective by showing that $\ker\phi = \{(0, 0)\}$. Asuume $A, B \in \mathfrak{sl}_n$ s.t. $\phi(A, B) = 0$. This means

$$\forall X \in \mathbb{C}^{n\times n} : \phi(A, B)(X) = 0.$$

First we only consider diagonal matrices $X = \mathrm{diag}\,(x_1, \ldots, x_n) \in \mathbb{C}^{n\times n}$:

$$\begin{aligned}0 &\overset{!}{=} \phi(A, B)(X) = AX + XB^T \\ &= \begin{pmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \ldots & a_{nn}\end{pmatrix} \mathrm{diag}\,(x_1, \ldots, x_n) + \mathrm{diag}\,(x_1, \ldots, x_n) \begin{pmatrix} b_{11} & \ldots & b_{n1} \\ \vdots & \ddots & \vdots \\ b_{1n} & \ldots & a_{nn}\end{pmatrix} \\ &= \begin{pmatrix} a_{11}x_1 + b_{11}x_1 & a_{12}x_2 + b_{21}x_1 & \ldots & a_{1n}x_n + b_{n1}x_1 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}x_1 + b_{1n}x_n & a_{n2}x_2 + b_{2n}x_n & \ldots & a_{nn}x_n + b_{nn}x_n\end{pmatrix}\end{aligned}$$

Thus for $a_{ij} = b_{ji} = 0$ for $i \neq j$. And then with $X \in \mathbb{C}^{n\times n}$ arbitrary:

$$\begin{aligned}0 &\overset{!}{=} \mathrm{diag}\,(a_{11}, a_{22}, \ldots, a_{nn}) \begin{pmatrix} x_{11} & \ldots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \ldots & x_{nn}\end{pmatrix} + \begin{pmatrix} x_{11} & \ldots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \ldots & x_{nn}\end{pmatrix} \mathrm{diag}\,(b_{11}, b_{22}, \ldots, b_{nn}) \\ &= \begin{pmatrix} (a_{11} + b_{11})x_{11} & (a_{11} + b_{22})x_{12} & \ldots & (a_{11} + b_{nn})x_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ (a_{nn} + b_{11})x_{11} & (a_{nn} + b_{22})x_{n2} & \ldots & (a_{nn} + b_{nn})x_{nn}\end{pmatrix}\end{aligned}$$

A consequence of this is $a_{11} = \ldots = a_{nn}$ and $b_{11} = \ldots = b_{nn}$. We also know

$$\mathrm{trace}\,(A) = \sum a_{ii} = na_{11} = 0 = nb_{11} = \sum b_{ii} = \mathrm{trace}\,(B)\,,$$

hence $a_{ii} = b_{ii} = 0$. We have thus shown that ϕ is injective.
We already know, that $\dim \mathfrak{sl}_n \oplus \mathfrak{sl}_n = \dim \mathfrak{g}_{\det}$, therefore ϕ is an isomorphism. □

An interesting property of $\mathfrak{sl}_n$ is that if we take a generic element from $A \in \mathfrak{sl}_n$, then A can be diagonalized with high probability, and the transformation matrix is in SL_n.

Lemma 4.9. *For Zariski-almost-all elements $A \in \mathfrak{sl}_n$ we can find $S \in SL_n$, s.t.*

$$SAS^{-1} \textit{ is diagonal}$$

Proof. By lemma 2.11 Zariski-almost-all matrices have distinct eigenvalues and the intersection of the set of diagonal matrices with distinct eigenvalues with $\mathfrak{sl}_n$ is non-empty, consider for example

$$M := \operatorname{diag}\left(1, 2, \ldots, n-1, -\sum_{k=1}^{n-1} k\right) \in \mathfrak{sl}_n$$

So the property transfers to $\mathfrak{sl}_n$ and Zariski-almost all matrices in $\mathfrak{sl}_n$ have distinct eigenvalues.
Now let $A \in \mathfrak{sl}_n$ with distinct eigenvalues. Then A is diagonalizable with some matrix U in GL_n.

$$\operatorname{diag}(d_1, \ldots, d_n) = UAU^{-1} = \underbrace{\left(\frac{1}{\det U} U\right)}_{\in \mathrm{SL}_n} A \underbrace{\left(\det U U^{-1}\right)}_{\in \mathrm{SL}_n}$$

□

Remark 4.10. *The centralizer of a diagonal matrix $A = diag(a_{11}, \ldots, a_{nn})$ with $a_{ii} \neq a_{jj}$ (distinct diagonal elements) is a set of diagonal matrices.*

Proof. Let $X \in \operatorname{Cent}(A)$, then we get

$$(a_{ii} - a_{jj})x_{ij} = 0.$$

Thus $x_{ij} = 0$ for all $i \neq j$ and X is a diagonal matrix. □

Remark 4.11. *The dimension of $\mathfrak{g}_{\det}$ is $(2n^2 - 2)$ and for Zariski-almost-all $A \in \mathfrak{g}_{\det}$ the dimension of its centralizer $Cent(A)$ is $(2n - 2)$.*

Proof. The dimension of $\mathfrak{sl}_n$ is $n^2 - 1$, so by corollary 4.8 the dimension of $\mathfrak{sl}_n \oplus \mathfrak{sl}_n$ is $(n^2 - 1) + (n^2 - 1) = 2n^2 - 2$.
Zariski-almost-all $A \in \mathfrak{sl}_n$ are diagonalizable, so $\operatorname{Cent}(A)$ is conjugate to the centraliser of a diagonal matrix D. The centralizer of a diagonal matrix consists of diagonal matrices (remark 4.10). The additional constraint $\operatorname{trace}(X) = 0$ for all $X \in \mathfrak{sl}_n$ yields $\dim \operatorname{Cent}(A) = \dim \operatorname{Cent}(D) = n - 1$. The bracket operation on $\mathfrak{sl}_n \oplus \mathfrak{sl}_n$ acts componentwise, So for $A, B \in \mathfrak{sl}_n \oplus \mathfrak{sl}_n$ we have $\operatorname{Cent}((A, B)) = \operatorname{Cent}(A) \oplus \operatorname{Cent}(B)$. This yields the claim. □

Lemma 4.12 ([Kay12], Proposition 65)**.** *For Zariski-almost all $A \in \mathfrak{g}_{\det}$ there exists an $S \in \mathcal{G}_{\det}$ s.t.*

$$S^{-1} \, Cent(A) \, S$$

is a set of diagonal matrices.

