Lower bounds and reconstruction algorithms for sums of affine powers

Timothée Pecatte

N° d'ordre NNT : 2018LYSEN029

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON
opérée par
l'École Normale Supérieure de Lyon

École Doctorale N°512
École Doctorale en Informatique et Mathématiques de Lyon

Discipline : Informatique

Soutenue publiquement le 11/07/2018, par :
Timothée PECATTE

Bornes inférieures et algorithmes de reconstruction pour des sommes de puissances affines

---

Devant le jury composé de :

Alin BOSTAN, INRIA Saclay Île-de-France	Rapporteur
Markus BLÄSER, Universität des Saarlandes	Rapporteur
Evelyne HUBERT, INRIA Méditerranée	Examinatrice
Claire MATHIEU, École normale supérieure	Examinatrice
Bruno SALVY, École normale supérieure de Lyon	Examinateur
Pascal KOIRAN, École normale supérieure de Lyon	Directeur de thèse

Titre : Bornes inférieures et algorithmes de reconstruction pour des sommes de puissances affines

Résumé : Le cadre général de cette thèse est l'étude des polynômes comme objets de modèles de calcul. Cette approche permet de définir de manière précise la complexité d'évaluation d'un polynôme, puis de classifier des familles de polynômes en fonction de leur difficulté dans ce modèle. Dans cette thèse, nous nous intéressons en particulier au modèle *AffPow* des sommes de puissance de forme linéaire, i.e. les polynômes qui s'écrivent $f = \sum_{i=1}^{s} \alpha_i \ell_i^{e_i}$, avec $\deg \ell_i = 1$. Ce modèle semble assez naturel car il étend à la fois le modèle de Waring $f = \sum \alpha_i \ell_i^d$ et le modèle du décalage creux $f = \sum \alpha_i \ell^{e_i}$, mais peu de résultats sont connus pour cette généralisation. Nous avons pu prouver des résultats structurels pour la version univarié de ce modèle, qui nous ont ensuite permis d'obtenir des bornes inférieures et des algorithmes de reconstruction, qui répondent au problème suivant : étant donné $f = \sum \alpha_i (x - a_i)^{e_i}$ par la liste de ses coefficients, retrouver les α_i, a_i, e_i qui apparaissent dans la décomposition optimale de f. Nous avons aussi étudié plus en détails la version multivarié du modèle, qui avait été laissé ouverte par nos précédents algorithmes de reconstruction, et avons obtenu plusieurs résultats lorsque le nombre de termes dans une expression optimale est relativement petit devant le nombre de variables ou devant le degré du polynôme.

Mots-clés : complexité algébrique, théorie de Valiant, bornes inférieures, indépendance linéaire, algorithmes de reconstruction, problème de Waring, décalage creux.

Introduction

Nous demandons fréquemment à nos ordinateurs de résoudre des problèmes pour nous, comme par exemple : quel est l'itinéraire le plus court, depuis ma position, pour me rendre à la soutenance de cette thèse ? Quel est le billet d'avion le moins cher pour partir en vacances une fois cette soutenance terminée ? Nous sommes en général assez exigeant vis-à-vis de la réponse à notre requête. D'une part, nous voulons que l'ordinateur réponde *rapidement* : il serait dommage de louper la soutenance de thèse à cause d'un temps de calcul trop long ! Mais d'autre part, nous voulons la *meilleure* solution pour éviter de gaspiller inutilement nos ressources. Ces deux exigences ne sont pourtant pas toujours compatibles, comme l'indique le proverbe :

« Mieux vaut bien faire que faire vite. » (Dicton français)

Par exemple, lorsque nous cherchons le *meilleur* billet, il faut nécessairement parcourir toutes les possibilités proposées par les différentes compagnies afin de garantir qu'il s'agit du moins cher (ou du plus rapide) possible. Ainsi, quelque soit la méthode utilisée pour classer et trier tous ces résultats, nous ne pourrons pas mettre moins de temps pour trouver le meilleur billet que le temps nécessaire pour collecter et lister ces données. Ce *temps minimal nécessaire* pour résoudre notre problème est communément appelé *borne inférieure* en informatique théorique. Il est tristement moins connu que son pendant *l'algorithme*, qui consiste en une méthode pour résoudre le problème posé. Lorsqu'un algorithme est trouvé, cela montre que le problème peut être résolu en *au plus* le temps correspondant, d'où parfois l'appellation de *borne supérieure*. A l'inverse, lorsqu'une borne inférieure est prouvée, cela montre que le problème nécessite *au moins* le temps correspondant pour être résolu via n'importe quelle méthode. Ainsi, le travail des chercheurs concernant la résolution d'un problème est double : trouver des algorithmes de plus en plus efficaces, c'est-à-dire qui nécessitent de moins en moins de temps ; et prouver des bornes inférieures de plus en plus grandes, qui permettent d'affiner le temps minimal nécessaire pour résoudre le problème. Dans certains cas, par exemple le tri de données, les algorithmes

et les bornes inférieures se rejoignent : on dispose alors d'une méthode pour résoudre notre problème, et on sait qu'on ne pourra pas faire mieux. Dans la terminologie de l'informatique théorique, on dit alors qu'on a trouvé un algorithme *optimal* et que l'on connaît la *complexité* du problème.

Il est alors assez naturel de classifier les différents problèmes en fonction de leur complexité. C'est ainsi que dans les années 1960, Cobham et Edmonds ont indépendamment défini la classe P des problèmes pour lesquels un algorithme *polynomial* existe, c'est-à-dire un algorithme « efficace ». Par exemple, les problèmes de trouver l'itinéraire le plus court ou le billet le moins cher appartiennent tous les deux à la classe P. On peut alors se demander : y a-t-il des problèmes qui ne sont pas dans la classe P ? La réponse est positive : il existe une méthode automatique pour construire des problèmes complexes qui ne sont pas dans P, mais les problèmes générés ne sont pas très intéressants car ils sont construits pour être difficiles, ce qui les rend assez artificiels. Cependant, il existe une classe de problèmes naturels qui pourrait être plus complexe que P : il s'agit de la classe NP des problèmes pour lesquelles on peut *vérifier* de manière efficace si une solution donnée est valide. Par exemple, imaginons le problème du touriste qui arrive en France et qui veut visiter certaines villes (par exemple Angers, Bordeaux, Caen, Clermont-Ferrand, Grenoble, Lille, Lyon, Nancy, Nice, Paris et Rennes) mais qui n'a à sa disposition qu'une voiture de location avec un forfait de 1500 kilomètres. Peut-il trouver un itinéraire de moins de 1500km passant par toutes ces villes ? Ce problème, connu sous le nom du *problème du voyageur de commerce*, appartient à la classe NP puisque, étant donné un itinéraire, il est facile de vérifier qu'il passe bien par toutes ces villes et qu'il fait moins de 1500km. Cependant, aucun algorithme efficace résolvant ce problème n'est connu à ce jour, donc la question reste ouverte de savoir si ce problème appartient également à la classe P. Plus généralement, la question de l'égalité des classes P et NP est souvent connue sous la dénomination « $\mathsf{P} = \mathsf{NP}$? », et sa solution a été mise à prix à un million de dollars par l'Institut de mathématiques Clay. La plupart des spécialistes du sujet pensent que ces deux classes sont différentes, mais la solution semble aujourd'hui encore hors de portée.

Dans cette thèse, les problèmes auxquels nous nous intéresserons seront principalement de nature *algébrique*. En d'autres termes, nous nous posons la question suivante : étant donné un polynôme f, quel est le nombre minimal d'opérations arithmétiques (l'addition, la soustraction ou la multiplication par exemple) nécessaires pour calculer f ? Ce genre de questions intervient assez naturellement dès le collège où l'on apprend *l'identité remarquable* : $(a+b)^2 = a^2 + 2ab + b^2$. Ainsi, nous préférerons par exemple représenter le polynôme $f(x) = 5x^4 + 30x^3 + 70x^2 + 75x + 31$ sous la forme $f(x) = (x+2)^5 - (x+1)^5$, qu'on appellera *somme de puissances affines* dans la suite. En 1979, Valiant introduit un modèle de calcul, aujourd'hui connu sous le nom *modèle de Valiant* pour étudier ce genre de question. De manière similaire à ce qui précède, il classifie alors les familles de polynômes en fonction de leur complexité et définit en particulier les classes VP et VNP, analogues des classes P et NP dans le monde algébrique. Intuitivement, la classe VP est constituée des polynômes qui admettent une représentation de taille polynomiale

tandis que la classe VNP est constituée des polynômes qui admettent une description (implicite) de taille polynomiale. Il se pose alors la même question « VP = VNP ? » concernant l'égalité de ces classes que dans le cas booléen. Grâce à la structure riche des anneaux de polynômes, on espère que cette question soit plus simple à résoudre que « P = NP ? », et que sa résolution permette d'avoir un éclairage nouveau sur cette question du millénaire.

Dans cette thèse, nous nous intéresserons en particulier au modèle des sommes de puissances affines, c'est-à-dire aux représentations de polynômes sous la forme

$$\sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}, \qquad \text{avec } \alpha_i, a_i \in \mathbb{F}, e_i \in \mathbb{N}.$$

Au travers des six chapitres de ce manuscrit, nous étudierons plusieurs aspects de ce modèle : résultats structurels, bornes inférieures et algorithmes de reconstruction.

Dans le premier chapitre, nous ferons une petite balade à travers la complexité algébrique pour expliquer quelles sont les motivations qui nous ont poussées à étudier ce modèle. Ce sera aussi l'occasion de présenter deux modèles plus classiques et déjà étudiés, les *décompositions de Waring* et *« sparsest shift »*, et de montrer en quoi le modèle principal est une généralisation naturelle de ces deux modèles.

Dans le deuxième chapitre, nous étudierons plus en détails les différences puissances d'expressivité de ces trois modèles. Nous montrerons en particulier que les modèles de Waring et du sparsest shift sont orthogonaux, au sens où aucun polynôme non-trivial ne peut admettre une représentation compacte simultanément dans ces deux modèles. Nous prouverons également des résultats structurels concernant notre modèle, notamment des conditions suffisantes pour garantir l'unicité de la décomposition optimale, et une borne supérieure sur les exposants pouvant intervenir dans une décomposition optimale. Ces résultats serviront par la suite à beaucoup d'autres endroits : pour les résultats de bornes inférieures, d'indépendance linéaire, et pour les algorithmes de reconstruction. Le chapitre est organisée en deux parties : une pour l'étude des polynômes à coefficients réels, qui s'appuie sur des résultats récents d'interpolation de Birkhoff; et une dans laquelle on cherche à étendre ces résultats à des polynômes à coefficients dans un corps arbitraire de caractéristique zéro.

Nous introduirons ensuite l'outil principal de cette thèse au chapitre 3 : « les équations différentielles décalées », qui sont des équations différentielles linéaires à coefficients polynomiaux, satisfaisant certaines contraintes sur les degrés des coefficients. Puis nous utiliserons cet outil pour prouver des bornes inférieures pour notre modèle, à l'aide de l'argument clé suivant : certaines familles de polynômes ne peuvent vérifier aucune équation différentielle décalée de petite taille alors que les puissances affines, et donc leurs sommes, en vérifient. Le reste du chapitre concernera l'étude de l'indépendance linéaire des puissances affines, qui intervient naturellement lorsque l'on cherche à prouver des bornes inférieures pour des polynômes qui s'écrivent comme sommes de puissances affines. Nous ferons la conjecture suivante : des puissances affines $\{(x-a_i)^{e_i} : i \in [\![1, s]\!]\}$ sont linéairement indépendantes dès lors que $e_i \geq as + b$ pour certaines constantes a et b. Dans le cas réel, nous montrerons que cette conjecture est vraie avec $a = 2$, $b = -4$ à l'aide des résultats

d'interpolation de Birkhoff, et nous prouverons que c'est optimal. Dans le cas complexe, nous montrerons une version plus faible de cette conjecture, en remplaçant la borne linéaire $as + b$ par la borne quadratique $\binom{s}{2}$. Finalement, nous étudierons la question, plus simple, de borner inférieurement la dimension de l'espace vectoriel engendré par un ensemble de puissances affines. Dans le cas réel, nous obtiendrons une borne inférieure optimal en $s/2$ et dans le cas complexe, nous parviendrons cette fois à conserver une borne inférieure linéaire en $(1 - \sqrt{2})s$, sans toutefois savoir si elle est optimale.

Nous nous concentrerons ensuite sur la conception d'algorithmes de reconstruction qui répondent au problème suivant : étant donné un polynôme, quelle est sa décomposition optimale comme somme de puissances affines ? Le chapitre 4 concerne l'étude du cas univarié, pour lequel nous commencerons par considérer le cas plus simple où les nœuds de la décomposition optimale ne sont pas répétés et où tous les exposants sont grands. Un premier algorithme sera décrit dans ce cadre, qui fonctionne avec un nombre polynomial d'opérations, et dont la complexité binaire est également polynomiale. Nous relâcherons ensuite l'hypothèse sur les exposants pour obtenir un algorithme plus général, de complexité polynomiale (arithmétique et binaire) en la taille de la décomposition optimale. Dans la deuxième partie de ce chapitre, nous étudierons le cas où les nœuds peuvent être répétés et fournirons des algorithmes pour deux scenarios particuliers : lorsque les exposants correspondant à un même nœud appartiennent à un petit intervalle, ou au contraire lorsque les exposants d'un même nœuds sont tous éloignés.

Dans le chapitre 5, nous nous intéresserons au cas des polynômes multivariés et nous concevrons des algorithmes lorsque le nombre de termes dans la décomposition optimale est petit par rapport au nombre de variable ou au degré. Un premier algorithme sera présenté pour le cas de base où le nombre de terme dans la décomposition optimale est égale au nombre de variables essentielles du polynôme. Deux directions seront ensuite étudiées pour généraliser ce résultat. Dans la première, nous conserverons la limitation sur le nombre de formes affines pouvant intervenir dans la décomposition, en autorisant cependant celles-ci à être répétées. Ceci nous conduira à étudier le problème de *décomposition univarié* et à concevoir un algorithme pour résoudre ce dernier. La deuxième direction que nous explorerons sera d'autoriser plus de formes affines dans la décomposition, en imposant cependant qu'elles ne se répètent pas. Nous obtiendrons ainsi un algorithme qui peut reconstruire jusqu'à $\binom{s+1}{2}$ puissances de formes affines, sous l'hypothèse que les exposants soient plus grands que 5, et que les coefficients du polynôme considéré soient génériques. Pour finir ce chapitre, nous proposerons un algorithme qui se ramène au cas univarié du chapitre 4 en procédant par projections univariées aléatoires, puis qui effectue un relèvement de ces solutions univariées en une solution du problème multivarié.

Enfin, en guise de conclusion, le chapitre 6 synthétisera les différents résultats de cette thèse, et listera des problèmes laissés ouverts et des questions qu'il pourrait être intéressant d'étudier.

Contents

1

Prolegomena

In this chapter, we will introduce the models and the questions investigated in this work. We will also explain the various motivations that lead to the study of these models. We begin by giving a quick and abrupt presentation of the main model of interest: *sums of affine powers*. Let $\mathbb{F}$ be any field of characteristic zero and let $f \in \mathbb{F}[X] = \mathbb{F}[x_1, \ldots, x_n]$ be a polynomial.

Model 1. *We consider expressions of f of the form*

$$\sum_{i=1}^{s} \alpha_i \ell_i^{e_i}(X)$$

with ℓ_i an affine form, $e_i \in \mathbb{N}$, $\alpha_i \in \mathbb{F}$. We denote by $\mathit{AffPow}_{\mathbb{F}}(f)$ the minimum value s such that there exists a representation of the previous form with s terms.

In most of the following chapters, the polynomials we consider are univariate, in which case Model 1 can be rewritten as follow.

Model 2. *We consider expressions of $f \in \mathbb{F}[x]$ of the form*

$$\sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}$$

with $\alpha_i, a_i \in \mathbb{F}$, $e_i \in \mathbb{N}$. Since increasing the number of variables does not help, we also denote by $\mathit{AffPow}_{\mathbb{F}}(f)$ the minimum value s of such a representation of f.
We will usually refer to the e_i's as the exponents *of the decomposition, and to the a_i's as the* nodes *of the decomposition.*

Example 1.0.1. *The polynomial $f = x^3 + 5x^2 + 14x + 7$ can be written in Model 2 in the following ways:*

$$\begin{aligned} f &= 1 \times (x-0)^3 + 5 \times (x-0)^2 + 14 \times (x + 1/2) \\ &= 1 \times (x+2)^3 - 1 \times (x-1)^2 \end{aligned}$$

The first equality shows that $\mathit{AffPow}_{\mathbb{F}}(f) \leq 3$, and the second expression shows that in fact $\mathit{AffPow}_{\mathbb{F}}(f) \leq 2$. One could easily prove that $\mathit{AffPow}_{\mathbb{F}}(f) \neq 1$, hence we have $\mathit{AffPow}_{\mathbb{F}}(f) = 2$.

This choice of model may seem arbitrary at first, but we will see in the following that it has motivations coming from various areas, such as algebraic complexity, algebra and symbolic computation.

1.1 Algebraic complexity: an introduction

The classical boolean complexity aims to classify languages, i.e. sets of words over a finite alphabet. Once a model of computation is settled, we associate complexity

measures to a language and then group together languages that have a similar complexity. For example, if the computation is done using Turing Machines, a language is associated with two complexity measures: the time and the memory needed to recognize this language. There are a lot of boolean complexity classes, the most famous ones being P and NP, for which we still do not know whether they are equal or not. In algebraic complexity, the objects that are studied are no longer set of words over a finite alphabet but families of polynomials over a field $\mathbb{F}$. However, the question remains the same: how hard is it to compute a polynomial f? More precisely, we need to define a model of computation for polynomials and the associated complexity measures. There are many models of computation, the most famous being *arithmetic circuits*, an algebraic analogous of boolean circuits, see Figure 1.1 for an example.

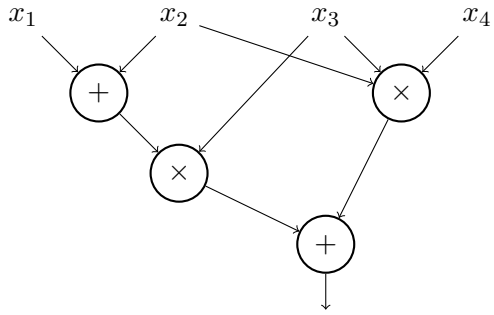

Figure 1.1: An arithmetic circuit computing $(x_1 + x_2)x_3 + x_2x_3x_4$, of depth 3 and of size 4.

In this model, the inputs are variables $x_1, \ldots, x_n$, and the computation is performed using arithmetic operations $+, \times, -$, and may involve constants from the underlying field $\mathbb{F}$. The output of an arithmetic circuit is thus a polynomial (or a set of polynomials) in the input variables. Notice that we do not put any restriction on the *fan-in* of a gate which denotes the number of inputs of the gate. The usual complexity measures associated are *size* and *depth* of the circuit which capture the number of arithmetic operations and the maximal distance between an input gate and an output gate, respectively. These measures of complexity capture the *parallel complexity* of a polynomial P, i.e., how many steps does it take to compute P with an unlimited amount of processors. As in boolean complexity, we can gather together families of polynomials that have similar complexities and define algebraic complexity classes.

1.1.1 Valiant's complexity classes

Arithmetic classes VP and VNP were first defined in work of Valiant [70], in which he gave analogous definitions for the classes P and NP in the algebraic world, and exhibited a complete problem for the later class. We now give some definitions of these classes to give an insight of the motivations for the problem we studied. More material about basic arithmetic complexity can be found in [17, 16]. The first complexity class aims to capture "polynomially bounded" families of polynomials.

However, it is not enough to require polynomial size circuits, since the polynomial $f_n = x^{2^n}$ has $O(n)$ size circuits, as illustrated on Figure 1.2, but its degree is not polynomial in n. As a consequence, we will say that a family of polynomials $\{f_n :$

Figure 1.2: An arithmetic circuit computing a polynomial that has exponential degree in the size of the circuit.

$n \geq 1\}$ is a *p-family* if there exists some polynomial $p : \mathbb{N} \to \mathbb{N}$ such that the number of variables and the degree of f_n are bounded by $p(n)$. The definition of VP will consist of the p-families that admit arithmetic circuits of polynomial size. Notice also that the definition of all the algebraic complexity classes depends on the underlying fields over which we allow computations to take place.

Definition 1.1.1 (VP)**.** *A p-family of polynomials $\{f_n\}$ over $\mathbb{F}$ is* p-computable *if there exists some polynomial $p : \mathbb{N} \to \mathbb{N}$ such that there is an arithmetic circuit of size at most $p(n)$ computing f_n. The class $\mathsf{VP}_\mathbb{F}$ consists of all p-computable families over $\mathbb{F}$.*

Remark 1.1.2. *This restriction on the degree makes in fact VP more analogous to NC^2 than to P. Indeed, a depth reduction theorem proved in [71] states that any polynomial size algebraic circuit computing a n-variate polynomial of degree d can be turned into an algebraic circuit of polynomial size and depth $O(\log d \log n)$ (in fact we even have stronger depth reduction theorems, as we will see in Section 1.1.2). If the degree d is polynomially bounded in n, we end up with circuits of depth $O(\log^2 n)$, giving the analogy with NC^2.*

Example 1.1.3. *A natural family in VP is the family of determinants:*

$$\mathsf{DET}_n(X) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} x_{i,\sigma(i)}$$

An easy way to see that $(\mathsf{DET}_n) \in \mathsf{VP}$ *is to compute it using Gauss pivot algorithm, which yields* $O(n^3)$ *size circuits, and then use the method of elimination of divisions due to Strassen [68]. One could also directly design an efficient parallel algorithm without division, as in [64].*

Similarly to NP, the class VNP is defined as follows from the class VP using some notion of "definability".

Definition 1.1.4 (VNP)**.** *A* p*-family of polynomials* $\{f_n\}$ *over* $\mathbb{F}$ *is* p-definable *if there exists a* p*-family* $\{g_n\}$ *in* $\mathsf{VP}_\mathbb{F}$ *and two polynomially bounded functions* $p, k : \mathbb{N} \to \mathbb{N}$ *such that for every* $n \in \mathbb{N}$*:*

$$f_n(x_1, \ldots, x_{k(n)}) = \sum_{w \in \{0,1\}^{p(n)}} g_{p(n)}\left(x_1, \ldots, x_{k(n)}, w_1 \ldots, w_{p(n)}\right)$$

The class $\mathsf{VNP}_\mathbb{F}$ *consists of all* p*-definable families over* $\mathbb{F}$*.*

Remark 1.1.5. *In the definition of* VNP*, the tuple* $(w_1, \ldots, w_{p(n)})$ *can be seen as the "witness" and the summation is the algebraic equivalent of the existential quantifier for* NP *problems, showing the analogy between the two classes. In fact, the summation is more powerful than a simple existential quantifier, which makes* VNP *more analogous to* #P *than to* NP*, and indeed they share complete problems (see Definition 1.1.8 for the definition of* p*-projection and* VNP*-complete problems).*

Example 1.1.6. *A natural family in* VNP *is the family of permanents:*

$$\mathsf{PERM}_n(X) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} x_{i,\sigma(i)}$$

To see that $(\mathsf{PERM}_n) \in \mathsf{VNP}$*, we follow the proof of [16, Lemma 2.6]: we sum over all* $n \times n$ *matrices with* $0/1$ *entries and keep only the ones that correspond to a permutation. In other words, we write:*

$$\mathsf{PERM}_n(X) = \sum_{Y \in \{0,1\}^{n \times n}} \mathrm{Permutation}(Y) \cdot \prod_{i=1}^{n} \left(\sum_{j=1}^{n} x_{i,j} Y_{i,j} \right),$$

where $\mathrm{Permutation}(Y)$ *is a polynomial that evaluates to* 1 *if the input matrix is a permutation matrix, and* 0 *otherwise. One expression of* $\mathrm{Permutation}(Y)$ *is given by:*

$$\mathrm{Permutation}(Y) = \underbrace{\prod_{i=1}^{n} \sum_{j=1}^{n} Y_{i,j}}_{\text{at least one 1 in each row}} \cdot \underbrace{\prod_{i=1}^{n} \prod_{j=1}^{n} \prod_{\substack{k=1 \\ k \neq j}}^{n} (1 - Y_{i,j} Y_{i,k})(1 - Y_{j,i} Y_{k,i})}_{\text{at most one 1 in each row/col}}$$

By definition of VP and VNP, it directly follows that $\mathsf{VP} \subseteq \mathsf{VNP}$. As an analogue to the famous P vs. NP question, Valiant conjectured that the inclusion is strict:

Conjecture 1.1.7. *[70]* $\mathsf{VP} \neq \mathsf{VNP}$.