Proof. Let $A \in \mathfrak{g}_{\det}$ and let Φ and ϕ be defined as in the proof of corollary 4.8:

$$\begin{aligned}\Phi : \mathrm{SL}_n \times \mathrm{SL}_n &\to \mathcal{G}_{\det} \\ (A, B) &\mapsto (X \mapsto AXB^T) \\ \phi : \mathfrak{sl}_n \oplus \mathfrak{sl}_n &\to \mathfrak{g}_{\det} \\ (A, B) &\mapsto (X \mapsto AX + XB^T)\end{aligned}$$

By corollary 4.8 ϕ is an isomorphism so there exist $(A_1, B_1) \in \mathfrak{sl}_n \oplus \mathfrak{sl}_n$ s.t. $\phi(A_1, A_2) = A$. Remark 4.9 tells us that for Zariski-almost-all $A_1, A_2 \in \mathfrak{sl}_n$ we can find $S_1, S_2 \in \mathrm{SL}_n$ s.t. $S_1 A_1 S_1^{-1}, S_2 A_2 S_2^{-1}$ are diagonal matrices. By theorem 3.16 we get

$$\begin{aligned}\Phi(S_1, S_2) A \Phi(S_1, S_2)^{-1} &= \Phi(S_1, S_2)\phi(A_1, A_2)\Phi(S_1, S_2)^{-1} && (4) \\ &= \phi(S_1 A S_1^{-1}, S_2 A_2 S_2^{-1}). && (5)\end{aligned}$$

Next we show that ϕ maps diagonal matrices to diagonal matrices. Let $(C, D) \in \mathfrak{sl}_n \oplus \mathfrak{sl}_n$ be diagonal matrices, then

$$\begin{aligned}\phi(C, D)(X) &= (CX + XD^T) \\ &= \begin{pmatrix} (c_{11} + d_{11})X_{11} & (c_{11} + d_{22})X_{12} & \dots & (c_{11} + d_{nn})X_{1n} \\ (c_{22} + d_{11})X_{21} & (c_{22} + d_{22})X_{22} & \dots & (c_{22} + d_{nn})X_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ (c_{nn} + d_{11})X_{n1} & (c_{nn} + d_{22})X_{n2} & \dots & (c_{nn} + d_{nn})X_{nn} \end{pmatrix} \\ &= \begin{pmatrix} (c_{11} + d_{11}) & 0 & 0 & \dots & 0 \\ 0 & (c_{11} + d_{22}) & 0 & \dots & 0 \\ \vdots & \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & 0 & (c_{nn} + b_{nn}) \end{pmatrix} \cdot \begin{pmatrix} X_{11} \\ X_{12} \\ \vdots \\ X_{nn} \end{pmatrix}\end{aligned}$$

Theorem 4.7 tells us that $\Phi(S_1, S_2) \in \mathcal{G}_{\det}$. Together with (5) this proves the claim. $\square$

5 Algorithmic subprocedures

In this section we will have a look at various subprocedures that will assist us in solving PolyEquiv .
Before we dive into details, a short note about random choices that are made in the subrocedures. We will always assume that random values are uniformly chosen from some finite subset $S \subseteq \mathbb{C}$.
If a polynomial is given by its coefficient list, it is a trivial task to compute its partial derivatives, but we only want to assume that a polynomial is given as a blockbox. The following algorithm can give us a blackbox for any partial derivative of a given polynomial $f \in \mathbb{C}[X_1, \dots, X_n]$.

Algorithm 5.1 Compute partial derivative of a polynomial

```
Input:      Blackbox access to $f \in \mathbb{C}[X_1, \dots, X_n]$ of known degree $d$, $i \in [n]$,
            $a \in \mathbb{C}^n$ point at which to evaluate $\frac{\partial f}{\partial X_i}$
Output:     $\frac{\partial f}{\partial X_i}(a)$
Step (i):   Calculate $\hat{f}(X_i) := f(a_1, \dots, X_i + a_i, \dots, a_n)$ using polynomial in-
            terpolation.
Step (ii):  Return $\frac{\partial \hat{f}}{\partial X_i}(0)$.
```

Proposition 5.1 ([Kay12, Proposition 18]). *Let $f \in \mathbb{C}[X_1, \dots, X_n]$ be an n-variate polynomial of degree d. Given blackbox access to f, algorithm 5.1 gives blackbox access to any derivative $\frac{\partial f}{\partial X_i}$ of f in Poly(dn) time.*

Proof. W.l.o.g. $i = 1$. Let $f \in \mathbb{C}[X_1, \dots, X_n]$. We write f as

$$f = \sum_{j=0}^{d} f_j(X_2, \dots, X_n) X_1^j \in \mathbb{C}[X_2, \dots, X_n][X_1].$$

We define

$$\hat{f}(X_1) = f(a_1 + X_1, a_2, \dots, a_n) = \sum_{j=0}^{d} f_j(a_2, \dots, a_n)(X_1 + a_1)^j.$$

We can interpolate $\hat{f}(X_1)$ by evaluating $\hat{f}$ at $d+1$ points $\{b_1, \dots, b_{d+1}\}$. We get the system of linear equations

$$\sum_{j=0}^{d} f_j(a_2, \dots, a_n)(b_k + a_1)^j = \hat{f}(b_k), \text{ for } k = 1, \dots, d+1$$

and can solve for the $f_j(a_2, \dots, a_n)$.

With those coefficients and we can easily find the derivative of $\hat{f}$, by the standard method:

$$\frac{\partial \hat{f}}{\partial X_1} = \sum_{j=1}^{d} j \cdot f_j(a_2, \dots, a_n)(X_1 + a_1)^{j-1}.$$

As a consequence we can evalute $\frac{\partial f}{\partial X_i}$ at the point a by evaluating $\frac{\partial \hat{f}}{\partial X_i}(0)$:

$$\begin{aligned}\frac{\partial f}{\partial X_1}(a_1, \dots, a_n) &= \sum_{j=1}^{d} j \cdot f_j(a_2, \dots, a_n) a_1^{j-1} \\ &= \sum_{j=1}^{d} j \cdot f_j(a_2, \dots, a_n)(0 + a_1)^{j-1} = \frac{\partial \hat{f}}{\partial X_1}(0).\end{aligned}$$ □

Lie algebras are vector spaces, hence we can use their bases to describe and compare them. What we will learn now is that computing the Lie algebra of a polynomial reduces to finding linear dependencies among a set of polynomials related to its partial derivatives. Remembering that lemma 2.12 yields a basis to the kernel of the linear dependencies we can formulate an algorithm.