One could hope Valiant's conjecture to be easier than its classical counterpart for several reasons. Firstly, arithmetic circuits have a lot of structure, which makes them easier than Turing Machines to work with. Secondly, as discussed before, the classical counterpart of VP and VNP are NC^2 and $\#P$, which are easier to separate than P and NP. Still, how do we compare families of polynomials? In boolean complexity, we have many-one reductions and Turing reductions for comparing languages; what is the algebraic analogue? Valiant proposed *projections* as reductions for two families of polynomials:

Definition 1.1.8. *The family* $\{f_n\}$ *is a* p-projection *of* $\{g_n\}$ *if there exists a polynomially bounded* $p : \mathbb{N} \to \mathbb{N}$ *such that for all* n, f_n *can be derived from* $g_{p(n)}$ *by a substitution of the variables by other variables or constants in* $\mathbb{F}$.

As one would expect, both VP and VNP are closed under p-projections. Similarly to the definition of P-complete and NP-complete problems, we define VP-complete and VNP-complete families of polynomials. The choice of (PERM_n) as an example of a family in VNP is not arbitrary: Valiant showed in [70] PERM is complete for VNP. In particular, this implies that Conjecture 1.1.7 is equivalent to proving a super-polynomial lower bound on the size of the circuits computing the permanent.

1.1.2 Restricted arithmetic circuit classes and depth reduction

As an intermediate step to obtain lower bounds for general circuits, people usually first prove lower bounds for restricted circuit classes. We focus here on depth restriction, defined as follows:

Definition 1.1.9 (Bounded-depth circuits)**.** *A family of circuits* $\{C_i\}$ *is of bounded depth if there exists a constant* $d \in \mathbb{N}$ *such that for any* n, C_n *has depth at most* d.

Remark 1.1.10. *These algebraic constant depth circuits are the algebraic counterpart of the class* AC^0 *which denotes the set of boolean circuits of fixed depth. To obtain interesting circuits, it is crucial that the fan-in (number of inputs) of the gates is unbounded, as there would be only finitely many circuits of depth* d *otherwise.*

In the following, we will consider the case of depth-4 circuits, also known as $\Sigma\Pi\Sigma\Pi$ circuits (we will explain this choice later on). A $\Sigma\Pi\Sigma\Pi$ circuit is a depth-4 circuit with an addition gate at the bottom (output) then a layer of multiplication gates, then a layer of addition gates, then multiplication gates at top, as illustrated in Figure 1.3.

In other words, it computes a polynomial of the form:

$$\sum_{i=1}^{k} \prod_{j=1}^{m} \sum_{l=1}^{t} \prod_{p \in S_{i,j,l}} x_p$$

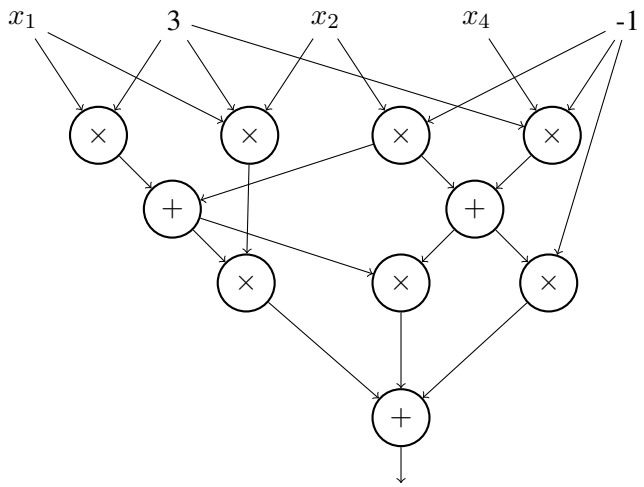

Figure 1.3: A depth-4 arithmetic circuit.

where x_p is either an input variable or a constant in $\mathbb{F}$. We will also use the notation $\Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ to denote a $\Sigma\Pi\Sigma\Pi$ circuits with $m = a$ and $|S_{i,j,l}| \leq b$ for all i, j, k. Notice that each product gate at top defines a monomial, so that we usually define

$$f_{i,j}(x_1, \ldots, x_n) \stackrel{\text{def}}{=} \sum_{l=1}^{t} \prod_{p \in S_{i,j,l}} x_p,$$

and we set $r = \max_{i,j} \deg(f_{i,j})$. Therefore, $\Sigma\Pi\Sigma\Pi$ circuits compute polynomials of the form $\sum_{i=1}^{k} \prod_{j=1}^{m} f_{i,j}(x_1, \ldots, x_n)$, where the $f_{i,j}$'s are multivariate polynomials such that $\deg(f_{i,j}) \leq r$, and the $f_{i,j}$ are given as sum of monomials.

Remark 1.1.11. *The choice of $\Sigma\Pi\Sigma\Pi$ rather than $\Pi\Sigma\Pi\Sigma$ isn't arbitrary: when we consider depth-d circuits, it's usually more interesting to consider circuits with an additive output gate. Indeed, if a polynomial f is computed by a circuit of depth d with a multiplicative output gate, we can always consider sub-circuits of depth $d-1$ which computes the factors of f. In the case of the additive output gate, it's more difficult to do the same because of possible cancellations: the sub-circuits of depths $d-1$ may compute polynomials of degree $> \deg(f)$ and the final addition may cancel the term of higher degrees.*

The importance of $\Sigma\Pi\Sigma\Pi$ circuit comes from the following result: Agrawal and Vinay [1] and subsequent strengthenings of Koiran [50] and Tavenas [69] showed that depth-4 circuits are as interesting as general circuits:

Theorem 1.1.12 ([1] [50] [69] Depth-reduction)**.** *Let f be an n-variate polynomial computed by a circuit of size s and of degree d. Then f is computed by*

a $\Sigma\Pi^{[O(\alpha)]}\Sigma\Pi^{[\beta]}$ *circuit* C *of size* $2^{O\left(\sqrt{d\log(ds)\log n}\right)}$ *where* $\alpha = \sqrt{d\frac{\log n}{\log ds}}$ *and* $\beta = \sqrt{d\frac{\log ds}{\log n}}$.
In particular when $s, d = n^{O(1)}$*, then* f *is computed by a* $\Sigma\Pi^{[O(\sqrt{d})]}\Sigma\Pi^{[O(\sqrt{d})]}$ *circuit* C *of size* $n^{O(\sqrt{d})}$.

This depth-reduction theorem implies that lower bounds for the depth-4 arithmetic circuit model will give lower bounds for general arithmetic circuits. Recent results of [36, 46, 28] gave lower bound that comes very close to the required threshold for different polynomial. For instance, Gupta, Kamath, Kayal and Saptharishi [36] showed the following lower bound for DET and PERM:

Theorem 1.1.13 ([36])**.** *Any* $\Sigma\Pi^{[O(\sqrt{n})]}\Sigma\Pi^{[\sqrt{n}]}$ *circuit computing* DET_n *or* PERM_n *has bottom fan-in* $2^{\Omega(\sqrt{n})}$.

Using the following formula due to Fischer [26], we can replace the product gate with a powering gate:

$$x_1 \cdot \ldots \cdot x_d = \frac{1}{d!} \sum_{\varepsilon \in \{-1,1\}^{d-1}} (x_1 + \varepsilon_1 x_2 + \ldots + \varepsilon_{d-1} x_d)^d.$$

As a consequence of this formula and Theorem 1.1.12, one could proved the following result that we first stated in [42].

Proposition 1.1.14. *Let* $\{f_n(X) : n \geq 1\}$ *be a family of* n*-variate polynomials of degree* $d = d(n)$ *over an underlying field* $\mathbb{F}$ *which is algebraically closed and has characteristic zero. If this family is in* VP *then* $f_n(X)$ *admits a representation of the form*

$$f_n(X) = \sum_{i=1}^{s} Q_i(X)^{e_i} \quad \textit{where} \quad \deg(Q_i) \leq \sqrt{d}$$

and where the number of summands s *is at most* $n^{O(\sqrt{d})}$.

This proposition motivates the investigation of the following model of sum of powers of bounded degree polynomials:

Model 3. *We consider expressions of* $f \in \mathbb{F}[X]$ *of the form*

$$\sum_{i=1}^{s} Q_i(X)^{e_i} \quad \textit{with } \deg(Q_i) \leq r.$$

As a consequence, similarly to $\Sigma\Pi\Sigma\Pi$ circuits, strong enough lower bounds for Model 3 imply general circuit lower bound. In particular, the contrapositive version of Proposition 1.1.14 means that a strong enough (at least $n^{w(\sqrt{d})}$) lower bound for representing an explicit family of polynomials $\{f_n(X) : n \geq 1\} \in \mathsf{VNP}$ in

Model 3 will imply that this family is not in VP, thereby separating VP and VNP. Promising progress along this direction has been recently obtained. In [41], Kayal already investigated Model 3 and proved a $2^{\Omega(\sqrt{d})}$ lower bound in this model, using a complexity measure called *dimension of shifted partials* (see Chapter 3 for more details). Follow up work [36, 46] obtained an $n^{\Omega(\sqrt{d})}$ lower bound for Model 3, thereby coming tantalizingly close to the threshold required for obtaining superpolynomial lower bounds for general circuits. Since then, these techniques have been intensely investigated and followup work by [27, 53, 45] have used these techniques to obtain optimality of the known depth reduction results in many interesting cases. Some of these works also suggest that the dimension of shifted partials in itself might not be strong enough to separate VP from VNP, and indeed this result was proved a bit later in [24]. In a very recent paper [23], the authors generalized this result to most the "rank methods", emphasizing the need for new lower bounds techniques.

1.1.3 The quest for new techniques

This was the starting point of this work: try to find new methods to prove lower bounds for Model 3. The angle of attack we chose was to first to focus on the univariate case:

Model 4. *We consider expressions of $f \in \mathbb{F}[x]$ of the form*

$$\sum_{i=1}^{s} Q_i(x)^{e_i} \quad \textit{with } \deg(Q_i) \leq r.$$

We denote by $s_r(f)$ the minimum value s such that there exists a representation of the previous form with s terms.

The main advantage of the univariate approach is that univariate polynomials are well-known objects, and one could hope to use e.g. some real or complex analysis tools to obtain lower bounds for this model. Our underlying hope is that some such improved proof technique or proof idea might admit a suitable generalization to the multivariate case as well. This could be one potential way to attack the VP versus VNP problem. Moreover, there are also formal results essentially following from the work of Koiran [51] which imply that seemingly mild lower bounds for a slight variant of Model 4 directly implies a separation of VP from VNP.

Proposition 1.1.15 (Implicit in [51])**.** *If there is an explicit family of* univariate *polynomials $\{f_d(x) : d \geq 1\}$ over an underlying field $\mathbb{F}$ which is algebraically closed and has characteristic zero such that any representation of the form $f_d(x) = \sum_{i=1}^{s} Q_i(x)^{e_i}$, where $\mathrm{Sparsity}(Q_i) \leq t$, requires the number of summands s to be at least $\left(\frac{d}{t}\right)^{\Omega(1)}$, then $\mathsf{VP} \neq \mathsf{VNP}$.*

This means that proving relatively mild lower bounds on a similar model (but with the degree bound replaced by the corresponding sparsity bound) already implies that VP is different from VNP. In fact, in the same paper, Koiran already proposed a proof

technique quite different from all the "rank methods" which relies on the number of real roots of polynomial. He made the following τ-conjecture, which directly implies that $\mathsf{VP} \neq \mathsf{VNP}$:

Conjecture 1.1.16. *Consider a nonzero polynomial of the form:*

$$\sum_{i=1}^{k} \prod_{j=1}^{m} f_{i,j}(x)$$

where each $f_{i,j}$ has at most t monomials. Then the number of real roots of f is bounded by a polynomial function of kmt.

In [42], we also proposed a new proof technique to prove lower bounds for Model 4 that uses the Wronskian. We managed to find two families of (explicit) polynomials such that the minimal number of summands to find a decomposition in Model 4 is $\Omega\left(\sqrt{\frac{d}{r}}\right)$ (the proofs are detailed in Section 3.2.3). This should be compared to the fact that for a random polynomial $f(x)$ of degree d, it is almost surely the case that $s_r(f) \geq \frac{d+1}{r+1}$ (an even stronger result is proven in Corollary 2.2.14). To this day, this is still the best lower bound for Model 4, and even for the case $r = 1$ no $\Omega(d)$ lower bound is known, except in the case when $\mathbb{F} = \mathbb{R}$ where Garcìa-Marco and Koiran [29] proved an optimal $\Omega(d)$ lower bound, using Birkhoff interpolation techniques. This was one of the main motivations to study Model 2 and its multivariate counterpart, Model 1. An interesting fact is that the generic case is also the worst case, and we have a simple explicit construction for such cases. More precisely, we have the following constructive upper bound (stated for the case $r = 1$, but can be generalized to arbitrary r), already mentioned in [29, Proposition 18].

Proposition 1.1.17. *For all polynomials $f \in \mathbb{F}[x]$ of degree d, we have*

$$\mathit{AffPow}_{\mathbb{F}}(f) \leq \left\lceil \frac{d+1}{2} \right\rceil.$$

Proof. We use induction on d. Since the result is obvious for $d = 0, 1$ we consider a polynomial $f = \sum_{i=0}^{d} a_i x^i$ of degree $d \geq 2$, and we assume that the result holds for polynomials of degree $d-2$. We observe that $g := f - a_d(x + (a_{d-1}/da_d))^d$ has degree $\leq d-2$. Applying the induction hypothesis to g we get that $\mathrm{AffPow}_{\mathbb{F}}(g) \leq \lceil (d-1)/2 \rceil$, proving that $\mathrm{AffPow}_{\mathbb{F}}(f) \leq \lceil (d-1)/2 \rceil + 1 = \lceil (d+1)/2 \rceil$. □

1.2 Waring and Sparsest Shift models

Model 2 extends two already well-studied models: Waring and Sparsest Shift. The decompositions allowed in these models must satisfy additional constraints on either the exponents or the nodes, making Model 2 more general.

1.2.1 Waring decompositions

In a Waring decomposition, all the exponents are equal to the degree of the polynomial, i.e., $e_i = \deg(f)$ for all i.

Model 5. *For a polynomial f of degree d, we consider expressions of f of the form:*

$$\sum_{i=1}^{s} \alpha_i (x - a_i)^d$$

with $\alpha_i, a_i \in \mathbb{F}$. We denote by $\text{Waring}_{\mathbb{F}}(f)$ the Waring rank of f, which is the minimum value s such that there exists a representation of the previous form with s terms.

Usually, the Waring rank is studied for homogeneous multivariate polynomials, that is, a polynomial whose nonzero terms all have the same degree. In this context, the study of Model 5 is reduced via homogenization to the study of bivariate homogeneous polynomials for the following model.

Model 6. *For a n-variate homogeneous polynomial $f \in \mathbb{F}[X]$ of degree d, we consider expressions of f of the form:*

$$\sum_{i=1}^{s} \alpha_i \ell_i^d(X)$$

with $\alpha_i \in \mathbb{F}$ and ℓ_i a linear form. Since there is no ambiguity, we will also denote by $\text{Waring}_{\mathbb{F}}(f)$ the Waring rank of f in this model.

Waring rank has been studied by algebraists and geometers since the 19th century. The algorithmic study of Model 5 (bivariate Model 6) is usually attributed to Sylvester. We refer to [38] for the historical background and to section 1.3 of that book for a description of the algorithm (see also Kleppe [48] and Proposition 46 of Kayal [41]). Most of the subsequent work was devoted to the general case of Model 6 (that is, for ≥ 3 variables) with much of the 20th century work focused on the determination of the Waring rank of generic polynomials [2, 14, 38]. A few recent papers [55, 8] have begun to investigate the Waring rank of specific polynomials such as monomials, sums of co-prime monomials, the permanent and the determinant. Model 6 has also been studied from an algorithmic point of view, see e.g. [43, 44, 13, 62].

At the moment, the best upper bound we have for Model 1 are the ones given by the Waring model. For a homogeneous polynomial $f \in \mathbb{F}[X]$ of degree d with n variables, we have the trivial upper bound $\text{Waring}_{\mathbb{F}}(f) \leq \binom{n+d}{d}$ by a dimension argument, but several recent works proved non trivial improvements on the maximum value of $\text{Waring}_{\mathbb{F}}(f)$, see e.g. [7, 39]. As a consequence, we have the following upper bound on $\text{AffPow}_F F(f)$:

Proposition 1.2.1. *Let $f \in \mathbb{F}[X]$ be a polynomials of degree d with n variables. Then*

$$\textit{AffPow}_{\mathbb{F}} \leq \binom{n+d-1}{d-1} - \binom{n+d-5}{d-3}$$

Proof. Homogenize f and apply results of [39]. □

Remark 1.2.2. *One could define an interesting intermediate model between Model 5 and Model 2 by only asking all the exponents to be equal, i.e. $e_i = e_j$ for all i, j instead of $e_i = \deg(f)$ for all i. For $e \in \mathbb{N}$, define Waring(f, e) as the minimum number of terms needed to express f as a sum of e-th powers of affine forms, and define the* generalized Waring rank *of f as GWaring$(f) \stackrel{def}{=} \min_{e\in\mathbb{N}}$ Waring(f, e). The interesting part of this "generalized Waring model" is that it allows higher order cancellations. For instance, the polynomial $H_d(x) = (x + 1)^{d+1} - x^d$ has a generalized Waring rank of 2, but we will prove in Proposition 3.3.5 that it has maximal Waring rank with Waring$_{\mathbb{F}}(H_d) = \lceil \frac{d+1}{2} \rceil$. A natural question to ask is whether there is a bound on the common value of the exponent of an optimal generalized Waring expression. We will answer this question in Corollary 2.2.10 by optimal bound on the maximum exponent in an optimal expression in Model 2. Another interesting object that we did not investigate is the sequence $(\text{Waring}(f, e))_{e\in\mathbb{N}}$: is it monotonous? If two consecutive values are equal, can we deduce that they are equal to the generalized Waring rank?*

1.2.2 Sparsest Shift

The second model that we generalize is the Sparsest Shift model, where all the shifts a_i are required to be equal.

Model 7. *For a univariate polynomial $f \in \mathbb{F}[x]$, we consider expressions of f of the form:*

$$\sum_{i=1}^{s} \alpha_i (x - a)^{e_i}$$

with $\alpha_i, a \in \mathbb{F}, e_i \in \mathbb{N}$. We denote by Sparsest$_{\mathbb{F}}(f)$ the minimum value s such that there exists a representation of the previous form with s terms.

This model and its variations have been studied in the computer science literature at least since Borodin and Tiwari [9]. Some of these papers deal with multivariate generalizations [35, 60], with "supersparse" polynomials[1] [33] or establish condition for the uniqueness of the sparsest shift [54]. It is suggested at the end of [60] to allow "multiple shifts" instead of a single shift, and this is just what we did in this thesis. More precisely, as is apparent from Model 2, we do not place any constraint on the number of distinct shifts: it can be as high as the number s of affine powers.

Remark 1.2.3. *As for the Waring model, one could define an intermediate model by placing an upper bound k on the number of distinct shifts. This would provide a smooth interpolation between the sparsest shift model (where $k = 1$) and Model 2, where $k = s$.*

[1] In that model, the size of the monomial x^d is defined to be $\log d$ instead of d as in the usual dense encoding.

This model is deeply linked with the notion of sparse representations of polynomials: instead of encoding a polynomial in a dense way, i.e. by giving the list of all its coefficients, one could encode only the nonzero coefficients along with their associated exponent. This representation is efficient for sparse polynomials, that is, polynomials that have a few nonzero terms. However, if we take the polynomial $f(x) = (x-2)^d$, both its dense and its sparse representation are large. Yet, the "shifted version" $f(x+2)$ of f is 1-sparse and thus one could encode f as the shift, which is 2, and then the sparse representation of $f(x+2)$. The optimal decomposition in Model 7 yields the smallest such representation, that is, a shift a such that $f(x+a)$ is the sparsest possible, and hence such a shift a is usually called a *sparsest shift*. As such, algorithms for computing a sparsest shift could therefore be considered simplification tools.

Remark 1.2.4. *The values of the sparsest shifts of a polynomial $f \in \mathbb{F}[x]$ are linked with the roots of f and its derivatives. Indeed, if f admits the following decomposition in Model 7:*

$$f(x) = \sum_{i=1}^{s} \alpha_i (x-a)^{e_i}$$

with $\alpha_i, a \in \mathbb{F}, e_i \in \mathbb{N}$, then for all $k \notin \{e_i : i \in [\![1, s]\!]\}$, we have $f^{(k)}(a) = 0$. In other words, the Taylor expansion of f about a is s-sparse. As a consequence, for a random polynomial $f \in \mathbb{F}[x]$, we have $\text{Sparsest}_{\mathbb{F}}(f) = d$ with high-probability.

2

Structural results and model comparisons

In this chapter we compare the expressive power of our 3 models: sums of affine powers, sparsest shift and the Waring decomposition. We will see in Section 2.2 that some polynomials have a much smaller expression as a sum of affine powers than in the sparsest shift or Waring models. Moreover, we show that Model 5 and Model 7 are "orthogonal" in the sense that (except in one trivial case) no polynomial can have a small representation in both models at the same time.
We begin this investigation of structural properties with the field of real numbers, where an especially strong version of orthogonality holds true. We also show that some real polynomials have a short expression as a sum of affine powers over the field of complex numbers, but not over the field of real numbers. This observation has algorithmic implications: given a polynomial $f \in \mathbb{F}[x]$, we may have to work in a field extension of $\mathbb{F}$ to find the optimal representation for f. These "real" results can be derived fairly quickly from results in [29]. We then move to arbitrary fields of characteristic zero in Section 2.2. In both cases, we also study the uniqueness of optimal representations. These results about uniqueness have a lot of non-trivial implications in the remaining chapters, e.g. lower bounds, reconstruction algorithms, linear independence.

Let us introduce an equality that we will use to obtain several extremal examples throughout this chapter. It is a generalization of the famous equality $(x+1)^2 - (x-1)^2 = 4x$, where we replace 1 and -1 by the successive powers of a primitive root of unity.

Example 2.0.1. *One can slightly modify [29, Proposition 19] to obtain the following equality of complex polynomials of degree d:*

$$\sum_{j=1}^{k} \xi^{\lambda j}(x+\xi^j)^d = \sum_{\substack{0\le i\le d \\ i\equiv\lambda \pmod k}} k\binom{d}{i}x^{d-i}$$

where $k, \lambda \in \mathbb{N}$ and $\xi \in \mathbb{C}$ is a k-th primitive root of unity.

2.1 The real case

In [29] the authors considered polynomials with real coefficients and proved the following result, which can be seen as a linear independence result for affine powers.

Theorem 2.1.1. *[29, Theorem 13] Consider a polynomial identity of the form:*

$$\sum_{i=1}^{k} \alpha_i (x-a_i)^d = \sum_{i=1}^{l} \beta_i (x-b_i)^{e_i}$$

where the $a_i \in \mathbb{R}$ are distinct constants, the constants $\alpha_i \in \mathbb{R}$ are not all zero, the $\beta_i \in \mathbb{R}$ and $b_i \in \mathbb{R}$ are arbitrary constants, and $e_i < d$ for every i. Then, we must have $k + l \ge \lceil (d+3)/2 \rceil$.

Theorem 2.1.1 will be our main tool in Section 2.1, and in Section 2.2 we will prove a similar result for fields of characteristic zero which will also be the main tool of Section 2.2.