Algorithm 5.2 Compute basis for the Lie algebra of a polynomial

```
Input:      $f \in \mathbb{C}[X_1, \dots, X_n]$
Output:     Return $A_1, \dots, A_k$ a basis for $\mathfrak{g}_f$.
Step (i):   Compute the polynomials $f_{ij} := X_j \frac{\partial f}{\partial X_i}$ using algorithm 5.1.
Step (ii):  Let $\{b_1, \dots, b_{n^2}\} \subseteq \mathbb{C}^n$ be a set of distinct random values in $\mathbb{C}^n$.
            Return a basis for the kernel of $M = (f_{ij}(b_i))_{i,j \in [m]}$.
```

Proposition 5.2 ([Kay12, Lemma 22]). *Given an n-variate polynomial $f(X) \in \mathbb{C}[X_1, \dots, X_n]$, algorithm 5.2 computes with high probability a basis for the Lie algebra of its group of symmetries $\mathfrak{g}_f$ in polynomial time.*

Proof. From theorem 4.2 we know, that $A \in \mathfrak{g}_f$ if and only if

$$\sum_{i,j \in [n]} a_{ij} X_j \frac{\partial f}{\partial X_i} = 0 \tag{6}$$

Choose a set $\{b_{11}, \dots, b_{nn}\} \subseteq \mathbb{C}^{n \times n}$ of size n^2 randomly. Define the matrix

$$P(b_{11}, \dots, b_{nn}) = \begin{pmatrix} \left(X_1 \frac{\partial f}{\partial X_1}\right)(b_{11}) & \dots & \left(X_n \frac{\partial f}{\partial X_n}\right)(b_{11}) \\ \vdots & \ddots & \vdots \\ \left(X_1 \frac{\partial f}{\partial X_1}\right)(b_{nn}) & \dots & \left(X_n \frac{\partial f}{\partial X_n}\right)(b_{nn}) \end{pmatrix} \in \mathbb{C}^{n^2 \times n^2}.$$

By corollary 2.13 the basis of $P(b_{11}, \dots, b_{nn})$ yields a basis for the solution space of (6) for Zariski-almost-all b_{ij}. Lemma 4.2 tell us that a basis of (6) is also a basis for the Lie algebra $\mathfrak{g}_f$. □

The centralizer of an element in a Lie algebra is a vector space. We can directly derive a system of linear equations from its definition. Solving the system yields a basis.

Algorithm 5.3 Compute basis of centralizer

```
Input:     $A = (a_{ij})_{1\leq i,j\leq n} \in \mathbb{C}^{n\times n}$
Output:    Return $X_1, \dots, X_m$, s.t. the $X_i$ are a basis for $\mathrm{Cent}\,(A)$.
STEP (I):  Compute a basis for the kernel of $AX - XA$, i.e. solve the system
           of $n^2$ linear equations given by
               $\sum_{l=1}^{n} a_{il}X_{li} - \sum_{k=1}^{n} a_{kj}X_{jk} = 0, \forall i,j \in [n]$    (7)
           using linear algebra. Return basis $X_1, \dots, X_k$.
```

Proposition 5.3 ([Kay12, Fact 24])**.** *Let $\mathfrak{g} \subseteq \mathbb{C}^{n\times n}$ be a Lie algebra, with the bracket operation defined as $[A,B] = AB - BA$. Let $A \in \mathfrak{g}$, then algorithm 5.3 computes Cent(A) in polynomial time.*

Proof. Let $A = (a_{ij}) \in \mathfrak{g}$ and $B = (b_{ij})_{i,j\in[n]} \in \mathrm{Cent}\,(A)$. By definition 3.5 this is equivalent to

$$AB - BA = 0$$

Simple expansion of the matrix multiplication and the difference yields

$$[A,B]_{ij} = \sum_{l=1}^{n} a_{il}b_{li} - \sum_{k=1}^{n} a_{kj}b_{jk} = 0, \forall i,j \in [n]$$

as equivalent condition to $B \in \mathrm{Cent}\,(A)$. Hence every $B \in \mathrm{Cent}\,(A)$ fulfills (7). The converse is also true, if $X \in \mathbb{C}^{n\times n}$ fulfills the system defined by (7), then $[A,X] = AX - XA = 0$ and therefore $X \in \mathrm{Cent}\,(A)$.
So a basis of the solution space for (7) is a basis for $\mathrm{Cent}\,(A)$, and can be computed in polynomial time by using Gaussian elimination. □

Another subroutine that Kayal uses in [Kay12] is related to matrix diagonalization. As Neeraj Kayal puts it in [Kay12, p. 24]:

> The second step of this algorithm, viz. simultaneous diagonalization of a set of (commuting) matrices is a standard linear algebra computation and can easily be accomplished in poly(n) time.

Diagonalization is in fact a subroutine, commonly used in numerical computations, but we are interested in an exact (algebraic) algorithm. To the author no such algorithm is known, but as it turns out, there exists an algorithm that can compute the eigenvalues and eigenspaces of a matrix up to a relative error bound of 2^{-b} in $O(n^3 + (n \log^2 n) \log b)$

arithmetic steps ([PC99]). Unfortunately this does not solve the problem completely, so we will simply assume the existence of a hypothetical polynomial time diagonalization algorithm.

Remark 5.4. *We assume the existence of an algorithm that can diagonalize a matrix in polynomial time.*

6 The Algorithm

Let us quickly recall what we are trying to achieve. We are given a polynomial $f \in \mathbb{C}[X_{11}, \ldots, X_{nn}]$ and want to find an invertible matrix $A \in \mathrm{GL}_{n^2}$ s.t. $f(X) = \det(A \cdot X)$, if such a matrix exists.
As a result of this, the Lie algebras $\mathfrak{g}_f$ and $\mathfrak{g}_{\det}$ are conjugate via A:

$$\mathfrak{g}_f = A^{-1}\mathfrak{g}_{\det} A.$$

As stated in previous chapters the main idea of the algorithm is to find the conjugation matrix A of $\mathfrak{g}_{\det}$ and $\mathfrak{g}_f$. We will find such a matrix with three seperate computation steps. There will be a fourth step that checks if the matrix A that we might have found indeed gives us the polynomial f from det.
In each step of the algorithm we will reduce the set of possible transformation matrices from a subgroup $G \leq \mathrm{GL}_n$ to a proper subgroup $H \leq G$.
For this we recall the definitions of PS_n and SC_n, the set of matrices that permute and scale and the set of scaling matrices respectively.
In our first step we will reduce an equivalence relation in GL_n, i.e. $A \in \mathrm{GL}_n$ to an equivalence relation in PS_n, i.e. $A \in \mathrm{PS}_n$. The algorithm works as follows:

Algorithm 6.4 Reduce GL_n-equivalence to PS_n-equivalence

```
Input:      $f \in \mathbb{C}[X_{11}, \ldots, X_{nn}]$
Output:     If $f$ is $\mathrm{GL}_n$-equivalent to $\det_n$, we return a matrix $D \in \mathrm{GL}_n$, s.t.
            $f(DX)$ is $\mathrm{PS}_n$-equivalent to $\det_n$.
Step (i):   Compute a basis $A_1, \ldots, A_k \in \mathfrak{gl}_n$ of $\mathfrak{g}_f$ using algorithm 5.2. If
            $k \neq (2n^2 - 2)$, return 'not equivalent to determinant'.
  Step (ii):   Pick $B \in \mathfrak{g}_f$ randomly and compute a basis $X_1, \ldots, X_k$ of Cent $(B)$
               using algorithm 5.3. If $k \neq (2n - 2)$ return 'not equivalent to
               determinant'.
  Step (iii):  Compute a matrix $D \in \mathrm{GL}_n$ that simultaneously diagonalizes
               $X_1, \ldots, X_{2n-2}$ using the hypothetical algorithm from remark 5.4.
               If $D$ doesn't exist, return 'not equivalent to determinant'.
```

Proposition 6.1 ([Kay12, Proposition 45]). *Let $f \in K[X_{11}, \ldots, X_{nn}]$ be a polynomial. Let $D \in \mathbb{C}^{n^2 \times n^2}$ be the output of algorithm 6.4. If f is GL_n-equivalent to* $\det_n$*, then with high probability $f(DX)$ is PS_n-equivalent to* $\det_n$.

Proof. Let f be GL_n-equivalent to $\det_n$, by theorem 4.5 then there exists a matrix $A \in \mathrm{GL}_{n^2}$ s.t.

$$\mathfrak{g}_f = A^{-1}\mathfrak{g}_{\det} A$$

Proposition 5.2 ensures that (I) can be accomplished by algorithm 5.2. By lemma 4.11 the dimension of $\mathfrak{g}_{\det}$ is $(2n^2 - 2)$, so let $A_1, \ldots, A_{n^2-2}$ be a basis for $\mathfrak{g}_f$.

Let $B \in \mathfrak{g}_f$ a random element. The correctness of step (II) follows from proposition 5.3 and remark 4.11. Let $X_1, \ldots, X_{2n-2}$ be the basis of $\mathrm{Cent}\,(B)$. With high probability we can find a matrix $D \in \mathrm{GL}_{n^2}$ that diagonalizes B, i.e. DBD^{-1} is diagonal and its diagonal entries are distinct. By remark 4.10 $\mathrm{Cent}\,(DBD^{-1})$ is a set of diagonal matrices. For each $i \in [2n-2]$ we have

$$\begin{aligned}[DBD^{-1}, DXD^{-1}] &= DBD^{-1}DX_iD^{-1} - DX_iD^{-1}DBD^{-1} \\ &= D(BX_i - X_iB)D^{-1} = 0\end{aligned}$$

Thus $DX_iD^{-1} \in \mathrm{Cent}\,(DBD^{-1})$ is diagonal for all $i \in [2n-2]$.
As $\mathfrak{g}_f$ and $\mathfrak{g}_{\det}$ are conjugate via A there exists a matrix $C \in \mathfrak{g}_{\det}$, s.t. $B = A^{-1}CA$, i.e. Corollary 4.6 also gives us that the centralisers of B and C are conjugate, hence

$$\begin{aligned}D^{-1}\mathrm{Cent}\,(B)\,D &= D^{-1}A^{-1}\mathrm{Cent}\,(C)\,AD &(8)\\ &= (AD)^{-1}\mathrm{Cent}\,(C)\,(AD). &(9)\end{aligned}$$

What we conclude from this is that AD diagonalizes $\mathrm{Cent}\,(C)$.
By lemma 4.12 there exists a matrix $S \in \mathcal{G}_{\det_n}$, that diagonalizes the centraliser of C:

$$S^{-1}\mathrm{Cent}\,(C)\,S \text{ is a set of diagonal matrices.}$$

We define

$$Z := S^{-1}AD$$

We now want to show that $Z \in \mathrm{PS}_n$. Take a look at

$$\begin{aligned}Z(D^{-1}\mathrm{Cent}\,(B)\,D)Z^{-1} &= (S^{-1}AD)(D^{-1}\mathrm{Cent}\,(B)\,D)(S^{-1}AD)^{-1} \\ &\overset{(9)}{=} (S^{-1}AD)((AD)^{-1}\mathrm{Cent}\,(C)\,(AD))((AD)^{-1}S) \\ &= S^{-1}\mathrm{Cent}\,(C)\,S\end{aligned}$$

Hence Z^{-1} diagonalizes $D^{-1}\mathrm{Cent}\,(B)\,D$, which is already a set of diagonal matrices. As a consequence Z can only be a permutation or scaling of the matrices in $D^{-1}\mathrm{Cent}\,(B)\,D$. Thus $Z \in \mathrm{PS}_n$. The fact that $S \in \mathcal{G}_{\det_n}$ yields:

$$\begin{aligned}f(DX) &= \det(ADX) \\ &= \det(SZX) \\ &= \det(S(ZX)) \\ &= \det(ZX)\end{aligned}$$

In particular $f(DX)$ is PS_n-equivalent to $\det_n$. □

Now that we have reduced general GL_n-equivalence to PS_n-equivalence we now want to get rid of the permutation part (the discrete part) of PS_n. What will assist us with this are the partial derivatives of the polynomial f. We make the following observation about the partial derivatives of det: second order partial derivatives of det on the same row or column vanish.