2.1.1 Uniqueness and field extension

As a consequence of Theorem 2.1.1, we obtain a sufficient condition for a polynomial to have a unique optimal expression in Model 5 over the reals. We first introduce some notation that we will reuse throughout this thesis: given a polynomial of the form $f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}$, for any $e \in \mathbb{N}$, we denote by n_e the number of exponents smaller than e, i.e., $n_e \stackrel{\text{def}}{=} \#\{i : e_i < e\}$. It is natural to enforce some conditions on the n_e's in order to guarantee optimality or uniqueness of expression. Indeed, if f has an expression with $n_e = e + 1$ for some $e \in \mathbb{N}$, then f could be rewritten with less terms since the affine powers with exponents smaller than e are linearly dependent.

Corollary 2.1.2. *Let $f \in \mathbb{R}[x]$ be a polynomial of the form:*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i} \tag{2.1}$$

with $\alpha_i \neq 0$. If $2n_e \leq \lceil (e+2)/2 \rceil$ for all $e \in \mathbb{N}$, then $\mathit{AffPow}_{\mathbb{R}}(f) = s$. Moreover, if $2n_e < \lceil (e+2)/2 \rceil$ for all $e \in \mathbb{N}$ then (2.1) is the unique optimal expression for f.

Proof. Suppose that f can be written in another way

$$f = \sum_{j=1}^{p} \beta_j (x - b_j)^{f_j} \tag{2.2}$$

with $p \leq s$. Set $d = \max\left((e_i)_{1 \leq i \leq s} \cup (f_j)_{1 \leq j \leq p}\right)$ and denote by s' (respectively, p') the index such that $d = e_1 = \cdots = e_{s'} > e_{s'+1} \geq \cdots \geq e_s$ (respectively, $d = f_1 = \cdots = f_{p'} > f_{p'+1} \geq \cdots \geq f_p$). Note that one of the two indices s', p' will be equal to 0 if the exponent d appears only in one of the two expressions (2.1) and (2.2).
Combining equations (2.1) and (2.2), we obtain the following equality:

$$\sum_{i=1}^{s'} \alpha_i (x - a_i)^d - \sum_{j=1}^{p'} \beta_j (x - b_j)^d = - \sum_{i=s'+1}^{s} \alpha_i (x - a_i)^{e_i} + \sum_{j=p'+1}^{p} \beta_j (x - b_j)^{f_j}$$

We can rewrite this as

$$\sum_{i=1}^{k} \alpha'_i (x - a'_i)^d = \sum_{i=1}^{l} \beta'_i (x - b'_i)^{e'_i}$$

with $\alpha'_i \neq 0$, $k \leq s' + p'$ and $l \leq (s - s') + (p - p')$.
To prove the first assertion, let us assume that $2n_e \leq \lceil (e+2)/2 \rceil$ for all e. Assume

also for contradiction that $p < s$ and $k > 0$. By Theorem 2.1.1, we must have $k+l \geq \lceil (d+3)/2 \rceil$. The upper bounds on k and l imply $2s > s+p \geq k+l \geq \lceil (d+3)/2 \rceil$. However we have from our assumption that $2s = 2n_{d+1} \leq 2\lceil (d+3)/2 \rceil$, which contradicts the previous inequality. This shows that $p < s \Rightarrow k = 0$, i.e., if $p < s$ then the highest degree terms are the same. Continuing by induction, we find that all the terms in the two expressions are equal. In particular we would have $p = s$, a contradiction. This shows that $p \geq s$, i.e., that $\text{AffPow}_{\mathbb{R}}(f) = s$.
To prove the second assertion, let us now assume further that $2n_e < \lceil (e+2)/2 \rceil$ for all e. Assume also that $p = s$. By Theorem 2.1.1, either $k = 0$ or $k+l \geq \lceil (d+3)/2 \rceil$. In the second case, the upper bounds on k and l imply that $2s = s+p \geq k+l \geq \lceil (d+3)/2 \rceil$. This is in contradiction with the assumption that $2n_{d+1} < \lceil (d+3)/2 \rceil$. We conclude that k must be equal to 0, i.e., the highest degree terms are the same. Continuing by induction, we obtain that all the terms of the two decompositions are equal, thus showing that (2.1) is the unique optimal expression for f in this model. □

Remark 2.1.3. *Consider the degree $d \geq 2$ polynomial*

$$f \stackrel{def}{=} (x+1)^d + (x-1)^d = \sum_{\substack{i \text{ even} \\ 0 \leq i \leq d}} 2\binom{d}{i} x^{d-i}.$$

This polynomial has an expression in Model 2 with $n_e \leq \frac{e+1}{2}$ but this expression is not optimal since $\text{AffPow}_{\mathbb{R}}(f) = 2$. Hence, the inequality in Corollary 2.1.2 is optimal up to a factor 2.

As a consequence, we obtain an explicit polynomial such that $\text{AffPow}_{\mathbb{R}}(f)$ is arbitrarily larger than $\text{AffPow}_{\mathbb{C}}(f)$.

Example 2.1.4. *For every $d \in \mathbb{N}$, we consider the polynomial*

$$f_d := \sum_{\substack{j \equiv 3 \pmod 4 \\ 0 \leq j \leq d}} 4\binom{d}{j} x^{d-j} \in \mathbb{R}[x]. \tag{2.3}$$

We can express f_d as $f_d = (x+1)^d - (x-1)^d + i(x+i)^d - i(x-i)^d$, which proves that $\text{AffPow}_{\mathbb{C}}(f_d) \leq 4$. Moreover, in expression (2.3) we have $n_e \leq \lceil e/4 \rceil$ for all $e \in \mathbb{N}$. Since $2\lceil e/4 \rceil \leq \lceil (e+2)/2 \rceil$, it follows from Corollary 2.1.2 that this expression for f_d is optimal over the reals, i.e., $\text{AffPow}_{\mathbb{R}}(f_d) = \lfloor (d+1)/4 \rfloor$.

This should be compared with the following result about sparsest shift on a field $\mathbb{F}$ and a field extension $\mathbb{K}$ of $\mathbb{F}$. Theorem 1 in [54] shows that whenever the value $\text{Sparsest}_{\mathbb{K}}(f)$ is "small", then it is equal to $\text{Sparsest}_{\mathbb{F}}(f)$; more precisely, if we have $\text{Sparsest}_{\mathbb{K}}(f) \leq (d+1)/2$ then $\text{Sparsest}_{\mathbb{K}}(f) = \text{Sparsest}_{\mathbb{F}}(f)$. This is no longer the case for the Affine Power model as the previous example shows.

2.1.2 Orthogonality

As a consequence of Theorem 2.1.1 we can easily derive the following result about the orthogonality of Waring and sparsest shift models over the reals.

Corollary 2.1.5. *Let $f \in \mathbb{R}[x]$ be a polynomial of degree d. Either $f = \alpha(x-a)^d$ for some $\alpha, a \in \mathbb{R}$ (and $\mathit{Waring}_{\mathbb{R}}(f) = \mathit{Sparsest}_{\mathbb{R}}(f) = 1$), or the following holds:*

$$\mathit{Waring}_{\mathbb{R}}(f) + \mathit{Sparsest}_{\mathbb{R}}(f) \geq \frac{d+3}{2}$$

Proof. We set $k = \text{Waring}_{\mathbb{R}}(f)$ and $l = \text{Sparsest}_{\mathbb{R}}(f)$ and assume that $l \geq 2$. We write f in two different ways:

$$f = \sum_{i=1}^{k} \alpha_i (x-a_i)^d = \sum_{j=1}^{l} \beta_i (x-a)^{e_i},$$

where the $a_j \in \mathbb{R}$ are all distinct, and $e_1 < \cdots < e_l = d$. Let us move the term $\beta_l(x-a)^d$ to the left hand side of the equation. We then have two cases to consider:

- if $a \neq a_i$ for all i, we have $k+1$ terms on the left hand side of the equation and $l-1$ terms on the right hand side. Theorem 2.1.1 shows that $(k+1)+(l-1) \geq (d+3)/2$.
- If $a = a_i$ for some i, we have k or $k-1$ terms on the left hand side of the equation and $l-1$ terms on the right hand side. By Theorem 2.1.1, $k+(l-1) \geq (d+3)/2$.

□

Remark 2.1.6. *Consider the same polynomial as in Remark 2.1.3:*

$$f = (x+1)^d + (x-1)^d = \sum_{\substack{i \text{ even} \\ 0 \leq i \leq d}} 2\binom{d}{i} x^{d-i}.$$

We observe that $\mathit{Waring}_{\mathbb{R}}(f) = 2$ and $\mathit{Sparsest}_{\mathbb{R}}(f) \leq \lceil (d+1)/2 \rceil$. Hence, the inequality in Corollary 2.1.5 is optimal up to one unit.

A similar proof to that of Corollary 2.1.5 yields the following result about orthogonality of Waring decompositions and sums of affine powers over the reals:

Corollary 2.1.7. *Let $f \in \mathbb{R}[x]$ be a real polynomial of degree d. Then, either $\mathit{AffPow}_{\mathbb{R}}(f) = \mathit{Waring}_{\mathbb{R}}(f)$ or the following inequality holds:*

$$\mathit{Waring}_{\mathbb{R}}(f) + \mathit{AffPow}_{\mathbb{R}}(f) \geq \frac{d+3}{2}$$

2.2 Fields of characteristic zero

We now switch from the real field to an arbitrary field $\mathbb{F}$ of characteristic zero. We first prove a similar result to Theorem 2.1.1 by using the Wronskian, which will be the main tool of this section. We then derive some sufficient conditions to ensure uniqueness of optimal expression, and we show they are best possible. Finally, we give a comparison of the power of the different models and prove that they are again orthogonal, even though the results we obtain for a arbitrary field are weaker than in Section 2.1.

2.2.1 The Wronskian and linear independence

In mathematics, the *Wronskian* is a tool mainly used in the study of differential equations, where it can be used to show that a set of solutions is linearly independent.

Definition 2.2.1 (Wronskian)**.** *For n univariate functions $f_1, \dots, f_n$, which are $n-1$ times differentiable, the Wronskian $Wr(f_1, \dots, f_n)$ is defined as*

$$Wr(f_1, \dots, f_n)(x) = \begin{vmatrix} f_1(x) & f_2(x) & \dots & f_n(x) \\ f_1'(x) & f_2'(x) & \dots & f_n'(x) \\ \vdots & \vdots & \ddots & \vdots \\ f_1^{(n-1)} & f_2^{(n-1)} & \dots & f_n^{(n-1)} \end{vmatrix}$$

The basic relation of the Wronskian to linear independence is that for any linearly dependent functions $f_1, \dots, f_n$, the Wronskian $\mathrm{Wr}(f_1, \dots, f_n)$ vanishes everywhere. The converse is false in general, first Peano and then Bôcher found counterexamples (see [25] for a history of these results). However, several sufficient conditions to ensure that the vanishing of the Wronskian everywhere implies linear dependence were found. For instance, Bôcher proved [15] that if the f_i's are analytic, then the converse holds. In particular, the Wronskian captures the linear dependence of polynomials in $\mathbb{F}[x]$.

Proposition 2.2.2. *[12] For $f_1, \dots, f_n \in \mathbb{F}[x]$, the functions are linearly dependent if and only if the Wronskian $Wr(f_1, \dots, f_n)$ vanishes everywhere.*

Let us illustrate an example of linear independence that can be proved using the Wronskian.

Proposition 2.2.3 (Folklore)**.** *For any integer d, for any distinct $(a_i) \in \mathbb{F}^{d+1}$, the set $S = \{(x-a_0)^d, \dots, (x-a_d)^d\}$ is a basis of $\mathbb{F}_d[x]$, where $\mathbb{F}_d[x]$ denotes the vector space of polynomials of degree at most d.*

Proof. Since $\dim \mathbb{F}_d[x] = d+1 = |S|$, we only have to show that S is linearly independent. Consider the Wronskian of the polynomials in S:

$$\mathrm{Wr}(x) = \mathrm{Wr}((x-a_0)^d, \dots, (x-a_d)^d) = \begin{vmatrix} (x-a_0)^d & \dots & (x-a_d)^d \\ d(x-a_0)^{d-1} & \dots & d(x-a_d)^{d-1} \\ \vdots & \ddots & \vdots \\ d! & \dots & d! \end{vmatrix}$$

It's enough to show that the Wronskian is not the null polynomial. In fact, we will show that it's a (non-zero) constant polynomial. For any $z \in \mathbb{F}$, define $b_i = z - a_i$ and we have:

$$\mathrm{Wr}(z) = \begin{vmatrix} b_0^d & \dots & b_d^d \\ d \cdot b_0^{d-1} & \dots & d \cdot b_d^{d-1} \\ \vdots & \ddots & \vdots \\ d! & \dots & d! \end{vmatrix} = c \cdot \begin{vmatrix} b_0^d & \dots & b_d^d \\ b_0^{d-1} & \dots & b_d^{d-1} \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{vmatrix}$$

for some non-zero $c \in \mathbb{N}^*$ which only depends on d. The last matrix is a Vandermonde matrix, so its determinant is equal to the product $\prod_{i<j}(b_i - b_j) = \prod_{i<j}(a_j - a_i)$, which is a non-zero constant since all the a_i's are distinct. The determinant is hence non-zero and so we have $\mathrm{Wr}(z) \neq 0$, thus the family S is linearly independent. □

In algebraic complexity, this tool was already used to establish a bound in [52] for sums of products of powers of sparse polynomials. The authors used some results from [72] that give a link between the number of roots of polynomials of the form $f = \sum_{i=1}^n f_i$ and the Wronskian $\mathrm{Wr}(f_1, \dots, f_n)$. In the following, we will use the fact that the Wronskian is a determinant and therefore inherits its properties. In particular, it can be factorized along its columns or rows. As the following result shows, this will be useful in our model where we have polynomials with factors of high multiplicity.

Proposition 2.2.4. *Let $f_1, \dots, f_n \in \mathbb{F}[x]$ be linearly independent polynomials and let $a \in \mathbb{F}$. If $f_j = Q_j^{e_j} g_j$ with $Q_j, g_j \in \mathbb{F}[x]$, then $Q \stackrel{def}{=} \prod_{j=1}^n Q_j^{d_j}$ divides Wr$(f_1, \dots, f_n)$, with $d_j = \max(0, e_j - n + 1)$. Moreover, we have Wr$(f_1, \dots, f_n) = Q(X)\,P(X)$ with $P \in \mathbb{F}[x]$ such that*

$$\deg(P) \leq \sum_{j=1}^n \left[\deg(g_j) + (n-1)\deg(Q_j)\right] - \binom{n}{2}.$$

Proof. Consider the $n \times n$ Wronskian matrix W whose (i,j)-th entry is $f_j^{(i-1)}(x)$ with $i, j \in [\![1, n]\!]$. Let $i \in [\![1, n]\!]$ such that $e_j \geq n$. Since $Q_j^{e_j}$ divides f_j, then $f_j^{(i)} = Q_j^{e_j - i}\, g_{i,j} = Q_j^{e_j - n + 1}\, Q_j^{n-1-i}\, g_{i,j}$, for some $g_{i,j} \in \mathbb{F}[x]$ of degree $\deg(g_j) + i\deg(Q_j) - i$. Since $Q_j^{e_j - n + 1}$ divides every element in the j-th column of W, we can factor it out from the Wronskian. This proves that Q divides $\mathrm{Wr}(f_1, \dots, f_n)$. Once we have factored out $Q_j^{e_j - n + 1}$ for all j, we observe that $\mathrm{Wr}(f_1, \dots, f_n) = Q(x)\,P(x)$, where $h(x)$ is the determinant of a matrix whose (i,j)-th entry has degree $\deg(g_j) + (n-1)\deg(Q_j) - (i+1)$ for all $i, j \in [\![1, n]\!]$. Hence, $\deg(h) \leq \sum_{j=1}^n \left[\deg(g_j) + (n-1)\deg(Q_j)\right] - \binom{n}{2}$. □

The following result is an analogue of Theorem 2.1.1 that holds for polynomials with coefficients over any field $\mathbb{F}$ of characteristic zero, yet with a bound weaker than the one in Theorem 2.1.1.

Theorem 2.2.5. *Consider a polynomial identity of the form:*

$$\sum_{i=1}^{k} \alpha_i (x - a_i)^d = \sum_{i=1}^{l} \beta_i (x - b_i)^{e_i}$$

where the $a_i \in \mathbb{F}$ are distinct, the $\alpha_i \in \mathbb{F}$ are not all zero, $\beta_i, b_i \in \mathbb{F}$ are arbitrary, and $e_i < d$ for every i. Then we must have $k + l > \sqrt{2(d+1)}$.

Proof. We assume $\alpha_1 \neq 0$ and we have the following equality:

$$\alpha_1 (x - a_1)^d = -\sum_{i=2}^{k} \alpha_i (x - a_i)^d + \sum_{i=1}^{l} \beta_i (x - b_i)^{e_i}$$

Consider an independent subfamily on the right hand side of this equality. We obtain a new identity of the form:

$$g = \sum_{i=1}^{p} \lambda_i \ell_i^{r_i}$$

with $g(x) = \alpha_1 (x - a_1)^d$, and $p \leq k + l - 1$. Since $\deg(g) = d$, then there exists i such that $r_i = d$. Moreover, since $e_j < d$ for all j, we assume without loss of generality that $\ell_1 = x - a_2$ and $r_1 = d$. By properties of the determinant, we have:

$$0 \neq \mathrm{Wr}(\lambda_1 \ell_1^{r_1}, \ell_2^{r_2}, \ldots, \ell_p^{r_p}) = \mathrm{Wr}(g, \ell_2^{r_2}, \ldots, \ell_p^{r_p})$$

We define $\Delta = \{i : 2 \leq i \leq p, r_i \geq p\}$ and, following Proposition 2.2.4, we factorise the Wronskians:

$$\begin{cases} \mathrm{Wr}(g, \ell_2^{r_2}, \ldots, \ell_p^{r_p}) & = (x - a_1)^{d-(p-1)} \prod_{i \in \Delta} \ell_i^{r_i - (p-1)} \cdot W_1 \\ \mathrm{Wr}(\lambda_1 \ell_1^{r_1}, \ell_2^{r_2}, \ldots, \ell_p^{r_p}) & = (x - a_2)^{d-(p-1)} \prod_{i \in \Delta} \ell_i^{r_i - (p-1)} \cdot W_2 \end{cases}$$

where $W_1, W_2 \in \mathbb{F}[x]$ are the remaining determinants whose degrees are upper bounded by $p(p-1)/2$ according to Proposition 2.2.4. After some simplifications, we obtain the following identity:

$$(x - a_2)^{d-(p-1)} W_2 = (x - a_1)^{d-(p-1)} W_1$$

Since $a_1 \neq a_2$, then $(x - a_1)^{d-(p-1)}$ must divide W_2 and therefore we should have

$$d - (p-1) \leq \frac{p(p-1)}{2}$$

Finally, we set $s = l + k$ and we use the fact that $p \leq s - 1$ to obtain the desired lower bound:

$$d \leq \frac{(p+2)(p-1)}{2} \leq \frac{(s+1)(s-2)}{2},$$

and finally, $2(d+1) < s^2$. □

Remark 2.2.6. *Example 2.0.1 shows that the order of this bound is tight when $\mathbb{F} = \mathbb{C}$, the field of complex numbers. Indeed, choosing $k = \sqrt{d+1}$ leads to the equality*

$$\sum_{i=1}^{k}(x+\xi^i)^d = \sum_{j=0}^{k-1} k\binom{d}{jk}x^{d-jk}$$

which has $2k = 2\sqrt{d+1}$ terms.

2.2.2 Uniqueness and field extension

As a consequence of Theorem 2.2.5 we obtain that whenever $\mathrm{AffPow}_{\mathbb{F}}(f)$ is sufficiently small, the terms of highest degree in an optimal expression of f as $f = \sum_{i=1}^{s}\alpha_i(x-a_i)^{e_i}$ are uniquely determined.

Corollary 2.2.7. *Let $f \in \mathbb{F}[x]$ be a polynomial of the form :*

$$f = \sum_{i=1}^{k}\alpha_i(x-a_i)^d + \sum_{j=1}^{l}\beta_j(x-b_j)^{e_j}$$

with $e_j < d$. If $k + l \leq \sqrt{\frac{d+1}{2}}$, then the highest degree terms are unique. In other words, for every expression of f as

$$f = \sum_{i=1}^{k'}\alpha'_i(x-a'_i)^d + \sum_{j=1}^{l'}\beta'_j(x-b'_j)^{e'_j}$$

with $e'_j < d$ and $k' + l' \leq \sqrt{\frac{d+1}{2}}$, then $k = k'$ and there exists a permutation $\pi \in \mathfrak{S}_k$ such that $\alpha_i = \alpha'_{\pi(i)}$ and $a_i = a'_{\pi(i)}$ for all $i \in [\![1, k]\!]$.

Proof. Let us assume that we have another different decomposition for f:

$$f = \sum_{i=1}^{k'}\alpha'_i(x-a'_i)^d + \sum_{j=1}^{l'}\beta'_j(x-b'_j)^{e'_j}$$

with $k' + l' \leq \sqrt{(d+1)/2}$. Hence, we have the following equality:

$$\sum_{i=1}^{k}\alpha_i(x-a_i)^d - \sum_{i=1}^{k'}\alpha'_i(x-a'_i)^d = \sum_{j=1}^{l}\beta_j(x-b_j)^{e_j} - \sum_{j=1}^{l'}\beta'_j(x-b'_j)^{e'_j}$$

Since $k + k' + l + l' \leq \sqrt{2(d+1)}$, the result follows from Theorem 2.2.5. □

Finally, as a direct consequence of Corollary 2.2.7, we obtain a sufficient condition for a polynomial to have a unique optimal expression in Model 2.

Corollary 2.2.8. *Let $f \in \mathbb{F}[x]$ be a polynomial of the form:*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}$$

If $n_e \stackrel{def}{=} \#\{i : e_i < e\} \leq \sqrt{\frac{e}{2}}$ for all $e \in \mathbb{N}$, then $\mathrm{AffPow}_{\mathbb{F}}(f) = s$ and the optimal representation of f is unique.

Whenever $f \in \mathbb{R}[x]$ satisfies the hypotheses of Corollary 2.2.8 and one term in the expression of f is of the form $\alpha_i(x - a_i)^{e_i}$ with $a_i \in \mathbb{C} - \mathbb{R}$, then there exists $j \neq i$ such that $\alpha_j = \overline{\alpha_i}, a_j = \overline{a_i}$ and $e_j = e_i$. Indeed, if we have a decomposition for f, taking the conjugate of α_i and a_i for all i gives another decomposition of f, but by Corollary 2.2.8 these two decompositions must be identical. We now prove a more general version of this fact.

Proposition 2.2.9. *Let $\mathbb{K}$ be a subfield of $\mathbb{F}$. Let $f \in \mathbb{K}[x]$ be a polynomial that can be expressed in the $\mathrm{AffPow}_{\mathbb{F}}$ model as*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i} \; with \; \alpha_i, a_i \in \mathbb{F},$$

and $n_e \leq \sqrt{\frac{e}{2}}$ for all $e \in \mathbb{N}$. Then, for all $e \in \mathbb{N}$, the truncated expression

$$\widetilde{f} = \sum_{e_i = e} \alpha_i (x - a_i)^{e_i}$$

belongs to $\mathbb{K}[x]$.