Lemma 6.2.

$$\frac{\partial^2 \det}{\partial X_{ij}\partial X_{kl}} \begin{cases} = 0 & \textit{if } i = k \vee j = l \\ \neq 0 & \textit{else} \end{cases}$$

Proof. Let $i, j, k, l \in [n]$. We take a look at the Leibniz formula for the determinant and we note that:

$$\begin{aligned}\frac{\partial \det}{\partial X_{ij}} &= \frac{\partial}{\partial X_{ij}} \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{m=1}^{n} X_{m\sigma(m)} \\ &= \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \underbrace{\frac{\partial}{\partial X_{ij}} \prod_{m=1}^{n} X_{m\sigma(m)}}_{\neq 0 \iff \sigma(i)=j} \\ &= \sum_{\sigma \in S_n : \sigma(i)=j} \operatorname{sgn}(\sigma) \prod_{m \in [n]: m \neq i}\end{aligned}$$

We consider two cases:
1. Case: $(i, j) = (k, l)$

$$\begin{aligned}\frac{\partial \det}{\partial X_{ij}\partial X_{ij}} &= \frac{\partial}{\partial X_{ij}} \sum_{\sigma \in S_n : \sigma(i)=j} \operatorname{sgn}(\sigma) \prod_{m \in [n]: m \neq i} X_{m\sigma(m)} \\ &= \sum_{\sigma \in S_n : \sigma(i)=j} \operatorname{sgn}(\sigma) \underbrace{\frac{\partial}{\partial X_{ij}} \prod_{m \in [n]: m \neq i} X_{m\sigma(m)}}_{=0} = 0\end{aligned}$$

2. Case: $(i, j) \neq (k, l)$

$$\begin{aligned}\frac{\partial \det}{\partial X_{ij}\partial X_{kl}} &= \frac{\partial}{\partial X_{kl}} \sum_{\sigma \in S_n : \sigma(i)=j} \operatorname{sgn}(\sigma) \prod_{m \in [n]: m \neq i} X_{m\sigma(m)} \\ &= \sum_{\sigma \in S_n : \sigma(i)=j} \operatorname{sgn}(\sigma) \underbrace{\frac{\partial}{\partial X_{kl}} \prod_{m \in [n]: m \neq i} X_{m\sigma(m)}}_{=0 \iff \sigma(k)=l} \\ &= \sum_{\sigma \in S_n : \sigma(i)=j, \sigma(k)=l} \operatorname{sgn}(\sigma) \prod_{m \in [n]: m \neq i, m \neq k} X_{m\sigma(m)}\end{aligned}$$

The set $\{\sigma \in S_n \mid \sigma(i) = j, \sigma(k) = l\}$ is empty if and only if either $i = k$ or $j = l$, because permutations are bijections. □

The lemma carries over to the second order partial derivatives of the projection f s.t. if the derivative $\frac{\partial^2 f}{\partial X_{ij}\partial X_{kl}}$ vanishes we know that X_{ij} and X_{kl} are in the same row or column. By identifying the variables of the first row and column in the input matrix, we can then assign the rest of the variables to their appropriate position.

Algorithm 6.5 Reducing PS_n-equivalence to SC_n-equivalence

```
Input:      $f \in \mathbb{F}[X_{11}, \dots, X_{nn}]$
Output:     If $f$ is $\mathrm{PS}_n$-equivalent to $\det_n$, we return a permutation $\sigma \in S_{[n]\times[n]}$,
            s.t. $f(\sigma(X))$ is $\mathrm{SC}_n$-equivalent to $\det_n$
STEP (I):   Set $\pi(1,1) = (1,1)$.
  STEP (II):  Find set $A \subseteq ([n] \times [n] \setminus \{(1,1)\})$ of size $(2n-2)$, s.t.
                  $\frac{\partial^2 f}{\partial X_{11}\partial X_{ij}} = 0$
              Calculate partial derivatives with algorithm 5.1. If $A$ does not exist
              return 'not equivalent to determinant'.
  STEP (III): Partition $A$ into two sets $R$ and $C$ of size $(n-1)$, s.t.
                  $\frac{\partial^2 f}{\partial X_{ij}\partial X_{kl}} \begin{cases} = 0 \text{ if } (i,j) \in R \text{ and } (k,l) \in R, \\ = 0 \text{ if } (i,j) \in C \text{ and } (k,l) \in C, \\ \neq 0 \text{ if } (i,j) \in R \text{ and } (k,l) \in C \end{cases}$   (10)
              by calculating the partial derivatives with algorithm 5.1 If no such
              partition exists, return 'not equivalent to determinant'.
  STEP (IV):  Let $R = \{(i_1, j_1), \dots, (i_{n-1}, j_{n-1})\}$ and let $C =$
              $\{(k_1, l_1), \dots, (k_{n-1}, l_{n-1})\}$. For all $m \in [n-1]$ set
                  $\pi(1, m+1) = (k_m, l_m)$
                  $\pi(m+1, 1) = (i_m, j_m)$
  STEP (V):   For each $(i,j) \in ([n] \times [n]) \setminus (\{(1,1)\} \cup R \cup C)$ find a unique pair
              $(i_r, j_r) \in R$ and $(k_s, l_s) \in C$, s.t.
                  $\frac{\partial^2 f}{\partial X_{ij}\partial X_{k_r l_r}} = \frac{\partial^2 f}{\partial X_{ij}\partial X_{i_r j_r}} = 0$
              If the pair is not unique, return 'not equivalent to
              determinant'. Set
                  $\pi(r,s) = (i,j)$
              Return $\sigma := \pi^{-1}$.
```

Proposition 6.3 ([Kay12, Proposition 41]). *Let $f \in \mathbb{C}[X_{11}, \ldots, X_{nn}]$ be PS$_n$-equivalent to* det*, and let $\pi \in [n] \times [n]$ be the output of algorithm 6.5 for f. Then $f(\pi(X))$ is SC$_n$-equivalent to* det.

Proof. The basic idea is to group variables by their row and column. Lemma 6.2 justifies this approach. Let $f \in \mathbb{C}[X_{11}, \ldots, X_{nn}]$ be PS$_n$-equivalent to det via $f(X_{11}, \ldots, X_{nn}) = \det(\lambda_{11}X_{\sigma(1,1)}, \ldots, \lambda_{nn}X_{\sigma(n,n)})$. By lemma 6.2, the set

$$A := \left\{ (i,j) \in [n] \times [n] \mid (i,j) \neq (1,1) \wedge \frac{\partial^2 f}{\partial x_{11} x_{ij}} = 0 \right\}$$

consists of the indices (i,j) s.t. $\sigma(i,j)$ is in the same row or in the same column as $\sigma(1,1)$.
Permutation of rows and columns only change the determinant up to a sign, so we may permute rows and columns in such a way that $\sigma(1,1) = (1,1)$. As a consequence we get that, if we take any $(i,j) \in A$ its image $\sigma(i,j)$ is either in the first row or in the first column of the matrix X.
We can thus partition A into indices that correspond to the first row and to the first column. We can find the partition A into a set R and a set C that correspond to the first row and first column respectively, by checking second order partial derivatives. Let $(i,j), (k,l) \in A$, then

$$\frac{\partial^2 f}{\partial x_{ij} x_{kl}} = 0$$

if and only if $\sigma(i,j)$ is either in the same row or in the same column as $\sigma(k,l)$. From this we can easily construct R and C. A semantic of description of R and C:

$$R = \{(i,j) \in A \mid (i,j) \neq (1,1) \wedge \sigma(i,j) = (1,k) \text{ for some } k\}$$
$$C = \{(i,j) \in A \mid (i,j) \neq (1,1) \wedge \sigma(i,j) = (k,1) \text{ for some } k\}$$