Proof. By Corollary 2.2.8, we know that $\mathrm{AffPow}_{\mathbb{F}}(f) = s$ and, hence, α_i, a_i are algebraic over $\mathbb{K}$. We denote by $\mathbb{T}$ the splitting field of the minimal polynomials of all the α_i, a_i over $\mathbb{K}$ (i.e., the smallest field $\mathbb{T}$ such that $\mathbb{K}(\alpha_i, a_i) \subset \mathbb{T}$ and $\mathbb{K} \subset \mathbb{T}$ is normal). Since $\mathbb{K}$ is of characteristic 0 (and, thus, the extension $\mathbb{K} \subset \mathbb{T}$ is separable), then $\mathbb{K} \subset \mathbb{T}$ is a Galois extension.
Take now σ any element of the Galois group of the extension $\mathbb{K} \subset \mathbb{T}$. Since $f \in \mathbb{K}[x]$, if we apply σ to f we obtain that $f = \sigma(f) = \sum_{i=1}^{s} \sigma(\alpha_i)(x - \sigma(a_i))^{e_i}$. Moreover, by Corollary 2.2.8, we know that $\mathrm{AffPow}_{\mathbb{T}}(f) = s$ and f has a unique optimal expression in the $\mathrm{AffPow}_{\mathbb{T}}$ model, then $\{(\alpha_i, a_i, e_i) \,|\, 1 \leq i \leq s\} = \{(\sigma(\alpha_i), \sigma(a_i), e_i) \,|\, 1 \leq i \leq s\}$. In particular, for every $e \in \mathbb{N}$, we have that

$$\{(\alpha_i, a_i, e_i) \,|\, e_i = e\} = \{(\sigma(\alpha_i), \sigma(a_i), e_i) \,|\, e_i = e\}. \tag{2.4}$$

Now, we consider $\widetilde{f} = \sum_{e_i = e} \alpha_i (x - a_i)^{e_i}$, by (2.4) we get that

$$\sigma(\widetilde{f}) = \sum_{e_i = e} \sigma(\alpha_i)(x - \sigma(a_i))^{e_i} = \sum_{e_i = e} \alpha_i (x - a_i)^{e_i} = \widetilde{f}.$$

Summarizing, if we denote $\widetilde{f} = \sum_{i=0}^{e} f_i x^i \in \mathbb{T}[x]$, we have proved that $\sigma(f_i) = f_i$ for every $i \in [\![0, e]\!]$ and every σ in the Galois group of the extension $\mathbb{K} \subset \mathbb{T}$. This proves (see, e.g., [21, Theorem 7.1.1]) that $f_i \in \mathbb{K}$ for all $i \in [\![0, e]\!]$ and therefore $\widetilde{f} \in \mathbb{K}[x]$. □

2.2.3 Largest exponent in optimal expressions

It is not clear at first that the exponents involved in an optimal expression of f in Model 2 are bounded. Indeed, even though the trivial decomposition (see Proposition 1.1.17) involves only exponents smaller or equal than the degree of the polynomial, some polynomials require larger exponents in their optimal expressions, such as $f = (x+1)^{d+1} - x^{d+1}$ which is optimal by Corollary 2.2.8. We now prove, using Theorem 2.2.5, that the exponents in an optimal expression are upper bounded in terms of d.

Corollary 2.2.10. *Let $f \in \mathbb{F}[x]$ be a polynomial of degree d written as*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}$$

with $\alpha_i, a_i \in \mathbb{F}$, $e_i \in \mathbb{N}$. We set $e \stackrel{def}{=} \max\{e_i : i \in [\![1, s]\!]\}$. Then we have that

$$e < d + \frac{s^2}{2}$$

In particular, if $s = \mathit{AffPow}_{\mathbb{F}}(f)$, then $e < d + \frac{(d+2)^2}{8}$.

Proof. If $e = d$, then the result is trivial. Assume therefore that $e > d$. Now, we differentiate $d+1$ times the expression for f to obtain the identity:

$$0 = f^{(d+1)} = \sum_{e_i > d} \alpha_i \frac{e_i!}{(e_i - d - 1)!} (x - a_i)^{e_i - d - 1}.$$

By Theorem 2.2.5 we have $s > \sqrt{2(e-d)}$ and we conclude that $e < d + \frac{s^2}{2}$. To finish the proof it suffices to recall that $s = \mathrm{AffPow}_{\mathbb{F}}(f) \leq \lceil (d+1)/2 \rceil \leq (d+2)/2$ by Proposition 1.1.17. □

Remark 2.2.11. *When $\mathbb{F} = \mathbb{R}$, we can use Theorem 2.1.1 to obtain a better upper bound on e: we have*

$$\left\lceil \frac{d+1}{2} \right\rceil \geq s \geq \left\lceil \frac{e-d+2}{2} \right\rceil,$$

and therefore $e \leq 2d$.

Remark 2.2.12. *One can find examples that are close to the bound obtained in Corollary 2.2.10. Indeed, if we take $k = \sqrt{d+1}$ in Example 2.0.1, we get an expression of the 0 polynomial with $2k$ terms, namely:*

$$\sum_{j=1}^{k} (x + \xi^j)^d - \sum_{\substack{0 \leq i \leq d \\ i \equiv 0 \pmod{k}}} k \binom{d}{i} x^{d-i} = 0$$

If we integrate this expression $7(d+1)$ times we get a polynomial

$$f := \frac{d!}{(8d+7)!}\sum_{j=1}^{k}(x+\xi^j)^{8d+7} - \sum_{\substack{0\leq i\leq d\\ i\equiv 0 \pmod k}} k\binom{d}{i}\frac{(d-i)!}{(8d+7-i)!}\,x^{8d+7-i},$$

of degree $< 7(d+1)$ with $s := \mathit{AffPow}_{\mathbb{F}}(f) = 2k$ (by Corollary 2.2.8) and whose maximum exponent in the optimal expression is $8d+7 = 7(d+1)+d > \deg(f) + (s^2-4)/4$.

Remark 2.2.13. *As a consequence of Corollary 2.2.10, we obtain a naive brute force algorithm to find one optimal expression for any polynomial f. Indeed, for a fixed integer s, there are only a finite number of sequences of exponents $(e_1,\dots,e_s)$ with $e_i \leq d + s^2/2$. For one sequence, one can try to find an expression with these exponents by solving a system of polynomial equations in $2s$ variables. The smallest s with a solution gives the value of $\mathit{AffPow}_{\mathbb{F}}(f)$.*

Also, as a byproduct of Corollary 2.2.10, we obtain the exact value of $\mathrm{AffPow}_{\mathbb{F}}(f)$ for a generic polynomial f of degree d. It turns out to be equal to the worst case value of $\mathrm{AffPow}_{\mathbb{F}}(f)$, obtained in [29, Proposition 18].

Corollary 2.2.14. *For a generic polynomial $f \in \mathbb{F}[x]$ of degree d, we have that $\mathit{AffPow}_{\mathbb{F}}(f) = \lceil \frac{d+1}{2} \rceil$.*

Proof. The set of polynomials of degree $\leq d$ can be seen as a linear space W of dimension $d+1$. Given $f \in \mathbb{F}[x]$ a polynomial of degree d, by Proposition 1.1.17 we have $\mathrm{AffPow}_{\mathbb{F}}(f) \leq \lceil \frac{d+1}{2} \rceil$. For $k < \lceil \frac{d+1}{2} \rceil$, let us show that the set of polynomials g of degree d such that $\mathrm{AffPow}_{\mathbb{F}}(g) \leq k$ is contained in a variety of dimension $2k < d+1$. For every $e_1,\dots,e_k \in \mathbb{N}$ the set of polynomials that can be written as $\sum_{i=1}^{k}\alpha_i(x-a_i)^{e_i}$ with $a_i,\alpha_i \in \mathbb{F}$ is contained in a variety $V_{e_1,\dots,e_k}$ of dimension $2k$ If we set $M := d + \frac{(d+2)^2}{8}$, Corollary 2.2.10 proves that in every optimal expression of a polynomial of degree d, the exponents e_i are $\leq M$; thus the set of polynomials with $\mathrm{AffPow}_{\mathbb{F}}(f) \leq k$ and degree d is contained in $\bigcup_{e_i\leq M} V_{e_1,\dots,e_k}$, which is a variety of dimension $\leq 2k$ (it is a finite union of varieties of dimension $\leq 2k$). □

2.2.4 Orthogonality

By definition of the three models, we directly have $\mathrm{AffPow}_{\mathbb{F}}(f) \leq \mathrm{Waring}_{\mathbb{F}}(f)$ and $\mathrm{AffPow}_{\mathbb{F}}(f) \leq \mathrm{Sparsest}_{\mathbb{F}}(f)$ for any polynomial $f \in \mathbb{F}[x]$. We now exhibit some polynomials f such that $\mathrm{AffPow}_{\mathbb{F}}(f)$ is much smaller than both $\mathrm{Waring}_{\mathbb{F}}(f)$ and $\mathrm{Sparsest}_{\mathbb{F}}(f)$.

Example 2.2.15. *For every $d \in \mathbb{N}$, we consider the polynomial $f_d \in \mathbb{C}[x]$ given by $f_d \stackrel{def}{=} (x+1)^d - dx^{d-1}$. It is easy to check that $\mathit{AffPow}(f_d) = 2$ for all $d \geq 2$. By [8, Proposition 3.1] we have that if $x^{d-1} = \sum_{i=1}^{s}\alpha_i(x-a_i)^d$ with $\alpha_i, a_i \in \mathbb{C}$, then $s \geq d$; and thus we get that $\mathit{Waring}_{\mathbb{C}}(f_d) \geq d-1$.*

One can easily check that for every $i \in [\![0, d-1]\!]$, the polynomials $f_d^{(i)} = \frac{d!}{(d-i)!} f_{d-i}$ and $f_d^{(i+1)} = \frac{d!}{(d-i-1)!} f_{d-i-1}$ do not share a common root. Consider a decomposition of f in the sparsest shift model. By Remark 1.2.4, for any pair of consecutive coefficients in this decomposition at least one of the 2 coefficients is nonzero. This implies that Sparsest$_{\mathbb{C}}(f) \geq \lceil (d+1)/2 \rceil$.

We now give (in Proposition 2.2.17) a weaker version of Corollary 2.1.5 that works for any field of characteristic zero. Moreover, for $\mathbb{F} = \mathbb{C}$ we provide a family of polynomials showing that the bound from Proposition 2.2.17 is sharp. We will use Jordan's lemma [34] (see [38, Lemma 1.35] for a recent reference), which can be restated as follows.

Lemma 2.2.16 (Jordan's lemma)**.** *Let $d \in \mathbb{Z}^+$, $e_1, \ldots, e_t \in \{1, \ldots, d\}$, and let $a_1, \ldots, a_t \in \mathbb{F}$ be distinct constants. If $\sum_{i=1}^{t} (d+1-e_i) \leq d+1$, then the set of polynomials*

$$\bigcup_{i=1}^{t} \{(x-a_i)^e \,:\, e_i \leq e \leq d\}$$

is linearly independent.

Proposition 2.2.17. *Let $f \in \mathbb{F}[x]$ be a polynomial of degree d. Either $f = \alpha(x-a)^d$ for some $\alpha, a \in \mathbb{F}$ (and Waring$_{\mathbb{F}}(f) =$ Sparsest$_{\mathbb{F}}(f) = 1$), or the following holds:*

$$\textit{Waring}_{\mathbb{F}}(f) \cdot \textit{Sparsest}_{\mathbb{F}}(f) \geq d+1$$

Proof. We set $k = \text{Waring}_{\mathbb{F}}(f)$ and $l = \text{Sparsest}_{\mathbb{F}}(f)$ and assume that $k, l \geq 2$. We express f in two different ways:

$$f = \sum_{i=1}^{k} \alpha_i (x-a_i)^d = \sum_{j=1}^{l} \beta_j (x-a)^{e_j},$$

with $a_j \in \mathbb{F}$ all distinct and $e_0 := -1 < e_1 < \cdots < e_l = d$. First, we are going to prove that $e_{i+1} - e_i \leq k$ for all $i \in [\![0, l-1]\!]$. Indeed, if there exists $t \in [\![0, l-1]\!]$ such that $e_{t+1} - e_t \geq k+1$, then we set $r := e_t + 1$ and differentiate the previous equality r times to obtain

$$f^{(r)} = \sum_{i=1}^{k} \alpha_i \frac{d!}{(d-r)!} (x-a_i)^{d-r} = \sum_{j=t+1}^{l} \beta_j \frac{e_j!}{(e_j-r)!} (x-a)^{e_j-r},$$

where $e_j - r = e_j - e_t - 1 \geq e_{t+1} - e_t - 1 \geq k$ for all $j \in \{t+1, \ldots, l\}$. From this equality, we deduce that the set

$$\mathcal{B} := \{(x-a_i)^{d-r} \,|\, 1 \leq i \leq k\} \cup \{(x-a)^{e_i-r} \,|\, t+1 \leq i \leq l\}$$

is linearly dependent. However,

$$\mathcal{B} \subseteq \{(x-a_i)^{d-r} \,|\, 1 \leq i \leq k\} \cup \{(x-a)^i \,|\, k \leq i \leq d-r\}.$$

By Lemma 2.2.16, the $d - r + 1$ polynomials on the right-hand side are linearly independent. This is a contradiction since $\mathcal{B}$ is linearly dependent. We have proved that $e_{i+1} - e_i \leq k$ for all $i \in [\![0, l-1]\!]$, and we conclude that

$$d + 1 = e_l - e_0 = \sum_{i=1}^{l} (e_i - e_{i-1}) \leq kl.$$

□

Remark 2.2.18. *Example 2.0.1 shows that there are polynomials of degree d such that Waring$_{\mathbb{C}}(g) \leq k$ and Sparsest$_{\mathbb{C}}(g) \leq \lceil (d+1)/k \rceil$ and, thus the bound from Proposition 2.2.17 is tight.*

3

Lower bounds and linear independence

Circuit lower bounds against a class of circuits $\mathcal{C}$ are often obtained following the same pattern of "natural" proof (outlined in [47]):

Step 1: (normal form) For every circuit in the circuit class C of interest, express the polynomial computed as a *small sum of simple building blocks*.

Step 2: (complexity measure) Build a map $\Gamma : \mathbb{F}[x] \to \mathbb{Z}^+$ that is *sub-additive*, i.e. $\Gamma(f+g) \leq \Gamma(f) + \Gamma(g)$.

Step 3: (potential usefulness) Show that if B is a simple building block, then $\Gamma(B)$ is small. Further, check that $\Gamma(f)$ is large for a random polynomial f.

Step 4: (explicit lower bound) Find an explicit polynomial f for which $\Gamma(f)$ is large.

For arithmetic circuits, one of the most successful complexity measures is based on partial derivatives. One takes as complexity measure $\dim \partial^{=k} f$, where $\partial^{=k} f$ denotes the linear space of polynomials spanned by the partial derivatives of f of order k. This method already yields lower bounds for Model 1. Indeed, the derivatives of order k of a d-th power of an affine form $\ell(x_1, \ldots, x_n)$ are constant multiples of ℓ^{d-k} for all $k \leq d$. Therefore, by linearity of derivatives we have for any k the lower bound $\text{AffPow}(f) \geq \dim \partial^{=k} f$ on the AffPow rank of f. By taking a polynomial with big partial derivatives space, we obtain the following lower bound.

Theorem 3.0.1. *Let $f = \prod_{i=1}^{n} x_i$. Then AffPow$(f) \geq \frac{2^n - 1}{n}$*

Proof. For any $k \in [\![1, n]\!]$, all the non-zero derivatives of f of order k are linearly independent, we therefore have $\dim \partial^{=k} f = \binom{n}{k}$. Summing this equality for $k = 1, \ldots, n$, then dividing by n, gives the lower bound. □

The method of partial derivatives was introduced in the complexity theory litterature by Nisan and Wigderson [61], where lower bounds were given for more powerful models than Model 2 such as e.g. depth 3 arithmetic circuits. In such a circuit, the powers in Model 2 are replaced by products of d affine functions. We then have [61] the lower bound $r \geq (\dim \partial^* f)/2^d$, where r denotes as in Model 2 the fan-in of the circuit's output gate and $\partial^* f$ denotes the space spanned by partial derivatives of all order. More recently, a number of new lower bound results were obtained using a refinement of the method of partial derivatives. These new results are based on "shifted partial derivatives", introduced first in [41] to prove an exponential lower bound for sums of powers of bounded degree polynomials. More precisely, for a set $S \subseteq \mathbb{F}[x_1, \ldots, x_n]$, let $x^{\leq l} \cdot S$ denote the $\mathbb{F}$-span of products QR with $\deg(Q) \leq l$ and $R \in S$. The complexity measure of a polynomial f is then defined as the dimension of the vector space spanned by $x^{\leq l} \cdot \partial^{\leq k} f$.

In a attempt to find an univariate version of this method to prove lower bounds for our univariate settings, we defined in [42] the following space of shifted derivatives:

$$\left\langle x^{\leq i+l} \cdot f^{(i)} \right\rangle_{i \leq k} \stackrel{\text{def}}{=} \mathbb{F}\text{-span}\left\{ x^j \cdot f^{(i)}(x) \quad : \quad i \leq k,\ j \leq i + l \right\}$$

Using the complexity measure given by the dimension of this space, we proved in [42, Theorem 2] a lower bound for Model 4. In the following we will first give a simpler proof for Model 2 by exhibiting some polynomials that have a full shifted derivatives space. To do so, we relate the dimension of this space to some differential equations that f must satisfy.

3.1 Shifted Differential Equations

3.1.1 Definition

In this section, we introduce the main tool of this thesis: linear homogeneous differential equations with polynomial coefficients that satisfy some degree constraints. Although general solutions to these equations are known are *D-finite* functions, we will focus here on polynomial solutions. Throughout our papers [42, 30, 31, 32], we used several definitions of this notion depending on the properties we need them to satisfy. In the following, we give the most general version and discuss a few choices of interest of the parameters.

Definition 3.1.1. *A* Shifted Differential Equation *(SDE) of parameters $t, k, l \in \mathbb{N}$, with $t \leq k+1$, is a differential equation of the form*

$$\sum_{i=0}^{k} P_i(x) f^{(i)}(x) = 0 \tag{3.1}$$

where f is the unknown function and the P_i are polynomials in $\mathbb{F}[x]$ with $\deg(P_i) \leq i + l$ for all $i \in [\![0, t-1]\!]$, $\deg(P_i) \leq l$ for all $i \in [\![t, k]\!]$ and $P_k \neq 0$. We refer to the polynomials P_i as the coefficients *of the SDE.*
The quantity k is called the order *of the equation, and the quantity l is called the* shift*. We will usually denote such a differential equation by SDE(t, k, l).*

For conciseness, we also define a few other notations. Given a differential equation of the form (3.1) that satisfies the same conditions except that P_k might be 0, if one sets $k' \stackrel{\text{def}}{=} \max\{i | P_i \neq 0\}$ and $t' \stackrel{\text{def}}{=} \min(t, k'+1)$, then this differential equation is an SDE(t', k', l). In the following, we will use the notation SDE$_{\leq}(t, k, l)$ to describe such a differential equation. We will denote by $D(t, k, l)$ the set of pairs $(i, j) \in \mathbb{N}^2$ such that $i < t$ and $j \leq i + l$, or $t \leq i \leq k$ and $j \leq l$ (see Figure 3.1 for an example).

Remark 3.1.2. *A polynomial f satisfies a SDE$_{\leq}(t, k, l)$ if and only if the polynomials $(x^j f^{(i)}(x))_{(i,j) \in D(t,k,l)}$ are linearly dependent over $\mathbb{F}$. The existence of such a SDE can therefore be decided efficiently by linear algebra, and when a SDE$_{\leq}(t, k, l)$ exists it can be found explicitly by solving the corresponding linear system (see, e.g., [65, Corollary 3.3a] for an analysis of linear system solving in the bit model of computation). We use this fact repeatedly to design the reconstruction algorithms of Chapter 4.*

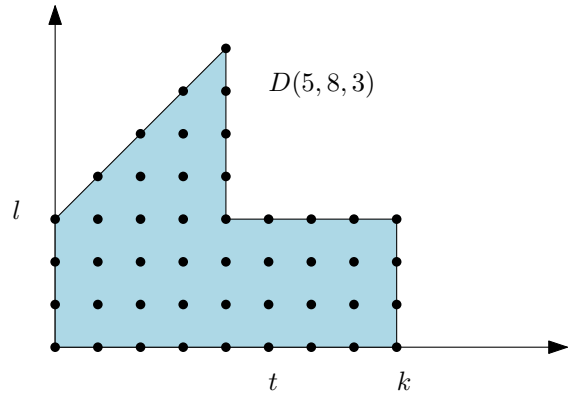

Figure 3.1: Domain $D(5,8,3)$

As an example, the affine power $(x-a)^e$ satisfies the SDE:

$$(x-a)f' - e.f = 0,$$

of order 1 and shift 1 with either $t=0$ or $t=1$ (this differential equation can also be seen as a SDE of order 1 and shift 0 with $t=2$). We generalize this observation to several affine powers in the following result, which is one of the main motivations for defining such differential equations.

Proposition 3.1.3. *Let* $F = \{(x-a_i)^{e_i} \ : \ a_i \in \mathbb{F}, e_i \in \mathbb{N},\ 1 \leq i \leq s\}$. *Then for any choice of parameters* (t,k,l) *such that*

$$s(l+k+1) < (k+1)(l+1) + \binom{t}{2} = |D(t,k,l)| \tag{3.2}$$

there exists a SDE$_{\leq}(t,k,l)$ *satisfied simultaneously by the* $f_i(x) = (x-a_i)^{e_i}$ *for* $i = 1, \ldots, s$.

Proof. The existence of this common SDE is equivalent to the existence of a common nonzero solution for the following equations $(E_r)_{1\leq r\leq s}$ in the unknowns $\lambda_{i,j}$:

$$\sum_{i=0}^{t-1} \left(\sum_{j=0}^{i+l} \lambda_{i,j} x^j \right) f_r^{(i)}(x) + \sum_{i=t}^{k} \left(\sum_{j=0}^{l} \lambda_{i,j} x^j \right) f_r^{(i)}(x) = 0 \qquad (E_r)$$

Therefore, there are $(k+1)(l+1) + \frac{t(t-1)}{2}$ unknowns, so we need to show that the matrix of this linear system has rank smaller than $(k+1)(l+1) + \frac{t(t-1)}{2}$. We are going to show that for each fixed value of $r \in \{1, \ldots, s\}$, the subsystem (E_r) has a matrix of rank $\leq l+k+1$. In other words, we have to show that the subspace V_r has dimension less than $l+k+1$, where V_r is the linear space spanned by the

polynomials $x^j f_r^{(i)}(x)$, with $(i,j) \in D(t,k,l)$. But V_r is included in the subspace spanned by the polynomials

$$\{(x-a_r)^{e_r+j};\ -k \le j \le l,\ e_r + j \ge 0\}.$$

This is due to the fact that the polynomials x^i belong to the span of the polynomials $\{(x-a_r)^\ell\ :\ 0 \le \ell \le i\}$. Hence, we have that $\dim V_r \le l+k+1$. Since the (E_r) subsystem has a matrix of rank $\le l+k+1$, the whole system has rank $\le s(l+k+1)$. Thus, there exists a nonzero solution if $(k+1)(l+1) + \frac{t(t-1)}{2} > s(l+k+1)$. □

As an SDE is a particular case of a linear homogeneous differential equation, it inherits the following property.

Lemma 3.1.4. *The set of polynomial solutions of a SDE of order k is a vector space of dimension at most k.*

As a consequence, if we assume that the elements of F are linearly independent, then they cannot be part of the solution set of a differential equation of order strictly less than s. Thus the differential equation we obtain from the previous proposition is a SDE(t,k',l) with $s \le k' \le k$. Notice also that the order and the shift play a symmetric role for the existence of a SDE as Equation (3.2) is symmetric in k and l. Depending on the application, we will usually set the parameter t to the following values.

$\boxed{t=k+1:}$ when designing algorithms, we will look out for small SDEs in terms of order and shift. To lower the constraints on k and l in the inequality of Proposition 3.1.3, we better take t as large as possible. We will therefore choose $t = k+1$ to rewrite Equation (3.2) as

$$s(l+k+1) < (k+1)\left(l + \frac{k}{2} + 1\right).$$

In particular, the choice of parameters $l = 0, k = 2s-1$ ensures the existence of a SDE of order $k' \le k$ with $\deg(P_i) \le i$ for all i, that is satisfied simultaneously by all the elements of F. For conciseness, we will use the notation SDE(k,l) to denote a SDE$(k+1,k,l)$ throughout this thesis (mostly in Chapter 4).

$\boxed{t=s:}$ to prove the linear independence results of Section 3.3, we will need to control the degree of the last coefficient $P_{k'}$. Whenever the set F is linearly independent, then we have $s \le k' \le k$ and thus choosing $t = s$ ensures that $\deg(P_{k'}) \le l$.