Also the determinant does not change under transposition of the input matrix, so it does not matter which of the sets corresonds to the first row or column. This justifies the assignments in step (IV).
For every other index $(i,j) \in [n] \times [n] \setminus A$ we can then determine the row and column of $\sigma(i,j)$ by finding the unique indices index $(i_R, j_R) \in R$ and (i_C, j_C) with

$$\frac{\partial^2 f}{\partial x_{ij} \partial x_{i_R j_R}} = \frac{\partial^2 f}{\partial x_{ij} \partial x_{i_C j_C}} = 0.$$

The association works in the following way:
If $\sigma(i_R, j_R) = (1,l)$ and $\sigma(i_C, j_C) = (k,1)$, we deduce $\sigma(i,j) = (k,l)$. □

What now remains of our proble is an equivalence transformation of the form

$$f \begin{pmatrix} X_{11} & \ldots & X_{1n} \\ \vdots & \ddots & \vdots \\ X_{n1} & \ldots & X_{nn} \end{pmatrix} = \det \begin{pmatrix} \lambda_{11}X_{11} & \ldots & \lambda_{1n}X_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{n1}X_{n1} & \ldots & \lambda_{nn}X_{nn} \end{pmatrix}$$

Thus evaluating f at a permutation matrix yields a product of the λ_{ij}, namely $f(P_\sigma) = \prod_{i=1}^n \lambda_{i\sigma(i)}$, where P_σ is the permutation matrix defined as

$$P_\sigma = (p_{ij})_{1\leq i,j\leq n} = \begin{cases} 1 \text{ if } \sigma(i) = j, \\ 0 \text{ else} \end{cases}$$

By choosing the permutations in a smart way we can solve for the λ_{ij}.

Algorithm 6.6 Resolve SC$_n$-equivalence

```
Input:       $f \in \mathbb{C}[X_{11}, \dots, X_{nn}]$
Output:      If $f$ is SC$_n$-equivalent to $\det_n$, return matrix $B \in$ SC$_n$, s.t. $f(X) = \det_n(BX)$.
STEP (I):    Set $\lambda_{11} = \lambda_{12} = \dots = \lambda_{1n} = \lambda_{21} = \dots = \lambda_{(n-1)1} = 1$
STEP (II):   For $i \in [n-1]$ and $j \in [n]$, choose $k \neq 1, i, j$ and define
                 $\sigma(x) = \begin{cases} j \text{ if } x = i, \\ k \text{ if } x = j, \\ i \text{ if } x = k, \\ x \text{ else} \end{cases}$ ,   $\pi(x) = \begin{cases} j \text{ if } x = 1, \\ k \text{ if } x = j, \\ i \text{ if } x = k, \\ 1 \text{ if } x = i, \\ x \text{ else} \end{cases}$
             Then set $\lambda_{ij} = -\frac{f(P_\sigma)}{f(P_\pi)}$
STEP (III):  For $j \in [n-1]$ define $c := \lambda_{jn} \prod_{i\in[n-1]:i\neq j} \lambda_{ii}$ and define
                 $\tau(x) = \begin{cases} n \text{ if } x = j, \\ j \text{ if } x = n \end{cases}$
             Set $\lambda_{nj} = -\frac{f(P_\tau)}{c}$.
STEP (IV):   Set $\lambda_{nn} = \frac{f(I_n)}{\prod_{i<n} \lambda_{ii}}$ and return $B := \operatorname{diag}(\lambda_{11}, \dots, \lambda_{nn})$.
```

Proposition 6.4 ([Kay12, Proposition 42]). *Let $f \in \mathbb{C}[X_{11}, \dots, X_{nn}]$ be a polynomial that is SC$_n$-equivalent to the determinant, i.e. $\exists \lambda_{11}, \dots, \lambda_{nn} \in \mathbb{C}$, s.t.*

$$f(X_{11}, \dots, X_{nn}) = \det(\lambda_{11}X_{11}, \dots, \lambda_{nn}X_{nn})$$

Then algorithm 6.6 computes the λ_{ij} in polynomial time.

Proof. Let $f \in \mathbb{C}[X_{11}, \dots, X_{nn}]$ be SC$_n$-equivalent to det, i.e. $f(X_{11}, \dots, X_{nn}) =$

$\det(\lambda_{11}X_{11}, \ldots, \lambda_{nn}X_{nn})$. Define $\Lambda := \operatorname{diag}(\lambda_{11}, \ldots, \lambda_{nn})$ and the matrices

$$U := \begin{pmatrix} \prod_{j=2}^n \lambda_{1j}^{-1}\lambda_{11}^{-1} & & & \\ & \ddots & & \\ & & \prod_{j=2}^n \lambda_{1j}^{-1}\lambda_{(n-1)1}^{-1} & \\ & & & \left(\prod_{j=2}^n \lambda_{1j}\right)^{n-1} \prod_{i=1}^{n-1} \lambda_{i1} \end{pmatrix} \in \mathbb{C}^{n\times n}$$

$$V := \begin{pmatrix} \prod_{j=2}^n \lambda_{1j} & & & \\ & \lambda_{12}^{-1} & & \\ & & \ddots & \\ & & & \lambda_{1n}^{-1} \end{pmatrix} \in \mathbb{C}^{n\times n}$$

Then we get

$$\det(U) = \left(\prod_{j=2}^n \lambda_{1j}^{-1}\right)^{n-1} \cdot \left(\prod_{i=1}^{n-1} \lambda_{i1}^{-1}\right) \cdot \left(\left(\prod_{j=2}^n \lambda_{1j}\right)^{n-1} \prod_{i=1}^{n-1} \lambda_{i1}\right) = 1$$

$$\det(V) = \left(\prod_{j=2}^n \lambda_{1j}\right) \cdot \lambda_{12}^{-1} \cdot \ldots \cdot \lambda_{1n}^{-1} = 1$$