3.1.2 Roots of coefficients of a differential equation

We now proceed to one of the main tools of the results of this chapter. We show that a differential equation with polynomial coefficients satisfied by every element of a family F of shifted powers must have some structure in the roots of its coefficients. In this section we will use the convenient notation:

$$x^{\underline{i}} = x(x-1).\cdots.(x-i+1)$$

where i is a positive integer (for $i = 0$ we set $x^{\underline{i}} = 1$). When x is a nonnegative integer, we have $x^{\underline{i}} = x!/(x-i)!$ for $x \geq i$ and $x^{\underline{i}} = 0$ for $x < i$. This notation allows us to write the i-th derivative of a shifted power $f(x) = (x-a)^e$ in a concise way: $f^{(i)} = e^{\underline{i}}(x-a)^{e-i}$.

We first make the following remark, which was the starting point for the study of the roots of the coefficients of a differential equation with polynomial coefficients.

Proposition 3.1.5. *Consider the following differential equation with polynomial coefficients:*

$$\sum_{i=0}^{k} P_i(x) f^{(i)}(x) = 0.$$

Assume that $(x-a)^e$ satisfies this equation, with $e \geq k$. Then we have $P_k(a) = 0$.

Proof. Since $(x-a)^e$ is a solution, we have

$$\sum_{i=0}^{k} P_i(x) e^{\underline{i}} (x-a)^{e-i} = 0.$$

We deduce that there exists $q \in \mathbb{F}[x]$ such that $P_k(x)(x-a)^{e-k} = (x-a)^{e-k+1} q(x)$, from which we deduce that $P_k(a) = 0$. □

As a direct consequence, if $(x-a)^e$ and $(x-b)^f$ with $a \neq b$ and $e, f \geq k$ both satisfy the same equation, then both a and b are roots of P_k. However, if $(x-a)^e$ and $(x-a)^f$ both satisfy the same equation, can we say more than just $P_k(a) = 0$? We will answer positively this question thanks to the following proposition:

Proposition 3.1.6. *Let $(*)$ be the following differential equation with polynomial coefficients:*

$$\sum_{i=0}^{k} P_i(x) f^{(i)}(x) = 0 \qquad (*)$$

If $(*)$ is satisfied simultaneously by $x^{e_1}, \ldots, x^{e_n}$ where $n \leq k$ and $e_1 > e_2 > \cdots > e_n \geq k - n + 1$, then for all $m = 0 \ldots n-1$, x^{n-m} divides P_{k-m}.

Proof. By injecting x^{e_j} into $(*)$, we get the following equation:

$$\sum_{i=0}^{\min(e_j,k)} P_i(x)\ e_j^{\underline{i}}\ x^{e_j - i} = 0$$

If $e_j \leq k$, we multiply the equation by x^{k-e_j}, and otherwise, we factor out x^{e_j-k}. In both cases, we obtain:

$$\sum_{i=0}^{k} e_j^{\underline{i}}\ \cdot P_i(x) x^{k-i} = 0 \qquad (E_j)$$

We have n such equations $(E_j)_{1\leq j\leq n}$ and we will now take a "good" linear combination to deduce the result.
Fix an integer $0 \leq m < n$, and consider the n-tuple $\vec{u}_m \in \mathbb{K}^n$ such that $(\vec{u}_m)_i = 0$ for all $i \neq n-m$ and $(\vec{u}_m)_{n-m} = 1$. A "good" linear combination is a n-tuple $(\alpha_1, \ldots, \alpha_n)$ such that there exists $(b_0, \ldots, b_{k-n})$ satisfying

$$\begin{pmatrix} e_1^{\underline{0}} & \cdots & e_n^{\underline{0}} \\ e_1^{\underline{1}} & \cdots & e_n^{\underline{1}} \\ \vdots & \ddots & \vdots \\ e_1^{\underline{k-1}} & \cdots & e_n^{\underline{k-1}} \\ e_1^{\underline{k}} & \cdots & e_n^{\underline{k}} \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} b_0 \\ \vdots \\ b_{k-n} \\ \vec{u}_m \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \\ a \\ a \end{matrix}}\right\} k-n+1 \text{ rows} \\ \left.\vphantom{\begin{matrix} a \end{matrix}}\right\} n \text{ rows} \end{matrix}$$

We claim that it is always possible to find such a tuple. Assuming this fact, we then compute the following equation:

$$0 = \sum_{j=1}^{n} \alpha_j (E_j) = \sum_{i=0}^{k-n} b_i \; P_i(x) x^{k-i} + P_{k-m}(x) x^m$$

This directly implies that x^{n-m} divides $P_{k-m}(x)$.
To prove the claim, we will use the proof technique of Lemma 2 from [54] to show that the following square submatrix is invertible:

$$\begin{pmatrix} e_1^{\underline{k-n+1}} & \cdots & e_n^{\underline{k-n+1}} \\ \vdots & \ddots & \vdots \\ e_1^{\underline{k}} & \cdots & e_n^{\underline{k}} \end{pmatrix}.$$

We first factorize its determinant using the fact that $a^{\underline{b+c}} = a^{\underline{b}} \cdot (a-b)^{\underline{c}}$ to obtain

$$\begin{vmatrix} e_1^{\underline{k-n+1}} & \cdots & e_n^{\underline{k-n+1}} \\ \vdots & \ddots & \vdots \\ e_1^{\underline{k}} & \cdots & e_n^{\underline{k}} \end{vmatrix} = \prod_{i=1}^{n} e_i^{\underline{k-n+1}} \cdot \begin{vmatrix} d_1^{\underline{0}} & \cdots & d_n^{\underline{0}} \\ \vdots & \ddots & \vdots \\ d_1^{\underline{n-1}} & \cdots & d_n^{\underline{n-1}} \end{vmatrix}$$

where $d_i = e_i - k + n - 1$. Notice that the constant we have factorized is non-zero, since $e_i \geq k-n+1$. Assume for contradiction that the rows are linearly dependent. This implies that there exists a nonzero tuple (α_i) such that for all $j = 1, \ldots, n$, we have $\sum_{i=0}^{n-1} \alpha_i d_j^{\underline{i}} = 0$. In other words, if we consider the polynomial $P(x) = \sum_{i=0}^{n-1} \alpha_i x^{\underline{i}}$, then $P(d_j) = 0$ for all j. However, all the d_j's are distinct and P is of degree at most $n-1$, a contradiction. □

As a consequence, we obtain the following refinement of Proposition 3.1.5.

Corollary 3.1.7. *Let $(*)$ be the following differential equation with polynomial coefficients of order $k > 0$:*

$$\sum_{i=0}^{k} P_i(x) f^{(i)}(x) = 0 \qquad (*)$$

Consider a family $F = \{(x - a_i)^{e_i} : 1 \leq i \leq s, e_i \geq k\}$ such that $(*)$ is satisfied simultaneously by all the elements of F. Then $\prod_{i=1}^{s}(x - a_i)$ divides P_k.

In other words, each node a_i is a root of P_k of multiplicity at least equal to the number of occurrences of this node in the family F.

Proof. We partition F in subfamilies along the values of the a_i's: $F = \uplus_{i=1}^{t} F_i$ where $F_i = \{(x - b_i)^{\varepsilon_{i,j}} : 1 \leq j \leq s_i\}$, such that $b_i \neq b_j$ for $i \neq j$. Notice that $\prod_{i=1}^{s}(x - a_i) = \prod_{i=1}^{t}(x - b_i)^{s_i}$, and since $b_i \neq b_j$, it is enough to show that for $i = 1 \dots t$, we have $(x - b_i)^{s_i}$ divides P_k. This is obtained directly by using Proposition 3.1.6 with $m = 0$. □

We now remove the hypothesis of "big exponents" ($e_i \geq k$) and prove that every node in the family should appear as a root of one of the coefficients of the differential equation.

Corollary 3.1.8. *Let $(*)$ be the following differential equation with polynomial coefficients:*

$$\sum_{i=0}^{k} P_i(x) f^{(i)}(x) = 0. \qquad (*)$$

with $P_0 \neq 0$. Define $I = \{i : P_i(x) \neq 0\}$. Consider a family $F = \{(x - a_i)^{e_i} : 1 \leq i \leq s\}$ such that $(*)$ is satisfied simultaneously by all the elements of F. Then $\prod_{i=1}^{s}(x - a_i)$ divides $\prod_{i \in I} P_i$ and, for a given index i, $(x - a_i)$ will divide P_j with $j = \max\{p : p \in I, p \leq e_i\}$.

Proof. Without loss of generality, we can assume that $e_1 \geq e_2 \geq \cdots \geq e_s$, and we write $F = \{f_1, \dots f_s\}$ where $f_i = (x - a_i)^{e_i}$. We consider the last index p such that $e_p \geq k$ and partition F into two sets: $F = \{f_1, \dots, f_p\} \cup \{f_{p+1}, \dots, f_s\}$. Using Corollary 3.1.7, we get that $\prod_{i=1}^{p}(x - a_i)$ divides P_k. We now consider the following equation:

$$\sum_{i=0}^{k-1} P_i(x) f^{(i)}(x) = 0 \qquad (*')$$

Notice that for $i > p$, f_i satisfies $(*')$. Since the order of the equation has decreased, and since $0 \in I$, we can proceed by induction to obtain that $\prod_{i=p+1}^{s}(x - a_i)$ divides $\prod_{i \in I \setminus \{k\}} P_i$. The combination of these two facts yields the desired result. □

3.1.3 Smallest SDE

We now make a small parenthesis to ask the following question: does there exist a "canonical smallest" SDE satisfied by a set of affine powers $S = \{(x - a_i)^{e_i} : 1 \leq i \leq s\}$? This question has a direct algorithmic implication: if one can show that the "smallest" SDE satisfied by a polynomial $f \in \langle S \rangle$ is the "canonical" one satisfied by S, then one could just compute this SDE and find the solutions that are affine powers to find back the decomposition of f. In order to define a notion of "canonical"

equation, we first have to define an equivalence relation on SDE, as multiplying a SDE by a polynomial does not change its space of solutions.

Definition 3.1.9. *Given two differential equations with polynomial coefficients:*

$$\sum_{i=0}^{k} p_i(x)g^{(i)}(x) = 0 \quad \textit{and} \quad \sum_{i=0}^{k'} q_i(x)g^{(i)}(x) = 0,$$

we say that they are equivalent if $p_i q_j = q_i p_j$ for all $i, j \in \mathbb{N}$ with the convention $p_i = 0$ whenever $i > k$, and $q_i = 0$ whenever $i > k'$.

Notice first that in order for two differential equations to be equivalent, they must have the same order as $p_i = 0 \Leftrightarrow q_i = 0$ for all $i \in \mathbb{N}$. As a consequence, given two differential equations of order k, it is enough to test whether $p_i q_k = q_i p_k$ for all $i \in [\![0, k]\!]$ to prove that they are equivalent. The following result can be found in [63, Property 61], and we include a short proof. It states that there is a unique (up to equivalence) differential equation of minimal order satisfied by a set of linearly independent polynomials.

Lemma 3.1.10. *For any set of $\mathbb{F}$-linearly independent polynomials $f_1, \dots, f_k \in \mathbb{F}[x]$, there exists a unique differential equation with polynomial coefficients (up to equivalence) of order k satisfied simultaneously by all the f_i's.*

Proof. Suppose first that there exist two different differential equations with polynomial coefficients of order k satisfied by $f_1, \dots, f_k$, namely:

$$\sum_{i=0}^{k} p_i(x)g^{(i)}(x) = 0 \quad \text{and} \quad \sum_{i=0}^{k} q_i(x)g^{(i)}(x) = 0.$$

Then, we set $r_i := p_k q_i - q_k p_i$ for all $i \in [\![0, k]\!]$. By definition we have that $r_k = 0$ and we aim at proving that $r_i = 0$ for all i. By linearity, the following SDE

$$\sum_{i=0}^{k-1} r_i(x)\, g^{(i)}(x) = 0$$

is satisfied by $f_1, \dots, f_k$ and has order $\leq k - 1$. By Lemma 3.1.4, we must have $r_j(x) = 0$ for all $j \in [\![0, k-1]\!]$, proving that the two differential equations are equivalent.

To prove that such an equation always exists, it is enough to show that the following one is suitable:

$$\mathrm{Wr}(g, f_1, \dots, f_k) = 0, \tag{3.3}$$

where g is the unknown. By definition of the Wronskian, this equation has order at most k. By properties of the Wronskian, all the f_i's are solution of Equation (3.3), proving by Lemma 3.1.4 that the order is precisely k, showing the result. □

Remark 3.1.11. *In particular, given a set of affine powers of size s, the previous construction yields a SDE$(s, \binom{s}{2})$ satisfied by all the elements of the set by factorizing the Wronskian following Proposition 2.2.4. Notice that this is also a consequence of Proposition 3.1.3, as the choice of parameters $t = k + 1$, $k = s$, $l = \binom{s}{2}$ satisfies Equation* (3.2)*. In fact, it was this SDE that was used in the first version of [30] because, as already pointed out, the fact that this is the unique SDE of order s yields simple reconstruction algorithms. However, as we will see in Proposition 3.2.4, this choice of parameters gives worse bound for algorithms that the ones we are going to provide in Chapter 4 (using the choice of parameters $t = k + 1$, $k = 2s - 1$, $l = 0$), even though these algorithms will require more work to ensure their correctness.*

Remark 3.1.12. *By Lemma 3.1.10 and Remark 3.1.11, the minimal shift one can hope in general for a SDE of minimal order s satisfied by a set of s affine powers is $\binom{s}{2}$. However, as already pointed out, if we allow the order to be slightly larger ($k = 2s - 1$), then the shift drops to 0. Similar results are proved in [11] for univariate algebraic functions: the authors proved that the linear differential equation of minimal order has coefficients with cubic degree and that there exists a linear differential equation of linear order whose coefficients only have quadratic degrees.*

As already pointed out in the two previous remarks, given a linearly independent set $S = \{(x - a_i)^{e_i} : 1 \leq i \leq s\}$, there always exists a SDE$_{\leq}(2s - 1, 0)$ and a SDE$(s, \binom{s}{2})$ satisfied by all the elements of S, and this is the best one can hope in general. Yet, we now prove that if S is either a "Waring set" (all the e_i's are equal) or a "Sparsest-shift set" (all the a_i's are equal), then in fact there exists a SDE$(s, 0)$ satisfied by all the elements of S, which is unique up to equivalence by Lemma 3.1.10. Moreover, following the ideas of Section 3.1.2, we will show that in fact in both cases this SDE is "completely unique" (i.e. up to scalar multiplication) by describing more precisely the coefficients in term of S.

Proposition 3.1.13. *Let S be a linearly independent set of the form*

$$S = \{(x - a_i)^d : 1 \leq i \leq s\},$$

with $a_i \in \mathbb{F}, d \in \mathbb{N}$. Then there exists a unique (up to scalar multiplication) SDE$(s, 0)$ satisfied by all the elements of S.

Proof. By Lemma 3.1.10, we consider the only candidate (up to equivalence) which is the SDE in the unknown g given by the Wronskian:

$$\mathrm{Wr}(g, (x - a_1)^d, \ldots, (x - a_s)^d)(x) = 0. \tag{3.4}$$

We have to show that this equation can be factored so that the remaining i-th coefficient has degree bounded by i. After factoring out $(x - a_i)^{d-s}$ for all i, we get the reduced SDE:

$$\sum_{i=0}^{s} R_i(x) g^{(i)}(x) = 0,$$

where

$$R_i = \begin{vmatrix} (x-a_1)^s & \dots & (x-a_s)^s \\ d^{\underline{1}}(x-a_1)^{s-1} & \dots & d^{\underline{1}}(x-a_s)^{s-1} \\ \vdots & \ddots & \vdots \\ d^{\underline{i-1}}(x-a_1)^{s-i+1} & \dots & d^{\underline{i-1}}(x-a_s)^{s-i+1} \\ d^{\underline{i+1}}(x-a_1)^{s-i-1} & \dots & d^{\underline{i+1}}(x-a_s)^{s-i-1} \\ \vdots & \ddots & \vdots \\ d^{\underline{s}} & \dots & d^{\underline{s}} \end{vmatrix}$$

Because of the nice structure induced by all the exponents being equal to d, we have that $R'_{i+1} = \frac{1}{(d-i)(s-i)} R_i$, and hence $\deg(R_i) = \deg(R_s) - (s-i)$. If we factor out the constants on each row in R_s we get that

$$R_s = \prod_{i=1}^{s} d^{\underline{i}} \cdot \begin{vmatrix} (x-a_1)^s & \dots & (x-a_s)^s \\ (x-a_1)^{s-1} & \dots & (x-a_s)^{s-1} \\ \vdots & \ddots & \vdots \\ (x-a_1) & \dots & (x-a_s) \end{vmatrix}$$

We factor $(x-a_i)$ on each row and use the known formula for the determinant of a Vandermonde matrix to obtain:

$$R_s = \prod_{i=1}^{s} d^{\underline{i}} \cdot \prod_{i=1}^{s} (x-a_i) \cdot \prod_{i<j} (a_i - a_j)$$

We have $\deg(R_s) = s$, and hence $\deg(R_i) = i$, which shows that the reduced SDE has in fact a zero shift.
To show that this SDE is unique, notice first that $d > s$ because S is linearly independent. As a consequence, for any SDE$(s, 0)$ satisfied by all the elements of S, then a_i is a root of the last coefficient of the SDE by Proposition 3.1.5. In particular, the last coefficient must be a scalar multiple of $\prod_{i=1}^{s}(x-a_i)^{e_i}$, proving that the SDE$(s, 0)$ is unique up to scalar multiplication. □

Proposition 3.1.14. *Let S be a linearly independent set of the form*

$$S = \{(x-a)^{e_i} : 1 \le i \le s\},$$

with $a \in \mathbb{F}, e_i \in \mathbb{N}^*$. Then there exists a unique (up to scalar multiplication) SDE$(s, 0)$ satisfied by all the elements of S.

Proof. Again, we consider the only candidate (up to equivalence) which is the SDE in the unknown g given by the Wronskian:

$$\mathrm{Wr}(g, (x-a)^{e_1}, \dots, (x-a)^{e_s})(x) = \sum_{i=0}^{s} P_i(x)\, g^{(i)}(x) = 0. \tag{3.5}$$

Because of the stepped sequence of degrees in the determinant defining P_i, there exists an integer Δ_i such that every permutation σ corresponds to a term $c_\sigma(x-a)^{\Delta_i}$ in the computation of P_i. More precisely, we have

$$\Delta_i = \left(\sum_{j=1}^{s} e_j\right) - \binom{s+1}{2} + i$$

Thus P_i is either 0, or some constant times $(x-a)^{\Delta_i}$. Moreover, we have $\Delta_{i+1} = \Delta_i + 1$ and hence we can rewrite the SDE as

$$\sum_{i=0}^{s} c_i (x-a)^{\Delta_0+i} g^{(i)}(x) = 0$$

with $c_i \in \mathbb{F}$. We factorize this equation by $(x-a)^{\Delta_0}$ to obtain an SDE$(s,0)$ satisfied by all the elements of S.

Assume now that all the elements of S satisfies the following SDE:

$$\sum_{i=0}^{s} Q_i(x) g^{(i)}(x) = 0.$$

Since the set $S \cup \{1\}$ is linearly independent (stepped degree sequence), then 1 is not a solution of this SDE by Lemma 3.1.4, proving that $Q_0 \in \mathbb{F} \setminus \{0\}$. By Lemma 3.1.10, we have $P_i Q_s = P_s Q_i$ for all $i \in [\![0, s]\!]$ and in particular $P_0 Q_s = c(x-a)^s Q_0$ with $c \in \mathbb{F} \setminus \{0\}$. Therefore, we have $Q_s = \frac{cQ_0}{P_0}(x-a)^s$, proving that this SDE is just a scalar multiple of the previous one. □

3.2 Lower bounds

As announced before, we will use a variant of the space of shifted derivatives that has an additional parameter:

$$\left\langle x^j \cdot f^{(i)} \right\rangle_{D(t,k,l)} \stackrel{\text{def}}{=} \mathbb{F}\text{-span}\left\{ x^j \cdot f^{(i)}(x) \ : \ (i,j) \in D(t,k,l) \right\}$$

We then define the complexity measure associated to this space: for a polynomial $f \in \mathbb{F}[x]$,

$$C_{t,k,l}(f) \stackrel{\text{def}}{=} \dim \left\langle x^j \cdot f^{(i)} \right\rangle_{D(t,k,l)}. \tag{3.6}$$

Notice that the original version we used in [42] corresponds to the case where t is set to $k+1$. As already pointed out in Remark 3.1.2, this complexity measure is deeply interconnected with the differential equations defined in previous section by the following result.

Proposition 3.2.1. *For any $f \in \mathbb{F}[x]$, if f doesn't satisfy any SDE$_{\leq}(t,k,l)$, then $\left\langle x^j \cdot f^{(i)} \right\rangle_{D(t,k,l)}$ is full, i.e. we have*

$$C_{t,k,l}(f) = |D(t,k,l)| = (k+1)(l+1) + \binom{t}{2}.$$

In the following, we will first prove that this complexity measure is small on simple building blocks of our model. Then, to obtain a lower bound, it will be enough to find an explicit polynomial f that doesn't satisfy any "small" SDE.

3.2.1 Potential usefulness

Using a similar argument to the proof of Proposition 3.1.3, we obtain the following upper bound on $C_{t,k,l}(f)$ for an affine power f.

Lemma 3.2.2. *Let $f(x) = (x-a)^e$. Then $C_{t,k,l}(f) \leq k + l + 1$.*

Proof. Notice that all the polynomials in $\left\langle x^j \cdot f^{(i)} \right\rangle_{D(t,k,l)}$ are multiple of $(x-a)^{e-k}$ and have degree bounded by $e + l$. Therefore, we have

$$\left\langle x^j \cdot f^{(i)} \right\rangle_{D(t,k,l)} \subseteq \mathbb{F}\text{-span}\{x^j \cdot (x-a)^{e-k} \ : \ 0 \leq j \leq k+l\},$$

and hence the dimension is upper bounded by $k + l + 1$. □

As a consequence, we derive an upper bound for an element that has a decomposition in Model 2 with s terms.

Proposition 3.2.3. *Let $f \in \mathbb{F}[x]$ be a polynomial such that f can be written as $f(x) = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}$. Then $C_{t,k,l}(f) \leq s(k + l + 1)$.*

Proof. The complexity measure $C_{t,k,l}(f)$ is sub-additive since the dimension is sub-additive and since we have

$$\left\langle x^j \cdot (f+g)^{(i)} \right\rangle_{D(t,k,l)} \subseteq \left\langle x^j \cdot f^{(i)} \right\rangle_{D(t,k,l)} + \left\langle x^j \cdot g^{(i)} \right\rangle_{D(t,k,l)}.$$

Therefore, using Lemma 3.2.2, we directly obtain the bound. □

Using this upper bound, we can obtain a weak version of Proposition 3.1.3: given a polynomial $f(x) = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}$ and parameters (t,k,l) that satisfy equation 3.2, then f satisfies an SDE$_{\leq}(t,k,l)$. Indeed, by Proposition 3.2.3, we have

$$C_{t,k,l}(f) \leq s(l + k + 1) < |D(t,k,l)|$$

and therefore the generating family of $\left\langle x^j \cdot f^{(i)} \right\rangle_{D(t,k,l)}$ is linearly dependent, which implies the existence of an SDE$_{\leq}(t,k,l)$ using Remark 3.1.2. This result is weaker than Proposition 3.1.3 since we have no guarantee that the SDE obtained this way

is also satisfied by the affine powers $(x-a_i)^{e_i}$ for all $i \in [\![1,s]\!]$. However, in the case where all the a_i's are distinct, we will now derive some conditions on the e_i's that will ensure this property, using the Wronskian. This result will be one of the main tools to design reconstruction algorithms of Chapter 4. Roughly speaking, this result says that if f satisfies a SDE, then every term in the optimal expression of f with exponent e_i big enough also satisfies the same SDE. It is interesting to notice that there is no dependency in the parameter t, which explain why we always take $t = k+1$ in Chapter 4.

Proposition 3.2.4. *Let $f \in \mathbb{F}[x]$ be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x-a_i)^{e_i},$$

with $\alpha_i \in \mathbb{F}$ nonzero, the $a_i \in \mathbb{F}$ all distinct, and $e_i \in \mathbb{N}$. Whenever f satisfies a SDE(t,k,l), then for all $e_i \geq k+(k+l)(s-1)+\binom{s}{2}$ we have that $(x-a_i)^{e_i}$ satisfies the same SDE.

Proof. Suppose that f satisfies the following SDE(t,k,l):

$$\sum_{i=0}^{k} P_i(x)\, g^{(i)}(x) = 0,$$

with $\deg(P_i) \leq i+l$ for all $i \in [\![0,t-1]\!]$, $\deg(P_i) \leq l$ for all $i \in [\![t,k]\!]$. We assume that $(x-a_1)^{e_1}$ does not satisfy this equation, and prove that it implies that e_1 is small. For every $j \in [\![1,s]\!]$, we denote by f_j and R_j the polynomials such that

$$f_j = \sum_{i=0}^{k} P_i(x)\, (\alpha_j (x-a_j)^{e_j})^{(i)} = R_j(x)\, (x-a_j)^{d_j},$$

where $d_j := \max\{e_j - k, 0\}$. We observe that $\deg(f_j) \leq e_j + l$, so $\deg(R_j) \leq k+l$, and that $-f_1 = \sum_{j=2}^{s} f_j \neq 0$. We consider a linearly independent subfamily of $f_2, \ldots, f_s$, namely $\{f_j \,|\, j \in J\}$ with $J = \{j_1, \ldots, j_p\} \subseteq \{2, \ldots, s\}$. We write $f_1 = \sum_{i=1}^{p} \alpha_i f_{j_i}$, and, by properties of the determinant, we have

$$0 \neq \mathrm{Wr}(\alpha_1 f_{j_1}, f_{j_2}, \ldots, f_{j_p}) = \mathrm{Wr}(f_1, f_{j_2}, \ldots, f_{j_p}).$$

Following Proposition 2.2.4, we factor the Wronskians to obtain

$$\begin{cases} \mathrm{Wr}(\alpha_1 f_{j_1}, f_{j_2}, \ldots, f_{j_p}) & = \prod_{d_i \geq p-1} (x-a_i)^{d_k-(p-1)} \cdot W_1 \\ \mathrm{Wr}(f_1, f_{j_2}, \ldots, f_{j_p}) & = (x-a_1)^{d_1-(p-1)} \cdot W_2 \end{cases}$$

with $\deg(W_1) \leq \sum_{i=1}^{p} [\deg(R_j)+p-1] - \binom{p}{2}$. Since a_1 is distinct from $a_{j_1}, \ldots, a_{j_p}$, then $(x-a_1)^{d_1-(p-1)}$ must divide W_1 and therefore we have

$$e_1 - k - (p-1) \leq \deg(W_1) \leq (k+l)p + \binom{p}{2}.$$

Since $p \leq s-1$, we get that $e_1 \leq k+s-2+(k+l)(s-1)+\binom{s-1}{2} < k+(k+l)(s-1)+\binom{s}{2}$, proving the result. □

3.2.2 Hard polynomials

In this section, we will prove lower bounds in Model 2 for two families of polynomials: when $f(x) = \prod_{i=1}^{s}(x-a_i)^{d/s}$ and when $f(x) = \sum_{i=1}^{s}(x-a_i)^d$. In both cases we will obtain a $\Omega(\sqrt{d})$ lower bound by proving that these polynomials don't satisfy any "small" SDE. More precisely, the conjunction of Proposition 3.2.1 and Proposition 3.2.3 gives the following result.

Proposition 3.2.5. *For any $f \in \mathbb{F}[x]$, if f doesn't satisfy any SDE$_{\leq}(t,k,l)$, then we have*

$$\mathit{AffPow}_{\mathbb{F}}(f) \geq \frac{(k+1)(l+1)+\binom{t}{2}}{l+k+1}.$$

Proof. By Proposition 3.2.1, we have

$$C_{t,k,l}(f) = (k+1)(l+1)+\binom{t}{2}.$$

Now, if there exists a decomposition of f in Model 2 with m terms, we have by Proposition 3.2.3 the following upper bound, which proves the result:

$$C_{t,k,l}(f) \leq m(l+k+1).$$

□

For the first family of polynomials, the proof of the lower bound is quite straightforward and relies on the following lemma.

Lemma 3.2.6. *Let $s, e \in \mathbb{N}$ be integers. Let $f(x) = \prod_{i=1}^{s}(x-a_i)^e$ with $a_1, \ldots, a_s \in \mathbb{F}$ distinct. If f satisfies a SDE$(0,k,l)$, then either $k > e$, or $l \geq s$.*

Proof. We will prove the results by showing that if f satisfies a SDE and $k \leq e$, then $l \geq s$ must hold. Assume that f satisfies a SDE$(0,k,l)$ of the following form:

$$\sum_{i=0}^{k} P_i(x) f^{(i)}(x) = 0,$$

with $k \leq e$. We use a similar argument to Proposition 3.1.5: set $Q = \prod_{i=1}^{s}(x-a_i)$ so that $f = Q^e$. Now, since $Q \wedge Q' = 1$, one can show by recurrence that $Q^{(i)} \wedge Q^e = Q^{e-i}$ for all $i \leq e$. As a consequence, we must have $Q | P_k$ and therefore $s = \deg(Q) \leq \deg(P_k) \leq l$, proving the result. □

As a consequence, we obtain our first lower bound by choosing the best values for s and e, which is $s = e = \sqrt{d}$ because of the symmetry between k and l in the bound of Proposition 3.2.5.

Proposition 3.2.7. *Let $d \in \mathbb{N}$ be a perfect square and $s = \sqrt{d}$. Let $f(x) = \prod_{i=1}^{s}(x-a_i)^s$ with $a_1, \ldots, a_s \in \mathbb{F}$ distinct. Then $\mathit{AffPow}_{\mathbb{F}}(f) = \Omega(\sqrt{d})$.*

Proof. We choose $t = 0$, $k = s$ and $l = s - 1$ so that f doesn't satisfy any $\mathrm{SDE}_{\leq}(t, k, l)$ by Lemma 3.2.6. By Proposition 3.2.5, we therefore have

$$\mathrm{AffPow}_{\mathbb{F}}(f) \geq \frac{(k+1)(l+1) + \binom{t}{2}}{l+k+1} = \frac{s(s+1)}{2s} = \frac{\sqrt{d}+1}{2}.$$

□

The second family of polynomials, when $f(x) = \sum_{i=1}^{s}(x - a_i)^d$, is interesting for several reasons. Firstly, we proved in Section 3.2.1 that such a polynomial should have a small complexity, however we will now prove that this complexity is in fact not too small. In particular, the order of the lower bound we will obtain is optimal since we will get that $\mathrm{AffPow}_{\mathbb{F}}(f) = \Omega(s)$ for $s = \Theta(\sqrt{d})$. Secondly, we will show that k and l are orthogonal for f in the following sense: if one of them is small, the other one must be large if an SDE is satisfied by f. This property is captured in the two following symmetric lemmas.

Lemma 3.2.8. *Let $s, d \in \mathbb{N}$ be integers such that $s \leq d + 1$. Let $f(x) = \sum_{i=1}^{s}(x - a_i)^d$ with $a_1, \ldots, a_s \in \mathbb{F}$ distinct. If f satisfies a SDE$(0, k, l)$, then at least one of the two following conditions holds:*

$$i) \quad l \geq s, \qquad\qquad ii) \quad k > \frac{d}{s} - \frac{3}{2}(s-1).$$

Proof. Assume that f satisfies a $\mathrm{SDE}(0, k, l)$ of the following form:

$$\sum_{i=0}^{k} P_i(x) f^{(i)}(x) = 0,$$

with $l < s$. Since $\deg(P_k) \leq l < s$ and the a_i's are distinct, at least one of the a_i is not a root of P_k and therefore Proposition 3.1.5 implies that the corresponding affine power $(x - a_i)^d$ doesn't satisfy the SDE. By Proposition 3.2.4, we therefore must have

$$d < k + (k + l)(s - 1) + \binom{s}{2}.$$

Using the fact that $l \leq s - 1$, we obtain the lower bound on k. □

Lemma 3.2.9. *Let $s, d \in \mathbb{N}$ be integers such that $s \leq d + 1$. Let $f(x) = \sum_{i=1}^{s}(x - a_i)^d$ with $a_1, \ldots, a_s \in \mathbb{F}$ distinct. If f satisfies a SDE(t, k, l), then at least one of the two following conditions holds:*

$$i) \quad k \geq s, \qquad\qquad ii) \quad l > \frac{d}{s-1} - \frac{3}{2}s.$$

Proof. Assume that f satisfies a $\mathrm{SDE}(t, k, l)$ of the following form:

$$\sum_{i=0}^{k} P_i(x) f^{(i)}(x) = 0,$$

with $k < s$. By Proposition 2.2.3, the $f_i \stackrel{\text{def}}{=} (x - a_i)^d$'s are linearly independent, and therefore by Lemma 3.1.4 at least of one the f_i's is not solution of this SDE. By Proposition 3.2.4, we therefore must have

$$d < k + (k + l)(s - 1) + \binom{s}{2}.$$

Using the fact that $k \leq s - 1$, we obtain the lower bound on l. □

Notice that Lemma 3.2.9 is more powerful than Lemma 3.2.8 to prove a lower bound because it makes no assumption on the value of t. Therefore, we only give a proof of a lower bound using Lemma 3.2.9 as the one obtained using Lemma 3.2.8 is slightly worse (same order, but worse constant).

Proposition 3.2.10. *Let $d \in \mathbb{N}$, $\alpha \in]0; \sqrt{2/3}[$, and let $s = \alpha\sqrt{d}$. Let $f(x) = \sum_{i=1}^{s}(x - a_i)^d$ with $a_1, \ldots, a_s \in \mathbb{F}$ distinct. Then AffPow$_{\mathbb{F}}(f) = \Omega(\sqrt{d})$.*

Proof. We set the parameters to the following values:

- $k = s - 1$,
- $t = k + 1 = s$,
- $l = (\frac{1}{\alpha} - \frac{3\alpha}{2})\sqrt{d} > 0$,

and apply Lemma 3.2.9: f doesn't satisfy any SDE$_{\leq}(t, k, l)$. By Proposition 3.2.5, we therefore have

$$\text{AffPow}_{\mathbb{F}}(f) \geq \frac{(k + 1)(l + 1) + \binom{t}{2}}{l + k + 1} \geq \frac{2(\alpha^2 - 1)\alpha}{\alpha^2 - 2}\sqrt{d},$$

proving that AffPow$_{\mathbb{F}}(f) \geq c\sqrt{d}$ with $c > 0$. □

Remark 3.2.11. *One could also obtain this lower bound as a consequence of Corollary 2.2.8. Indeed, if we take $f(x) = \sum_{i=1}^{s}(x - a_i)^d$ with $s = \sqrt{(d + 1)/2}$, then this expression is the unique optimal one by Corollary 2.2.8, proving that AffPow$(f) = s = \Theta(\sqrt{d})$. This idea is the main motivation to the study of linear independence results in Section 3.3.*

3.2.3 Extension and limitations

In this section, we will see how this method of shifted derivatives can be used to obtain lower bounds for Model 4 and see the limitations of this method. In particular, the limitations will also hold true for the simpler Model 2. The only difference with the previous part is the "simple blocks" of this model which have the following complexity:

Lemma 3.2.12. *Let $f(x) = Q(x)^e$ with $\deg(Q) \leq r$. Then $C_{t,k,l}(f) \leq l + kr + 1$.*

Proof. All the polynomials in $\left\langle x^j \cdot f^{(i)} \right\rangle_{D(t,k,l)}$ are multiple of $Q(x)^{e-k}$ and have degree bounded by $re + l$. Therefore, we have

$$\left\langle x^j \cdot f^{(i)} \right\rangle_{D(t,k,l)} \subseteq \mathbb{F}\text{-span}\{x^j \cdot Q(x)^{e-k} \ : \ 0 \leq j \leq kr + l\}.$$

□

As a byproduct, we obtain this new version of Proposition 3.2.5.

Proposition 3.2.13. *For any $f \in \mathbb{F}[x]$, if f doesn't satisfy any SDE$_{\leq}(t, k, l)$, then we have*

$$\textit{SmallPow}_{\mathbb{F}}(f) \geq \frac{(k+1)(l+1) + \binom{t}{2}}{l + kr + 1}.$$

Notice that if we set $r = 1$ we recover the results of previous sections, but the parameter r breaks the symmetry between k and l in the previous expression, and this will motivate the following choices of parameters to obtain lower bounds.

Proposition 3.2.14. *Let $d, r \in \mathbb{N}$. Let $f(x) = \prod_{i=1}^{\sqrt{dr}} (x - a_i)^{\sqrt{d/r}} \in \mathbb{F}[x]$, with $a_1, \ldots, a_{\sqrt{dr}} \in \mathbb{F}$ distinct. Then SmallPow$_{\mathbb{F}}(f) = \Omega(\sqrt{d/r})$.*

Proof. We choose $t = 0$, $k = \sqrt{d/r}$ and $l = \sqrt{dr} - 1$ so that f doesn't satisfy any SDE$_{\leq}(t, k, l)$ by Lemma 3.2.6. By Proposition 3.2.13, we therefore have

$$\text{SmallPow}_{\mathbb{F}}(f) \geq \frac{(k+1)(l+1) + \binom{t}{2}}{l + kr + 1} = \Omega\left(\sqrt{\frac{d}{r}}\right).$$

□

Proposition 3.2.15. *Let $d \in \mathbb{N}$ and let $s = \sqrt{d/r}$. Let $f(x) = \sum_{i=1}^{s} (x - a_i)^d$ with $a_1, \ldots, a_s \in \mathbb{F}$ distinct. Then SmallPow$_{\mathbb{F}}(f) = \Omega(\sqrt{d/r})$.*

Proof. We take $k = s - 1, t = k + 1$ and $l = \frac{d}{s} - \frac{3}{2}s$ so that f doesn't satisfy any SDE$_{\leq}(t, k, l)$ by Lemma 3.2.9. By Proposition 3.2.13, we therefore have

$$\text{SmallPow}_{\mathbb{F}}(f) \geq \frac{(k+1)(l+1) + \binom{t}{2}}{l + kr + 1} = \Omega\left(\sqrt{\frac{d}{r}}\right).$$

□

Unfortunately, these lower bounds are the best one can get using the method of shifted derivatives. Indeed, the trivial upper bound on the dimension is:

$$C_{t,k,l}(f) \leq \min\left(d + l + 1, \ \ (k+1)(l+1) + \binom{t}{2}\right)$$

This upper bound is tight since we just proved that the equality holds for two examples. Moreover, following Proposition 3.2.13 and since $t \leq k+1$, the best possible lower bound on the number of summands in Model 4 we can obtain is:

$$s \geq \frac{\min\left(d+l+1,\ (k+1)(l+1)+\binom{k+1}{2}\right)}{l+kr+1}$$

As seen before, the choice of parameters $k = O(\sqrt{d/r})$ and $l = O(\sqrt{dr})$ yields a $\Omega(\sqrt{d/r})$ lower bound, and we claim that this is optimal with this methods.

Lemma 3.2.16. *For all $k, l \in \mathbb{N}$, we have*

$$\min\left(d+l+1,\ \ (k+1)(l+1)+\binom{t}{2}\right) \leq \sqrt{\frac{d}{r}}+1$$

Proof. Denote

$$f(k,l) = \min\left(\frac{d+l+1}{l+kr+1},\ \ \frac{(k+1)(l+1)+\binom{k+1}{2}}{l+kr+1}\right) = \min(f_1, f_2)$$

We distinguish two cases:

- $k \geq \sqrt{d/r}$: the function $g : l \mapsto f_1(k,l)$ is monotonous on $[0;+\infty[$. Moreover we have $g(+\infty) = 1$ and $g(0) = \frac{d+1}{kr+1} < \sqrt{d/r}+1$. Therefore, for any $l \in \mathbb{N}$, we have $g(l) < \sqrt{d/r}+1$.
- $k \leq \sqrt{d/r}$: the function $h : l \mapsto f_2(k,l)$ is monotonous on $[0;+\infty[$. Moreover we have $h(+\infty) = k+1 \leq \sqrt{d/r}+1$ and $h(0) = \frac{(k+2)(k+1)}{2(kr+1)} \leq \frac{k+2}{r} < \sqrt{d/r}$. Thus we have $h(l) < \sqrt{d/r}+1$ for any $l \in \mathbb{N}$.

□

3.3 Linear independence

Studying the linear independence of shifted powers is a natural and challenging problem in its own right, but it also has a motivation coming from the search for lower bounds. The connection between these two problems arises when the "hard polynomial" f itself is defined as a linear combination of shifted powers, such as in Proposition 3.2.10. In this case, another decomposition of f in Model 2 can be rewritten as a linear dependence relation between shifted powers (those in the decomposition and those occurring in the definition of f). If we can show that such a linear dependence is impossible for small enough s, we have a lower bound on s, as illustrated in Remark 3.2.11. This was also the way that the lower bounds in [29] were obtained. The goal of this section is to prove sufficient conditions for the linear independence of a family of shifted powers. One trivial such condition is when the degrees of

the polynomials in the family are all distinct. Other well-known independent linear families of $\mathbb{F}[x]$ are the ones pictured in Proposition 2.2.3: for any distinct $(a_i) \in \mathbb{F}^{d+1}$, the polynomials $(x - a_i)^d$ are linearly independent. As already pointed out in Section 2.2.1, nullity of the Wronskian is a necessary and sufficient condition for the linear independence of polynomials, so our problem always reduces in principle to the verification that the Wronskian of F is nonzero. Unfortunately, the resulting determinant looks hardly manageable in general. As a result, little seems to be known in the case of unequal exponents (the case of equal exponents is tractable because the Wronskian determinant becomes a Vandermonde matrix after multiplication of rows by constants). One exception is the so-called Jordan's lemma (Lemma 2.2.16), which provides a generalization of Proposition 2.2.3.
So far, we have only discussed sufficient conditions for linear independence. The following "Pólya condition" is an obvious *necessary condition*:

Definition 3.3.1. *For a sequence $e = (e_1, \dots, e_s)$ of integers, we define again $n_i \stackrel{def}{=} |\{j : e_j < i\}|$. We say that e satisfies the Pólya condition if $n_i \leq i$ for all $i \in \mathbb{N}$.*
For a family $F = \{(x - a_i)^{e_i} : 1 \leq i \leq s\}$, we say that F satisfies the Pólya condition if $e = (e_1, \dots, e_s)$ does.

The name *Pólya condition* is borrowed from the theory of Birkhoff interpolation [58, 29]. This necessary condition for linear independence is not sufficient: for instance we have the linear dependence relation $(x+1)^2 - (x-1)^2 - 4x = 0$. As we shall see in Section 3.3.3, the Pólya condition turns out to be sufficient in a probabilistic sense: if the shifts a_i are taken uniformly at random, the resulting family is linearly independent with high probability. As pointed out above, little is known about deterministic sufficient conditions for linear independence. But there is an exception when $\mathbb{F}$ is the field of real numbers: in this case, some recent progress was made in [29] thanks to a connection between Birkhoff interpolation and linear independence of shifted powers. In particular, the authors showed that the Pólya condition is only a factor of 2 away from being also a sufficient condition for linear independence:

Theorem 3.3.2 (Theorem 3 in [29])**.** *Let F and the n_i's be as in Definition 3.3.1, and let $d = \max e_i$. If all the a_i's are real, and $n_1 \leq 1, n_j + n_{j+1} \leq j + 1$ for all $j = 1, \dots, d$, then the elements of F are linearly independent.*

They also gave an example of linear dependence that violates only one of the inequalities of Theorem 3.3.2, showing that this result is essentially optimal. However, Theorem 3.3.2 fails badly over the field of complex numbers, as shown by Example 2.0.1. Indeed, if we take $k = \sqrt{d}$ in Example 2.0.1, we have that the $2k = 2\sqrt{d}$ shifted powers in the set

$$\{(x + \xi^j)^d \mid 1 \leq j \leq k\} \cup \{x^{d-i} \mid i \equiv -1 \pmod{k}, 0 \leq i \leq d\}$$

are linearly dependent; and the exponents of these shifted powers clearly satisfy the hypothesis of Theorem 3.3.2. The question of finding a "good" sufficient condition for linear independence over $\mathbb{C}$ was left open in [29], and in [31] we proposed the following conjecture.

Conjecture 3.3.3. *There are absolute constants a and b such that for all large enough integers s, the elements in any family $F = \{(x - a_i)^{e_i} : 1 \leq i \leq s\}$ of s complex shifted powers are linearly independent if $e_i \geq as + b$ for all $i \in [\![1, s]\!]$.*

If this conjecture holds true, one must have $a \geq 1$. Indeed, a family where $e_i \leq s-2$ for all i will violate the Pólya condition. One can say more. Indeed, we will see in Proposition 3.3.4 that the counterpart of Conjecture 3.3.3 for the field of real numbers holds true, and that one may take $a = 2, b = -4$. We will also show that this result is best possible over $\mathbb{R}$: one cannot take $a = 2, b = -5$. Therefore, if Conjecture 3.3.3 holds true one must in fact have $a \geq 2$. In Section 3.3.2, we will show that a weak version of Conjecture 3.3.3 holds true: it suffices to replace the hypothesis $e_i \geq as+b$ by the stronger assumption $e_i \geq s(s-1)/2$.

3.3.1 The real case

In the following we derive from Theorem 3.3.2 another sufficient condition for the linear independence of families without small exponents, and show that the result is tight.

Proposition 3.3.4. *For any family*

$$F = \{(x - a_i)^{e_i} : a_i \in \mathbb{R}, e_i \geq \max(1, 2s - 4), 1 \leq i \leq s\},$$

the elements of F are linearly independent.

Proof. Assume without loss of generality that $d = e_1 \geq e_2 \geq \cdots \geq e_s$. We first eliminate a few trivial cases:

- $s = 1$: the elements of F are linearly independent.
- $s = 2$: two affine powers are linearly dependent if and only if they are equal, thus the elements of F are linearly independent.
- $e_1 = e_2 = \cdots = e_s$. Then all the a_i are distinct because all (e_i, a_i) are distinct. Since $d \geq 2s - 4$ and $s \geq 3$, we have $s \leq d + 1$ and the elements of F are linearly independent by Proposition 2.2.3.
- $e_1 > e_2$: no linear dependence could involve $(x - a_1)^{e_1}$, hence it is enough to show that the elements of the subfamily $F' = \{(x - a_i)^{e_i} : 2 \leq i \leq s\}$ are linearly independent. Since F' satisfies the hypotheses of the Proposition, we can deal with this case by induction on s.

We now have $e_1 = e_2 > e_s$ and $s \geq 3$ which implies that $d \geq 2s - 3$ and that $n_d \leq s - 2$. Such a family satisfies the hypotheses of Theorem 3.3.2, which directly yields the result. Indeed:

- For $i = 0$, we have $n_1 = 0 \leq 1$.
- For $i \leq 2s - 5$, we have $n_i + n_{i+1} = 0$.
- For $2s - 5 < i < d$, we have $n_i + n_{i+1} \leq 2(s - 2) \leq i + 1$.

- For $i = d$, we have $n_d + n_{d+1} \leq 2s - 2 \leq d + 1$.

□

In order to show that the bound $e_i \geq 2s - 4$ in the above result is tight, we will consider the real polynomial $H_{2d+1}(x) = (x+1)^{2d+2} - x^{2d+2}$ and show that it has a large Waring rank (see Section 1.2.1).

Proposition 3.3.5. *We have* $\mathit{Waring}_{\mathbb{R}}(H_{2d+1}) = \mathit{Waring}_{\mathbb{C}}(H_{2d+1}) = d + 1$.