Therefore $U, V \in \mathrm{SL}_n$. We conclude

$$\begin{aligned} f\begin{pmatrix} X_{11} & \ldots & X_{1n} \\ \vdots & \ddots & \vdots \\ X_{n1} & \ldots & X_{nn} \end{pmatrix} &= \det \begin{pmatrix} \lambda_{11}X_{11} & \ldots & \lambda_{1n}X_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{n1}X_{n1} & \ldots & \lambda_{nn}X_{nn} \end{pmatrix} \\ &= \det U \begin{pmatrix} \lambda_{11}X_{11} & \ldots & \lambda_{1n}X_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{n1}X_{n1} & \ldots & \lambda_{nn}X_{nn} \end{pmatrix} V \\ &= \det U \begin{pmatrix} \prod_{j=2}^n \lambda_{1j}\lambda_{11}X_{11} & X_{12} & \ldots & X_{1n} \\ \prod_{j=2}^n \lambda_{1j}\lambda_{21}X_{21} & * & \ldots & * \\ \vdots & \vdots & \ddots & \vdots \\ \prod_{j=2}^n \lambda_{1j}\lambda_{n1}X_{n1} & * & \ldots & * \end{pmatrix} \\ &= \det \begin{pmatrix} X_{11} & \ldots & & X_{1n} \\ X_{21} & * & \ldots & * \\ \vdots & \vdots & \ddots & \vdots \\ X_{(n-1)1} & * & \ldots & * \\ *X_{n1} & * & \ldots & * \end{pmatrix} \end{aligned}$$

Therefore we can assume $\lambda_{11} = \ldots = \lambda_{1n} = \lambda_{21} = \ldots = \lambda_{(n-1)1} = 1$. This assumption is crucial, as it allows us to find permuations σ, π, s.t. if we evaluate at their corresponding permuation matrices P_σ, P_π we can divide their results and get one of the λ_{ij}.

Let $i \in [n-1], j \in [n]$ and $\sigma, \pi \in S_{n\times n}$ as defined in step (II), then we get

$$\begin{aligned} f(P_\sigma) &= \det(\Lambda.P_\sigma) = \operatorname{sgn}(\sigma) \prod_{m=1}^{n} \lambda_{m\sigma(m)} \\ &= \lambda_{ij}\lambda_{jk}\lambda_{ki} \prod_{m\neq i,j,k} \lambda_{mm} \underbrace{=}_{\lambda_{11}=1} \lambda_{ij}\lambda_{jk}\lambda_{ki} \prod_{m\neq 1,i,j,k} \lambda_{mm} \\ f(P_\pi) &= \det(\Lambda.P_\pi) = \operatorname{sgn}(\pi) \prod_{m=1}^{n} \lambda_{m\pi(m)} \\ &= \operatorname{sgn}(\pi)\, \lambda_{1j}\lambda_{jk}\lambda_{ki}\lambda_{i1} \prod_{m\neq 1,i,j,k} \lambda_{mm} \underbrace{=}_{\lambda_{1j}=\lambda_{i1}=1} -\operatorname{sgn}(\sigma)\, \lambda_{jk}\lambda_{ki} \prod_{m\neq 1,i,j,k} \lambda_{mm} \end{aligned}$$

We conclude $\lambda_{ij} = -\frac{f(P_\sigma)}{f(P_\pi)}$.
Let $i = n$ and $j \in [n-1]$. Define $c = \lambda_{jn} \prod_{m\neq j} \lambda_{mm}$. and τ as in step (III). Then

$$f(P_\tau) = -\lambda_{jn}\lambda_{nj} \prod_{m\neq j,n} \lambda_{mm}$$

and therefore $\lambda_{nj} = -\frac{f(P_\tau)}{c}$.
We know that $f(I_n) = \prod_{i=1}^{n} \lambda_{ii}$, therefore $\lambda_{nn} = \frac{f(I_n)}{\prod_{i<n} \lambda_{ii}}$ as in step (IV).
This yields that algorithm 6.6 works correct. The number of evaluations of f at permutation matrices is $O(n^2)$, thus we get polynomial running time. $\square$

The last step is to combine all the previous algorithms. We have seen, that we can reduce GL_n equivalence to PS_n equivalence, then to SC_n equivalence and finally find all scaling factors. There might exist polynomials that are not equivalent to det but their Lie algebra $\mathfrak{g}_f$ is still conjugate to $\mathfrak{g}_{\det}$. To find such errors in the computation, we check if the matrix A that we get does indeed yield equivalence to det via the Schwartz-Zippel-Lemma.

Algorithm 6.7 Resolve GL-equivalence for det

```
Input:        $f \in \mathbb{C}[X_{11}, \dots, X_{nn}]$
Output:       If $f$ is equivalent to $\det_n$, then return a matrix $A \in \mathbb{C}^{n^2\times n^2}$, s.t.
              $f(X) = \det_n(AX)$
STEP (I):     Calculate a matrix $D \in \mathbb{C}^{n^2\times n^2}$ s.t. $f(DX) = \det_n(X)$ using algo
              6.4. Define $f_2(X) := f(DX)$.
  STEP (II):    Calculate a matrix $C \in \mathbb{C}^{n^2\times n^2}$ s.t. $f_2(CX) = \det_n(X)$ using algo
                6.5. Define $f_3(X) := f(CX)$.
  STEP (III):   Calculate a matrix $B \in \mathbb{C}^{n^2\times n^2}$ s.t. $f_3(X) = \det_n(BX)$ using algo
                6.6.
  STEP (IV):    Define $A := BC^{-1}D^{-1}$. Use Schwartz Zippel Lemma to check if
                $f(X) = \det_n(AX)$. If this is the case, return $A$, else return 'not
                equivalent to determinant'.
```

Theorem 6.5. *With high probability algorithm 6.7 solves the equivalence problem for the determinant in polynomial time.*

Proof. Let $f \in \mathbb{C}[X_1, \dots, X_n]$ s.t. f is equivalent to the determinant, i.e. $f(X) = \det(AX)$ for some $A \in \mathrm{GL}_n$.
Let D be the matrix calulated in step (I), by proposition 6.1, $f_2(X) = f(DX)$ is PS_n-equivalent to det.
Hence we can continue to compute the matrix C in step (II). $f_3(X) = f_2(CX)$ is SC_n-equivalent to det (Proposition 6.3).
Proposition 6.4 states that the matrix B computed in step (III) fullfills

$$\det(BX) = f_3(X) = f_2(CX) = f(DCX).$$

The matrix $A = BC^{-1}D^{-1}$ yields

$$\begin{aligned}\det(AX) &= \det(BC^{-1}D^{-1}X) = f_3(C^{-1}D^{-1}X) = f_2(CC^{-1}D^{-1}X)\\ &= f_2(D^{-1}X) = f(DD^{-1}X) = f(X).\end{aligned}$$