Proof. We will use the algorithmic result in [20, Section 3] to compute the complex Waring rank of H_{2d+1} and then prove that it coincides with its real Waring rank. We consider $P(x, y)$ the homogenization of H_{2d+1} with respect to the variable y:

$$P(x,y) = \sum_{i=0}^{2d+1} \binom{2d+2}{i} x^i y^{2d+1-i} = \sum_{i=0}^{2d+1} \binom{2d+2}{i+1} x^{2d+1-i} y^i.$$

We extract the coefficients

$$Z_i = \frac{\text{coeff}(P, x^{2d+1-i}y^i)}{\binom{2d+1}{i}} = \frac{\binom{2d+2}{i+1}}{\binom{2d+1}{i}} = \frac{2d+2}{i+1}$$

and, following [20], we construct the matrix

$$M = \begin{pmatrix} Z_0 & Z_1 & \cdots & Z_d \\ Z_1 & Z_2 & \cdots & Z_{d+1} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{d+1} & Z_{d+2} & \cdots & Z_{2d+1} \end{pmatrix} = (2d+2) \cdot \begin{pmatrix} \frac{1}{1} & \frac{1}{2} & \cdots & \frac{1}{d+1} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{d+2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{d+2} & \frac{1}{d+3} & \cdots & \frac{1}{2d+2} \end{pmatrix}$$

The last matrix is a Hilbert matrix with an additional row. Hilbert matrices are known to be invertible, as special cases of Cauchy matrices, therefore we have $rank(M) = d + 1$, which implies, according to [20] that the complex Waring rank is either $d + 1$ or $d + 2$. In order to show that it is in fact $d + 1$, we have to find a vector $f \in \mathbb{C}^{d+2}$ in the kernel of M^t (which is unique up to scalar multiplication) and prove that the corresponding polynomial $F(x) = \sum_{i=0}^{d+1} f_i x^i$ does not have multiple roots.
Notice that the i^{th} row of $M^t f$ can be rewritten as

$$(M^t f)_i = \sum_{j=0}^{d+1} \frac{1}{i+j+1} f_j = \sum_{j=0}^{d+1} \int_0^1 x^i \cdot f_j \, x^j dx = \int_0^1 x^i F(x) dx$$

The equality $M^t f = 0$ can thus be restated as $\langle F, x^i \rangle = 0$ for $i = 0 \dots d$, with the corresponding scalar product $\langle f, g \rangle = \int_0^1 f(x)g(x)dx$. Such a polynomial can be obtained by the Gram-Schmidt process to $\{1, x, \dots x^{d+1}\}$ and is classically known as the shifted Legendre polynomial: it can be obtained from the Legendre polynomial

by the affine transformation $x \mapsto 2x - 1$. A classical result (see, e.g., [4]) is that the Legendre polynomial of degree $d+1$ has $d+1$ distinct real roots in the interval $(-1, 1)$. Therefore our polynomial F has $d+1$ distinct real roots in the interval $(0, 1)$. This shows that $\text{Waring}_{\mathbb{C}}(H) = d+1$.
Moreover, if we denote by (a_i) the roots of F, there exist coefficients (α_i) such that $P(x,y) = \sum_{i=1}^{d+1} \alpha_i (x - a_i y)^{2d+1}$. Since the a_i are real, if we take the real part of this equality, we obtain $P(x,y) = \sum_{i=1}^{d+1} \Re(\alpha_i)(x - a_i y)^{2d+1}$ proving that $\text{Waring}_{\mathbb{R}}(P) = \text{Waring}_{\mathbb{C}}(P) = d+1$. Since $a_i \neq 0$ for all i, we conclude that $\text{Waring}_{\mathbb{R}}(H_{2d+1}) = \text{Waring}_{\mathbb{C}}(H_{2d+1}) = d+1$. □

Remark 3.3.6. *Up to multiplication of rows and columns by constants, the matrix M in the above proof is nothing but the matrix of d-th order partial derivatives of P. This explains why the Waring rank of P is at least equal to the rank of M. Again, we refer to [20] for a proof that the Waring rank is in fact equal to $rank(M)$ or $d + 2 - rank(M)$.*

Remark 3.3.7. *A similar proof shows that $\text{Waring}_{\mathbb{R}}(H_{2d}) = d+1$, proving that in general $\text{Waring}_{\mathbb{R}}(H_d) = \lceil \frac{d+1}{2} \rceil$.*

By this result, there exist $\alpha_1, \ldots, \alpha_{d+1} \in \mathbb{R}$ and $a_1, \ldots, a_{d+1} \in (0,1)$ such that

$$(x+1)^{2d+2} - x^{2d+2} = \sum_{i=1}^{d+1} \alpha_i (x - a_i)^{2d+1}$$

This equality is a linear dependence of $d+3$ terms of degree at least $2d+1 = 2(d+3) - 5$, showing the optimality of the bound $e_i \geq 2s - 4$.

3.3.2 The complex case

In the complex case, we can prove a similar sufficient condition for linear independence:

Proposition 3.3.8. *For any family*

$$F = \{(x - a_i)^{e_i} : a_i \in \mathbb{C}, e_i \geq (s-1)s/2, 1 \leq i \leq s\},$$

the elements of F are linearly independent.

Proof. Take $G \subseteq F$ a minimal generating subfamily of F and assume by contradiction that $|G| = t < s$. Using Proposition 3.1.3, there exists a SDE(t, k, t) for some $k \leq (t+1)t/2$ satisfied simultaneously by every element of G:

$$\sum_{i=0}^{k} P_i(x) f^{(i)}(x) = 0$$

Moreover, since $\langle G \rangle = \langle F \rangle$, we have that every elements of F satisfies this SDE. Using Corollary 3.1.7, since $e_i \geq (s-1)s/2 \geq t(t+1)/2 \geq k$, we thus have that $\prod_{i=1}^{s}(x - a_i)$ divides P_k. This yields $s \leq \deg P_k \leq t < s - 1$, a contradiction. □

We do not know if the bound $e_i \geq (s-1)s/2$ is tight. The best example we know is the one provided in the real case that achieves a linear dependence of s affine powers with $e_i \geq 2s-5$.

3.3.3 Genericity and linear independence

Let $\mathcal{P}_s$ denote the set of sequences $e = (e_1, \ldots, e_s) \in \mathbb{N}^s$ satisfying the Pólya condition. The goal of this section is to study two different random processes for generating a family of affine powers, and to bound the probability that the elements of the generated family are linearly independent. In Corollary 3.3.11, we study the case where the sequence $e \in \mathcal{P}_s$ is fixed, and the a_i's are taken uniformly and independently from a set S. In Theorem 3.3.15, we study the case where we take the a_i's uniformly and independently from a set S and we want them to give independent families for any $e \in \mathcal{P}_s$. Notice that Corollary 3.3.11 does not directly implies Theorem 3.3.15 as the number of sequences in $\mathcal{P}_s$ is infinite.
We will repeatedly use the notation $\langle F \rangle$ to denote the vector space spanned by the elements of F. We first prove that given a family F of s shifted powers such that $1 \notin \langle F \rangle$, the number of shifted powers in $\langle F \rangle$ of degree d is upper bounded by a linear expression in s. The condition $1 \notin \langle F \rangle$ is satisfied if the elements of the set of derivatives $F' = \{f' : f \in F\}$ are linearly independent.

Proposition 3.3.9. *Consider a family $F = \{(x-a_i)^{e_i} : 1 \leq i \leq s\}$ such that $1 \notin \langle F \rangle$. Let $S(d) = \{(x-a)^d \in \langle F \rangle\}$.*
Then for any $e \in \mathbb{N}$, we have $|S(e)| \leq 2s-1$.

Proof. Notice first that for any $e \in \mathbb{N}$, we must have $|S(e)| \leq e$. Otherwise, it would contain a basis of $\mathbb{F}_e[x]$ and we could obtain 1 as a linear combination of elements of F, which contradicts the hypothesis. Therefore we are done if $e \leq 2s-1$.
Otherwise, Proposition 3.1.3 ensures that there exists a SDE$(0,k,l)$ with $k = l = 2s-1$ of order $k' \leq k$ satisfied by all the elements of F:

$$\sum_{i=0}^{k'} P_i(x) f^{(i)}(x) = 0 \qquad (*)$$

Since this equation is satisfied by all the elements of $S(e)$, we use Corollary 3.1.7 to obtain: $|S(e)| \leq \deg P_{k'} \leq l \leq 2s-1$. □

Given a sequence $e \in \mathcal{P}_s$, we can take the a_i's uniformly and independently in an iterative way and use the previous result to lower bound the probability that the elements of the resulting family of shifted powers are linearly independent at each step. The resulting bound is given in Corollary 3.3.11. Its proof requires the following technical lemma.

Lemma 3.3.10. *Let $e = (e_1, \ldots, e_{s+1}) \in \mathcal{P}_{s+1}$, then $f = (e_1 - 1, \ldots, e_s - 1) \in \mathcal{P}_s$.*

Proof. We set $n_i := |\{j : e_j < i,\ 1 \le j \le s+1\}|$ and $n'_i := |\{j : e_j - 1 < i,\ 1 \le j \le s\}|$. Since e satisfies the Pólya condition we have that $n_i \le i$ for all i. Moreover, since e is non-increasing, we have that

- $n'_i = 0$ if $i \le e_{s+1}$,
- $n'_i = n_{i+1} - 1 \le i$ if $i > e_{s+1}$;

hence, f also satisfies the Pólya condition. □

Corollary 3.3.11. *Let $e = (e_1, \dots, e_s) \in \mathcal{P}_s$ and let S be a finite subset of $\mathbb{F}$. Let $a_1, \dots, a_s$ be selected at random independently and uniformly from S. Then*

$$\Pr\Big[\ \{(x-a_i)^{e_i} \ : 1 \le i \le s\} \text{ is linearly independent}\Big] \ge 1 - \frac{s(s-1)}{|S|}$$

Proof. We will prove this by induction on s: for $s = 1$, we always obtain a linearly independent family. We now consider $e = (e_1, \dots, e_{s+1}) \in \mathcal{P}_{s+1}$, and we define the following events on possible outcomes $(a_1, \dots, a_{s+1})$:

$$\begin{array}{lll} A & = \{\{(x-a_i)^{e_i} & : 1 \le i \le s+1\} \quad \text{is linearly independent}\} \\ B & = \{\{(x-a_i)^{e_i} & : 1 \le i \le s\} \quad \text{is linearly independent}\} \\ C & = \{\{(x-a_i)^{e_i-1} & : 1 \le i \le s\} \quad \text{is linearly independent}\} \end{array}$$

Notice that $C \subseteq B \subseteq A$. Using Lemma 3.3.10 and the induction hypothesis we obtain $\Pr(C) \ge 1 - \frac{s(s-1)}{|S|}$. From Proposition 3.3.9 we have $\Pr(A|C) \ge 1 - \frac{2s}{|S|}$. Since $\Pr(A) = \Pr(C) \cdot \Pr(A|C)$ we obtain the inequality

$$\Pr(A) \ge \left(1 - \frac{s(s-1)}{|S|}\right) \cdot \left(1 - \frac{2s}{|S|}\right) \ge 1 - \frac{s(s+1)}{|S|}.$$

□

In the remainder of this section, our goal is to prove Theorem 3.3.15 which states the following: if $a_1, \dots, a_s \in \mathbb{F}$ are selected independently and uniformly at random from a big enough finite set S, then with high probability $(x-a_i)^{e_i} : 1 \le i \le s$ are linearly independent for all $e = (e_1, \dots, e_s) \in \mathcal{P}_s$. A key ingredient of this proof is to have a bound on $\max(e_i)$ for a sequence $e \in \mathcal{P}_s$ such that the elements of the family might be linearly dependent. This bound is obtained from Corollary 2.2.10, and we therefore naturally define the bounded version of $\mathcal{P}_s$: $\mathcal{P}'_s = \{e \in \mathcal{P}_s : \max(e_i) \le \frac{s^2}{2} - 2\}$. The next Corollary ensures that if an outcome $(a_1, \dots, a_s)$ yields a linearly independent family for any $e \in \mathcal{P}'_s$, then it also yields a linearly independent family for any $e \in \mathcal{P}_s$.

Lemma 3.3.12. *Let $e = (e_1, \dots, e_s) \in \mathcal{P}_s$ for $s \ge 2$. It we take f_i such that $\min\{e_i, \frac{s^2}{2} - s\} \le f_i \le \frac{s^2}{2} - 2$ for all i, then $f = (f_1, \dots, f_s) \in \mathcal{P}'_s$.*

Proof. We have $n'_i = n_i$ for $i \leq \frac{s^2}{2} - s + 1$, and $n'_i \leq s \leq i$ for $i \geq \frac{s^2}{2} - s + 2$. □

Corollary 3.3.13. *We define the following events on possible outcomes* $(a_1, \ldots, a_s)$*:*

$$\begin{array}{lll} A & = \{\bigwedge_{e \in \mathcal{P}_s} & \{(x-a_i)^{e_i} \ : 1 \leq i \leq s\} \quad \textit{is linearly independent}\} \\ B & = \{\bigwedge_{e \in \mathcal{P}'_s} & \{(x-a_i)^{e_i} \ : 1 \leq i \leq s\} \quad \textit{is linearly independent}\} \end{array}$$

Then $A = B$*.*

Proof. We first observe that if $a_1 = \cdots = a_s$, then $A = B$ trivially because $\{(x - a_i)^{e_i} \ : 1 \leq i \leq s\}$ is linearly independent for every $e_1, \ldots, e_s$, so let us assume that they are not all equal.
Since $\mathcal{P}'_s \subseteq \mathcal{P}_s$, we have $A \subseteq B$. Given an outcome $a = (a_1, \ldots, a_s) \in B$ and a sequence $e \in \mathcal{P}_s$, we can distinguish two cases. If $e \in \mathcal{P}'_s$, then the s shifted powers $(x - a_1)^{e_1}, \ldots, (x - a_s)^{e_s}$ are linearly independent since $a \in B$. Otherwise, assume there exists α_i such that $\sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i} = 0$. We denote by $I = \{i : e_i > s^2/2 - 2\}$, and, using Corollary 2.2.10, we have that $\alpha_i = 0$ for $i \in I$. We therefore rewrite the equality as $\sum_{i=1}^{s} \alpha_i (x - a_i)^{f_i} = 0$, where $f_i := e_i$ for $i \notin I$; otherwise, f_i is chosen in $\{s^2/2 - s, \ldots, s^2/2 - 2\}$ in such a way that there are no two equal (a_i, f_i) (we observe that we can always choose f_i in this interval since there are $s - 1$ possible values to choose from and there are at least two different a_i's). Using Lemma 3.3.12, we have that $f \in \mathcal{P}'_s$ and thus $(x - a_i)^{f_i} \ : 1 \leq i \leq s$ are linearly independent since $a \in B$, proving that all the α_i's are zero. □

Now that we have restricted our attention to a finite set $\mathcal{P}'_s$, we can directly use the union bound on all possible sequences $e \in \mathcal{P}'_s$ to obtain the result using Corollary 3.3.11 for a fixed sequence. We only need to obtain a upper bound on $|\mathcal{P}'_s|$; this is done in following proposition. More precisely, we will compute exactly $|\mathcal{P}_{s,d}|$ where $\mathcal{P}_{s,d} = \{e \in \mathcal{P}_s : \max(e_i) < d\}$.

Proposition 3.3.14. *Let* $s \leq d$ *be integers. Then*

$$|\mathcal{P}_{s,d}| = \binom{s+d}{s} \frac{d+1-s}{d+1}$$

Proof. Notice first that a sequence $e \in \mathcal{P}_{s,d}$ can be represented by the d-tuple $(m_1, \ldots m_d)$, where $m_i = |\{j : e_j = i - 1\}|$. Therefore, there is a bijection between $\mathcal{P}_{s,d}$ and the following set:

$$Q_{s,d} = \left\{ (m_1, \ldots, m_d) \in \mathbb{N}^d \ : \ \forall j \leq d, \sum_{i=1}^{j} m_i \leq j \bigwedge \sum_{i=1}^{d} m_i = s \right\}$$

For each $m \in Q_{s,d}$, we associate a lattice path on the Cartesian plane as follows: start the path at $(0, 0)$, and at the i-th step move right 1 unit then go up m_i units. The resulting path ends at position (d, s), and never goes above the diagonal $y = x$. In fact, there is a bijection between $Q_{s,d}$ and the monotonic lattice paths starting at

position $(0,0)$, ending at position (d,s), and not passing above the diagonal $y=x$. These numbers are usually called *ballot numbers*, and have been studied since de Moivre (1711). The analytic expression can be found in [49, p. 451], proving the result. □

In the Birkhoff interpolation paper [57] a similar proof technique was used to count the number of Pólya matrices.

Theorem 3.3.15. *Let S be a finite subset of $\mathbb{F}$. Let $a_1,\dots,a_s$ be selected at random independently and uniformly from S. Then*

$$\Pr\left[\bigwedge_{e\in\mathcal{P}_s}\left\{\{(x-a_i)^{e_i}\ :1\le i\le s\}\ \textit{is linearly independent}\right\}\right]\ge 1-\frac{f(s)}{|S|}$$

where $f(s)=\binom{s+\frac{s^2}{2}-1}{s}(s-1)(s-2)$.

Proof. Following the notation of Corollary 3.3.13, we have $\Pr(A)=\Pr(B)$. We compute $\Pr(B)$ using the union bound:

$$\Pr(B)\ge 1-\sum_{e\in\mathcal{P}'_s}\Pr\left(\{(x-a_i)^{e_i}\ :1\le i\le s\}\text{ is linearly dependent}\right)$$

By Corollary 3.3.11 we have $\Pr(B)\ge 1-|\mathcal{P}'_s|\cdot\frac{s(s-1)}{|S|}$. Using Proposition 3.3.14 with $d=\frac{s^2}{2}-1$, we obtain

$$\Pr(B)\ge 1-\binom{s+\frac{s^2}{2}-1}{s}\cdot\frac{\frac{s^2}{2}-s}{\frac{s^2}{2}}\cdot\frac{s(s-1)}{|S|}.$$

□

Remark 3.3.16. *One can improve the previous lower bound by noticing that for some sequences $e\in\mathcal{P}'_s$, we have $\Pr(\{(x-a_i)\}$ is linearly dependent$)=0$. This is the case for instance for any sequence e with distinct e_i's, thus we have*

$$\Pr(B)\ge 1-\left(|\mathcal{P}'_s|-\binom{s^2/2}{s}\right)\cdot\frac{s(s-1)}{|S|}.$$

3.4 Dimension lower bounds

As sufficient conditions for linear independence are hard to find, we now try to lower bound the dimension of the span of families of affine powers instead. Of course, this can only be easier since linear independence implies full dimension of the corresponding space. We mostly investigate two different cases: when all the exponents are big and when small exponents are allowed. In the latter case, we will require that the family F satisfies the *Pólya condition* (see Definition 3.3.1). Under this condition, we first prove an easy lower bound which holds over any field.

Proposition 3.4.1. *Consider any family* $F = \{(x-a_i)^{e_i} \ : \ 1 \leq i \leq s\}$ *satisfying the Pólya condition. Then we have* $\dim\langle F\rangle \geq \sqrt{s}$.

Proof. We partition F according to the values of the exponents: $F = \cup_{i=1}^{t} F_i$ with $F_i = \{(x-a_{i,j})^{d_i} \ : \ 1 \leq j \leq t_i\}$, and $d_i \neq d_j$ for $i \neq j$. The fact that F satisfies the Pólya condition implies that $d_i \geq t_i - 1$, and therefore that every F_i is an independent family, using Proposition 2.2.3.
Therefore, if there exists $k \in [|1;t|]$ such that F_k contains at least $\sqrt{s}$ elements we have directly $\dim\langle F\rangle \geq \dim\langle F_k\rangle = |F_k| \geq \sqrt{s}$. Otherwise, we must have $t \geq \sqrt{s}$. We consider a family G obtained by taking one element in each F_i. Since the d_i's are distinct, the elements of G are linearly independent. This proves that $\dim\langle F\rangle \geq |G| = t \geq \sqrt{s}$. □

In the following, we will show that we can achieve a linear lower bound for both real and complex field.

3.4.1 The real case

The following result is a consequence of Proposition 3.3.4

Proposition 3.4.2. *Consider any family* $F = \{(x-a_i)^{e_i} \ : \ 1 \leq i \leq s\}$, *with* $a_i \in \mathbb{R}$ *satisfying the Pólya condition.*
Then we have $\dim\langle F\rangle \geq \lfloor \frac{s+4}{3} \rfloor$.

Proof. Assume that $e_1 \geq e_2 \geq \cdots \geq e_s$ and consider the family $G := \{(x-a_i)^{e_i} \ : \ 1 \leq i \leq t\} \subset F$ with $t := \lfloor \frac{s+4}{3} \rfloor$. Since F satisfies the Pólya condition, we have that $e_{s-i} \geq i$ and, hence, $e_i \geq s - t$ for all $i \in \{1, \ldots, t\}$. The inequality $s - t \geq 2t - 4$ holds, thus we conclude by Proposition 3.3.4 that the elements of G are linearly independent and $\dim\langle F\rangle \geq \dim\langle G\rangle = |G| = t$. □

With more work and additional techniques from Birkhoff interpolation, we can in fact achieve an even better lower bound. Consider a family $F = \{(x-a_i)^{e_i} \ : 1 \leq i \leq s\}$. By a *sequence in the node* $a \in \mathbb{R}$ we mean a maximal interval of integers O such that for all $e \in O$, the element $(x-a)^e$ belongs to F. A sequence O with an odd size is naturally called an *odd sequence*. The following result is just a restatement of [29, Corollary 9.(ii)]; this result was obtained by transforming the problem of linear independence of shifted powers into an equivalent problem in Birkhoff interpolation, and then applying a celebrated result of Atkinson and Sharma [5] concerning real Birkhoff interpolation (see [58, Theorem 1.5]).

Corollary 3.4.3. *Consider a family* $F = \{(x-a_i)^{e_i} \ : \ 1 \leq i \leq s\}$, *with* $a_i \in \mathbb{R}$ *satisfying the Pólya condition and set* $d := \max(e_i)$. *If every odd sequence* O *in any node* a_i *satisfies that* $\max(O) = d$, *then the elements of* F *are linearly independent.*

Proposition 3.4.4. *Consider any family* $F = \{(x-a_i)^{e_i} \ : \ 1 \leq i \leq s\}$, *with* $a_i \in \mathbb{R}$ *satisfying the Pólya condition.*
Then we have $\dim\langle F\rangle \geq \lfloor \frac{s}{2} \rfloor + 1$.

Proof. Let us denote $d = \max(e_i)$. If there is only one i such that $d = e_i$ we have $\dim\langle F\rangle = \dim\langle F \setminus \{(x-a_i)^{e_i}\}\rangle + 1$. So, without loss of generality we assume that there are at least two e_i's equal to d. Let $O_1, \dots, O_k$ be all the odd sequences such that $d \notin O_i$; notice that $k \leq s-2$. We denote $m_i = \min(O_i)$, $M_i = \max(O_i)$ and call b_i the corresponding node; we order $O_1, \dots, O_k$ so that $m_1 \leq \dots \leq m_k$. We claim that $F' := F \setminus \{(x-b_i)^{m_i} : 1 \leq i \leq \lceil k/2 \rceil\}$ yields a linearly independent family. Since the size of F' is $s - \lceil k/2 \rceil \geq s - \lceil (s-2)/2 \rceil = \lfloor \frac{s}{2} \rfloor + 1$, this will prove that $\dim\langle F\rangle \geq \dim\langle F'\rangle \geq \lfloor \frac{s}{2} \rfloor + 1$.
More precisely, we are going to prove that the family $F' \cup \{(x-b_i)^{M_i+1} : \lceil k/2 \rceil + 1 \leq i \leq k\}$ is linearly independent. Indeed, by construction of F', every odd sequence O in this family satisfies $\max(O) = d$; this is because we have removed an element from $O_1, \dots, O_{\lceil k/2 \rceil}$ and added one to $O_{\lceil k/2 \rceil + 1}, \dots, O_k$, which converts all these odd sequences into even sequences. Moreover, the new set also satisfies the Pólya condition since for every $1 \leq i < j \leq k$, we have $m_i \leq m_j \leq M_j + 1$, so removing a shifted power of exponent m_i and adding one of exponent $M_j + 1$ can never cause a violation of the Pólya condition. By Corollary 3.4.3 we are done. □

It is worth pointing out that this result does not only bound the dimension of $\langle F\rangle$ but also shows how to explicitly obtain a linearly independent subset of F of size at least $\lfloor \frac{s}{2} \rfloor + 1$. This will not be the case in the complex setting, where we will obtain a lower bound on the dimension of $\langle F\rangle$ but we will not provide an explicit linearly independent subset of F of size $\Omega(s)$.