Therefore the algorithm yields a correct solution. □

The algorithm uses random choices in several places, for calculating a basis for the Lie algebra of the input polynomial f, the choice of a random matrix $B \in \mathfrak{g}_f$ and the final identity test. The probability of an error in any of these steps can be bounded by an application of the Schwartz-Zippel-Lemma (corollary 2.8).
Formally the algorithm is a Monte Carlo algorithm, even though the final step yields a correctness test, it is still a probabilistic test.
If polynomial identity testing is derandomized at some point, the algorithm can be seen as a Las Vegas algorithm.
What remains as a problem is the unsolved existence of a matrix diagonalization algorithm, without diagonalization, the algorithm can not solve the PolyEquiv problem.

Appendix

Deutsche Zusammenfassung

Die algebraische Komplexitätstheorie beschäftigt sich mit der Komplexität von Problemen in einem algebraischen/arithmetischen Kontext, wie beispielsweise der möglichst effizienten Berechnung von Polynomen. Komplexität in diesem Kontext bezieht sich meistens auf die Anzahl an arithmetischen Operation ($+$, $-$, $\cdot$, $/$).
Ähnlich den Reduktionen in der (Standard-)Komplexitätstheorie untersucht man in der algebraischen Komplexitätstheorie sogenannte p-Projektionen (siehe [BCS97] oder [Bür00]). In dieser Arbeit behandle ich sogenannte Polynomäquivalenzen, dies sind lineare Abbildung (gegeben durch eine invertiebare Matrix $A \in \mathrm{GL}_n$). Zwei Polynome $f, g \in \mathbb{C}[X_1, \ldots, X_n]$ heißen äquivalent falls eine Matrix $A \in \mathrm{GL}_n$ existiert, sodass

$$f(X) = g(A \cdot X)$$

ist.
In dieser Arbeit beschäftige ich mich mit dem Äquivalenzproblem für das Determinantenpolynom.
Die Grundlage für diese Arbeit ist das Paper „Affine projections of polynomials“ ([Kay12]) von Neeraj Kayal. Unter der Annahme, dass Matrixdiagonalisierung mit polynomiellem Zeitaufwand bzgl. der Dimension der Eingabematrix möglich ist, konnte Neeraj Kayal in seiner Arbeit einen randomisierten Algorithmus entwickeln, der das Äquivalenzproblem für die Determinante in polynomieller Zeit löst.
In dieser Arbeit werden die Ergebnisse und der Algorithmus von Neeraj Kayal präsentiert und anhand der Lie-Theorie von Grund auf hergeleitet. Abschnitt 1 enthält eine erweiterte Einführung in das Problem und eine Zusammenfassung des Algorithmus.
Im zweiten Abschnitt werden fortgeschrittene Konzepte über Polynome entwickelt und der dritte Abschnitt dient als Einführung für Matrix-Lie-Gruppen und Lie-Algebren.
Im Abschnitt 4 werden verschiedene Zwischenergebnisse von Kayal auf Basis der Matrix-Lie-Gruppen und deren Lie-Algebren entwickelt. Diese Zwischenergebnisse dienen als Grundlage um die Korrektheit des Algorithmus herzuleiten.
Abschnitt 5 enthält eine Zusammenfassung der verwendenten Unterprozeduren und deren Korrektheitsbeweise.
Im sechsten Abschnitt wird der komplette Algorithmus in drei Teilschritten präsentiert und deren Korrektheit bewiesen.

References

[BC13] Peter Bürgisser and Felipe Cucker. *Condition. The geometry of numerical algorithms.* Berlin: Springer, 2013.

[BCS97] Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi. *Algebraic complexity theory. With the collaboration of Thomas Lickteig.* Berlin: Springer, 1997.

[Bos13] Siegfried Bosch. *Algebra.* Berlin: Springer, 8th corrected ed. edition, 2013.

[Bro89] Markus Brodmann. *Algebraische Geometrie. Eine Einführung. (Algebraic geometry. An introduction).* Basel etc.: Birkhäuser Verlag, 1989.

[Bür00] Peter Bürgisser. *Completeness and reduction in algebraic complexity theory.* Berlin: Springer, 2000.

[Gro12a] Joshua A. Grochow. Matrix Lie algebra isomorphism. In *IEEE Conference on Computational Complexity (CCC12)*, pages 203–213, 2012. Also available as arXiv:1112.2012 [cs.CC] and ECCC Technical Report TR11-168.

[Gro12b] Joshua A. Grochow. *Symmetry and equivalence relations in classical and geometric complexity theory.* PhD thesis, University of Chicago, Chicago, IL, 2012.

[Hal03] Brian C. Hall. *Lie groups, Lie algebras, and representations. An elementary introduction.* New York, NY: Springer, 2003.

[Kay11] Neeraj Kayal. Efficient algorithms for some special cases of the polynomial equivalence problem. In *Symposium on Discrete Algorithms (SODA).* Society for Industrial and Applied Mathematics, 2011.

[Kay12] Neeraj Kayal. Affine projections of polynomials. In *Symposium on Theory of Computing (STOC).* ACM, 2012.

[MM59] Marvin Marcus and B.N. Moyls. Linear transformations on algebras of matrices. *Can. J. Math.*, 11:61–66, 1959.

[PC99] Victor Y. Pan and Zhao Q. Chen. The complexity of the matrix eigenproblem. In Jeffrey Scott Vitter, Lawrence L. Larmore, and Frank Thomson Leighton, editors, *Proceedings of the 31st annual ACM symposium on theory of computing, STOC 1999. Atlanta, GA, USA, May 1–4, 1999.*, pages 507–516. New York, NY: ACM, Association for Computing Machinery, 1999.

[Rei16] Philipp Reichenbach. Stabilistator der Determinante und maximal lineare Teilräume. Bachelorarbeit, Technische Universität Berlin, 2016.

[Sax09] Nitin Saxena. Progress on polynomial identity testing. *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS*, 99:49–79, 2009.

[vG13] Joachim von zur Gathen and Jürgen Gerhard. *Modern computer algebra. 3rd ed.* Cambridge: Cambridge University Press, 3rd ed. edition, 2013.