We do not know if the above bound is sharp. In the following example we exhibit a family F of s shifted powers that satisfy the Pólya condition and we prove that $\dim\langle F\rangle = (3s-1)/4$.

Lemma 3.4.5. *Let $d \in \mathbb{Z}^+$ such that $d \equiv 2 \pmod 4$ and consider $F_1 := \{x^i : i$ odd and $i < d\}$ and $F_2 := \{(x+1)^i, (x-1)^i : i$ even and $(d+2)/2 \leq i \leq d\}$. The family $F := F_1 \cup F_2$ has $d+1$ elements, satisfies the Pólya condition and $\dim\langle F\rangle = (3d+2)/4$.*

Proof. It is easy to see that $|F_1| = d/2$, $|F_2| = (d+2)/2$ and that F satisfies the Pólya condition. We remark that F has an element of degree i for all $i \in \{1, 3, \dots, d/2, (d/2)+1, \dots, d\}$. This implies $\dim\langle F\rangle \geq (3d+2)/4$. Moreover, for every i even such that $(d+2)/2 \leq i \leq d$ we observe that $(x+1)^i - (x-1)^i = \sum_{\substack{j<i \\ j \text{ odd}}} 2\binom{i}{j} x^j \in \langle F_1\rangle$. Hence, we can combine every pair elements of F_2 of the same degree so that the combination can be expressed as a linear combination of the elements of F_1. This proves that $\dim\langle F\rangle \leq |F| - \frac{|F_2|}{2} = d + 1 - \frac{d+2}{4} = \frac{3d+2}{4}$. □

3.4.2 The complex case

In the complex case, our results rely on Corollary 3.1.7 and on a good choice of parameters in Proposition 3.1.3.

Proposition 3.4.6. *Fix an integer $p \in \mathbb{N}$ and $\alpha > 0$ and consider any family $F = \{(x-a_i)^{e_i} \ : \ 1 \leq i \leq p\}$ with $e_i \geq \alpha p$ for all i.*
Then we have $\dim\langle F\rangle > (1+\alpha-\sqrt{\alpha^2+1})p$.

Proof. We extract a basis $G \subseteq F$ of $\langle F\rangle$. We set $s := |G|$, $\gamma := (1+\alpha-\sqrt{\alpha^2+1})p$ and we assume for contradiction that $s \leq \gamma$. We take $t = s$, $k = \alpha p$ and $l = p-1$, we claim that the inequality in Proposition 3.1.3 is satisfied. Indeed, according to Proposition 3.1.3, it suffices to prove that the polynomial function

$$\begin{aligned} p(S) &= S^2 - (2l+2k+3)S + 2(k+1)(l+1) \\ &= S^2 - (2p+2\alpha p+1)S + 2p(\alpha p+1) \end{aligned}$$

is positive for $S = s$. We will prove that $p(S) > 0$ for all $S \in [0,\gamma]$. For this purpose, we consider

$$q(S) \ := \ p(S) + S - 2p \ = \ S^2 - (2p+2\alpha p)S + 2\alpha p^2;$$

it is straightforward to check that $q(S) \geq 0$ for all $S \in [0,\gamma]$ since γ is the smallest root of q. We conclude that $p(S) > 0$ in $[0,\gamma]$ because the following inequalities hold for every $S \in [0,\gamma]$:

$$p(S) = q(S) - S + 2p \geq q(S) - \gamma + 2p = q(S) + (\sqrt{\alpha^2+1} - \alpha + 1)p > q(S).$$

Hence, by Proposition 3.1.3, there exists an SDE$(s, k', p-1)$ with $s \leq k' \leq \alpha p$ satisfied by all the elements of G. Since G is a basis of $\langle F\rangle$, it follows that not only the elements of G are solutions of this SDE, but also the elements of F. Since for any $i \in [|1;p|]$ we have $e_i \geq \alpha p \geq k'$, it follows from Corollary 3.1.7 that $p \leq \deg P_{k'} \leq p-1$, a contradiction. □

Corollary 3.4.7. *Fix an integer $s \in \mathbb{N}$ and consider any family $F = \{(x-a_i)^{e_i} \ : \ 1 \leq i \leq s\}$ with $e_i \geq s$ for all i.*
Then we have $\dim\langle F\rangle > (2-\sqrt{2})s$.

We now extend Proposition 3.4.6 by allowing small exponents.

Theorem 3.4.8. *For any family $F = \{(x-a_i)^{e_i} \ : \ 1 \leq i \leq s\}$ satisfying the Pólya condition, we have* $\dim\langle F\rangle \geq (1-\frac{\sqrt{2}}{2})(s-1)$.

Proof. We start by dropping the $\lceil \frac{s}{2} \rceil$ terms with least exponents and call F' the resulting family. By construction we have $|F'| = s - \lceil \frac{s}{2} \rceil = \lfloor \frac{s}{2} \rfloor$, and because F satisfies the Pólya condition, every element $(x-a)^e \in F'$ must verify $e \geq \lceil \frac{s}{2} \rceil$. Thus the family F' satisfies the hypothesis of Corollary 3.4.7, implying that $\dim\langle F'\rangle \geq (2-\sqrt{2})\lfloor \frac{s}{2} \rfloor$. The result follows since $\langle F'\rangle$ is a subset of $\langle F\rangle$. □

4

Reconstruction algorithms

In this chapter, we design algorithms that reconstruct the optimal representation of polynomials in Model 2, i.e., algorithms that receive as input $f \in \mathbb{F}[x]$ and compute the exact value $s = \text{AffPow}_{\mathbb{F}}(f)$ and a set of triplets of coefficients, nodes and exponents $\{(\alpha_i, a_i, e_i) \,|\, 1 \leq i \leq s\} \subseteq \mathbb{F} \times \mathbb{F} \times \mathbb{N}$ such that $f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}$. We assume that f is given in dense representation, i.e. as a tuple of $\deg(f) + 1$ elements of $\mathbb{F}$, and, unless specified otherwise, we will measure the complexity in term of arithmetic operations in the underlying field: addition, multiplication and root finding. We will not be able to solve the problem in all its generality but under certain hypotheses. One typical result is as follows (see Theorem 4.1.4 in Section 4.1 for a more detailed statement which includes a description of the algorithm).

Theorem 4.0.1. *Let $f \in \mathbb{F}[x]$ be a polynomial that can be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i},$$

where the constants $a_i \in \mathbb{F}$ are all distinct, $\alpha_i \in \mathbb{F} \setminus \{0\}$, and $e_i \in \mathbb{N}$. Assume moreover that $n_{i+1} \leq (3i/4)^{1/3} - 1$ for all $i \geq 2$, where $n_i \stackrel{def}{=} \#\{j : e_j < i\}$.
Then, $\text{AffPow}_{\mathbb{F}}(f) = s$. Moreover, there is a polynomial time algorithm that receives $f = \sum_{i=0}^{d} f_i x^i \in \mathbb{F}[x]$ as input and computes the s-tuples of coefficients $C(f) = (\alpha_1, \ldots, \alpha_s)$, of nodes $N(f) = (a_1, \ldots, a_s)$ and exponents $E(f) = (e_1, \ldots, e_s)$.

As already pointed out in Section 2.1, it is quite natural to assume an upper bound on the numbers n_i from the point of view of the optimality of representations. It would nonetheless be interesting to relax the assumption $n_{i+1} \leq (3i/4)^{1/3} - 1$ in this theorem. Another restriction is the assumption that the shifts a_i are all distinct. We relax that assumption in Section 4.2 but we still need to assume that all the exponents e_i corresponding to a given shift $a_i = a$ belong to a "small" interval (see Theorem 4.2.6 for a precise statement). Alternatively, we can assume instead that there is a large gap between the exponents in two consecutive occurrences of the same shift as in Theorem 4.2.11.

It turns out that our reconstruction algorithms only work in a regime where the uniqueness of optimal representations is guaranteed. As such, we will repeatedly use the results from Section 2.2.2 to ensure that this is the case. For conciseness, we will usually talk about "the" optimal expression of f even before ensuring the uniqueness of the optimal expressions. It would be interesting to devise algorithms that don't require the uniqueness of optimal representations, but this seems out of reach with the current techniques.

The main tools of this chapter are the SDEs, defined in Section 3.1.1. As we will always take $t = k+1$ in the following, we will constantly use the notation $\text{SDE}(k, l)$ to denote a $\text{SDE}(k+1, k, l)$ for conciseness. When f is a polynomial with an expression of size s in Model 2, we proved in Proposition 3.1.3 that f satisfies a $\text{SDE}_{\leq}(2s-1, 0)$. The basic idea behind our algorithms is to look for one of these "small" SDEs satisfied by f, and hope that the powers $(x - a_i)^{e_i}$ in an optimal decomposition of f

satisfy the same SDE. This isn't just wishful thinking because the SDE from Proposition 3.1.3 is satisfied not only by f but also by the powers $(x-a_i)^{e_i}$.
Unfortunately, this basic idea by itself does not yield efficient algorithms. The main difficulty is that f could satisfy several SDE of order $2s-1$ and shift 0. By Remark 3.1.2 we can efficiently find such a SDE, but what if we don't find the "right" SDE, i.e., the SDE which (by Proposition 3.1.3) is guaranteed to be satisfied by f *and* by the powers $(x-a_i)^{e_i}$? Proposition 3.2.4 will be our main tool to overcome this difficulty as it gives some sufficient conditions on the exponents to ensure that the powers also satisfy a given SDE.

4.1 Algorithms for distinct nodes

This section concerns the case where the a_i in the optimal expression of f are all distinct. In this setting, our main result is Theorem 4.1.4 where we solve the problem when the number n_e of exponents in the optimal expression that are $< e$ is "small". We first study the simpler case where all the exponents are large enough and give a bitsize analysis of the devised algorithm, before moving to more general settings.

4.1.1 Big exponents

As a consequence of Proposition 3.2.4, we get Corollary 4.1.1 and Theorem 4.1.2. They provide an effective method to obtain the optimal expression of a polynomial f in Model 2 whenever all the terms involved have big exponents and all the nodes are different.

Corollary 4.1.1. *Let $f \in \mathbb{F}[x]$ be written as $f = \sum_{i=1}^{s} \alpha_i (x-a_i)^{e_i}$, with $\alpha_i \in \mathbb{F} \setminus \{0\}$, $a_i \in \mathbb{F}$ all distinct, and $e_i \geq 5s^2/2$ for all i. Then,*

a) $\{(x-a_i)^{e_i} \mid 1 \leq i \leq s\}$ are linearly independent,

b) If $f = \sum_{i=1}^{t} \beta_i (x-b_i)^{d_i}$ with $t \leq s$, then $t = s$ and we have the equality $\{(\alpha_i, a_i, e_i) \mid 1 \leq i \leq s\} = \{(\beta_i, b_i, d_i) \mid 1 \leq i \leq s\}$; in particular, $\mathit{AffPow}_{\mathbb{F}}(f) = s$,

c) f satisfies a SDE$(2s-1, 0)$,

d) if f satisfies a SDE$(k, 0)$ with $k \leq 2s-1$ then $(x-a_i)^{e_i}$ also satisfies it for all $i \in \{1, \ldots, s\}$, and

e) f does not satisfy any SDE$(k, 0)$ with $k < s$.

Proof. Notice first that (b) implies (a). Assume now that (b) does not hold, then there is another expression of f as $f = \sum_{i=1}^{t} \beta_i (x-b_i)^{d_i}$ with $t \leq s$. Hence, by Theorem 2.2.5, we get that

$$2s \geq t + s > \sqrt{2(\min(\{e_1, \ldots, e_s\}) + 1)} \geq \sqrt{5s^2},$$

a contradiction. From Proposition 3.1.3 we get (c). Assume f satisfies a SDE$(k,0)$ with $k \leq 2s-1$. For all $i \in \{1,\ldots,s\}$ we have that

$$e_i \geq 5s^2/2 \geq (2s-1)s + \binom{s}{2} \geq ks + \binom{s}{2},$$

and therefore Proposition 3.2.4 yields that $(x-a_i)^{e_i}$ is also a solution of this equation for all i, proving (d). Finally, f cannot satisfy a SDE$(k,0)$ with $k < s$; otherwise by (a) and (d), the vector space of solutions to this equation has dimension $\geq s$, which contradicts Lemma 3.1.4. □

Theorem 4.1.2 (Big exponents)**.** *Let $f \in \mathbb{F}[x]$ be a polynomial that can be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x-a_i)^{e_i},$$

where the constants $a_i \in \mathbb{F}$ are all distinct, $\alpha_i \in \mathbb{F} \setminus \{0\}$ and $e_i > 5s^2/2$. Then, $\mathit{AffPow}_{\mathbb{F}}(f) = s$. Moreover, there is a polynomial time algorithm `Build`(f) *that receives the polynomial $f = \sum_{i=0}^{d} f_i x^i \in \mathbb{F}[x]$ as input and computes the s-tuples of coefficients $C(f) = (\alpha_1,\ldots,\alpha_s)$, of nodes $N(f) = (a_1,\ldots,a_s)$ and exponents $E(f) = (e_1,\ldots,e_s)$. The algorithm* `Build`(f) *works as follows:*

Step 1. *Take r the minimum value such that f satisfies a SDE$(r,0)$ and compute explicitly one of these SDE.*

Step 2. *Compute $B = \{(x-b_i)^{d_i} \mid 1 \leq i \leq l\}$, the set of all the solutions of the SDE of the form $(x-b)^e$ with $(r+1)^2/2 \leq e \leq \deg(f) + (r^2/2)$.*

Step 3. *Determine $\beta_1,\ldots,\beta_l$ such that $f = \sum_{i=1}^{l} \beta_i (x-b_i)^{d_i}$*

Step 4. *Set $I := \{i \mid \beta_i \neq 0\}$ and output the sets $C(f) = (\beta_i \mid i \in I)$, $N(f) = (b_i \mid i \in I)$ and $E(f) = (d_i \mid i \in I)$.*

Proof. Corollary 4.1.1 proves the correctness of this algorithm. Indeed, by Corollary 4.1.1.(c) and (e), the value r computed in **Step 1** satisfies that $s \leq r \leq 2s-1$. We claim that the set B computed in **Step 2** satisfies that:

(1) it contains the set $\{(x-a_i)^{e_i} \mid 1 \leq i \leq s)\}$,

(2) it has at most r elements, and

(3) all its elements are $\mathbb{F}$-linearly independent.

The first claim follows from Corollary 4.1.1.(d), the fact that $(r+1)^2/2 \leq (2s)^2/2 < 5s^2/2$, and from Corollary 2.2.10, since $e_i \leq \deg(f) + (s^2/2) \leq \deg(f) + (r^2/2)$ for all i. To prove the second claim assume that B has more than r elements, then we take $t_1,\ldots,t_{r+1} \in B$. To reach a contradiction, by Lemma 3.1.4 it suffices to prove that $t_1,\ldots,t_{r+1}$ are linearly independent. If this were not the case, by

Theorem 2.2.5, we would have that $r+1 > \sqrt{(r+1)^2+2}$, which is not possible. A similar argument and the fact that B has at most r elements proves the third claim. By (1) and (3), the expression of f as a combination of the elements of B is unique and is the desired one.
Finally, the four steps can be performed in polynomial time. Only the first two steps require a justification. See Remark 3.1.2 in Section 3.1.1 regarding Step 1. In Step 2 we substitute for each value of e the polynomial $(x-b)^e$ in the SDE. This yields a polynomial $g(x)$ whose coefficients are polynomials in b of degree at most $r \leq 2s-1$. We are looking for the values of b which make g identically 0, so we find b as a root of the gcd of the coefficients of g. □

In the following result we are going to analyze the bitsize complexity of the algorithm proposed; for this purpose we assume that the output (and, hence, the input) have integer coefficients. With this analysis we intend to show a rough overestimate on the number of bitsize operations showing the polynomial time nature of the algorithm.

We recall that by the dense size of a polynomial $f = \sum_{i=0}^{d} f_i x^i \in \mathbb{Z}[x]$ we mean $size(f) := \sum_{i=0}^{d}[1+\log_2(1+|f_i|)]$. Also for an $n \times m$ matrix M with rational entries p_{ij}/q_{ij} where $p_{ij} \in \mathbb{Z}$, $q_{ij} \in \mathbb{Z}^+$, the bit size of M is $size(M) := \sum_{i=1}^{n}\sum_{j=1}^{m}[1 + \log_2(1+|p_{ij}|)+\log_2(1+q_{ij})]$. The notation $f(n) = \overline{\mathcal{O}}(g(n))$ means that there exists a $k \in \mathbb{N}$ such that $f(n) = \mathcal{O}(g(n)\log^k(\max(|g(n)|, 2)))$.

Proposition 4.1.3. *Let f be a polynomial of degree d that can be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x-a_i)^{e_i},$$

where the constants $a_i \in \mathbb{Z}$ are all distinct, $\alpha_i \in \mathbb{Z} \setminus \{0\}$ and $e_i > 5s^2/2$. The algorithm `Build`(f) *in Theorem 4.1.2 outputs the optimal expression of f in Model 2 in $\overline{\mathcal{O}}(d^{6.5} size(f) + d^8)$ bit size operations.*

Proof. A first observation is that the value r computed in **Step 1** of the algorithm is upper bounded in terms of d. Indeed, by hypothesis $5s^2/2 \leq \max(e_i)$ and, by Corollary 2.2.10, $\max(e_i) \leq d + (s^2/2)$, which implies that $d \geq 2s^2$. Moreover, in Corollary 4.1.1 we show that $r \leq 2s-1$; this gives $r = \mathcal{O}(\sqrt{d})$. Also by Corollary 4.1.1, we have that $s \leq r$ and then $\max(e_i) = O(d)$.
Let us study now the number of bit size operations needed to obtain a SDE$(r, 0)$ satisfied by f assuming that we know in advance the value of r in **Step 1** of the algorithm. We propose to follow the idea of Remark 3.1.2 and find the SDE by computing a vector in the kernel of the matrix M whose entries are the coefficients of the polynomials $x^j f^{(i)}$ with $0 \leq j \leq i \leq r$. We have that M has $1 + \cdots + r + (r+1) = (r+1)r/2$ rows and $d+1$ columns. Since $size(x^j f^{(i)}) = \mathcal{O}(size(f) + id\log(d))$, we have that $size(M) = \mathcal{O}(\sum_{i=0}^{r}(i+1)(size(f) + id\log(d))) = \mathcal{O}(r^2(size(f) + rd\log(d)))$, which is $\overline{\mathcal{O}}(d\, size(f) + d^{2.5})$. Now, we can obtain the required SDE by means of the Gauss pivoting method on M. Let E be the matrix in echelon form obtained by the

Gauss method. By [65, Theorem 3.3], to compute E one needs $\mathcal{O}(r^4 d)$ arithmetic operations, which is $O(d^3)$, and the biggest size of a coefficient appearing during the process of elimination by pivoting is $\mathcal{O}(size(M))$. Thus, the number of bit size operations needed to obtain the SDE$(r, 0)$ is $\overline{\mathcal{O}}(d^3\, size(M))$. Also, the biggest size of a coefficient appearing in the SDE$(r, 0)$ found is $\mathcal{O}(size(M))$. After multiplying by an appropriate integer, we can assume that each of these coefficients are integers of size $\mathcal{O}(size(M))$.

We now lift the assumption that r is known in advance. To perform **Step 1** we follow Remark 3.1.2 and we check whether f satisfies a SDE$(\ell, 0)$ starting from $\ell = 0$ and increasing ℓ. We observe that at each step, we can check if f satisfies a SDE$(\ell, 0)$ by checking if the matrix M_ℓ whose rows are the coefficients of the polynomials $x^i f^{(j)}$ with $0 \leq i \leq j \leq \ell$ has full row rank. This can be easily checked from the matrix E_ℓ in echelon form obtained by applying the Gauss method to M_ℓ. Since M_ℓ and E_ℓ are respectively submatrices of $M_{\ell+1}$ and $E_{\ell+1}$, the procedure of computing the SDE of smallest order satisfied by f can be done incrementally. Moreover, all the matrices M_ℓ and E_ℓ are submatrices of the matrices M and E described above. So, it is interesting to notice that knowing the exact value of r in advance does not give any advantage and **Step 1** can be performed in $\overline{\mathcal{O}}(d^3 size(M))$ bit size operations.

To perform **Step 2** we propose the following strategy. Assume that the SDE obtained in **Step 1** is $\sum_{i=0}^{r} P_i(x) f^{(i)}(x) = 0$. For each value e such that $(r+1)^2/2 \leq e \leq d + (r^2/2)$, we input in the SDE the polynomial $(x - Y)^e$, where Y is a new variable; we obtain an equation of the form $g(x, Y) = 0$. We first observe that $g(x, Y) = (x - Y)^{e-r} h(x, Y)$, where $h(x, Y) \in \mathbb{Z}[x, Y]$ has degree $\leq r$. We write $h = \sum_{i=0}^{r} h_i x^i$, where $h_i \in \mathbb{Z}[Y]$ is of degree $\leq r - i$.

The bit size of any coefficient of h_i is $\overline{\mathcal{O}}(r^2\, size(M))$, which is $\overline{\mathcal{O}}(d\, size(M))$. Moreover, since every $h_i \in \mathbb{Z}[Y]$ has degree $\leq r$, by [10, Proposition 21.22], the cost of computing the integer roots of each h_i is $\overline{\mathcal{O}}(d^2\, size(M))$. Since we have to solve $r + 1$ equations and take the common roots, this makes $\overline{\mathcal{O}}(d^{2.5} size(M))$ bit size operations and since we have to do it for at most d values of e, this gives $\overline{\mathcal{O}}(d^{3.5} size(M))$ bit operations overall. Moreover, the b_j's computed divide the independent term of all the h_i and, hence, the bit size of each b_i is $\overline{\mathcal{O}}(d\, size(M))$.

In **Step 3**, the corresponding matrix has at most $d + 1 + (r^2/2)$ rows, at most r columns (see the proof of Theorem 4.1.2), and its size is $\overline{\mathcal{O}}(d^{3.5} size(M))$. Since the rank of this matrix is $\leq r$ (indeed, as we proved in Theorem 4.1.2, this matrix has full column rank), when we are performing Gaussian elimination and treating a new row we have at most r already treated nonzero rows. As a consequence of this, we have to perform $\mathcal{O}(r^2 d)$ arithmetic operations to solve the system of equations by Gaussian elimination through pivoting. Hence the cost of this step is $\overline{\mathcal{O}}(d^{5.5} size(M))$, giving a total cost of $\overline{\mathcal{O}}(d^{6.5} size(f) + d^8)$ bit size operations. □

4.1.2 Low rank

We now lift the assumption on all the exponents being large and proceed with the main result of this section:

Theorem 4.1.4 (Different nodes)**.** *Let $f \in \mathbb{F}[x]$ be a polynomial that can be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i},$$

where the constants $a_i \in \mathbb{F}$ are all distinct, $\alpha_i \in \mathbb{F} \setminus \{0\}$, and $e_i \in \mathbb{N}$. Assume moreover that $n_{i+1} \leq (3i/4)^{1/3} - 1$ for all $i \geq 2$, where $n_i \stackrel{def}{=} \#\{j : e_j < i\}$.
Then, $\mathrm{AffPow}_{\mathbb{F}}(f) = s$. Moreover, there is a polynomial time algorithm $\mathtt{Build}(f)$ that receives $f = \sum_{i=0}^{d} f_i x^i \in \mathbb{F}[x]$ as input and computes the s-tuples of coefficients $C(f) = (\alpha_1, \ldots, \alpha_s)$, of nodes $N(f) = (a_1, \ldots, a_s)$ and exponents $E(f) = (e_1, \ldots, e_s)$. The algorithm $\mathtt{Build}(f)$ works as follows:

Step 1. *We take t the minimum value such that f satisfies a SDE$(t, 0)$ and compute explicitly one of these SDE.*

Step 2. *Consider $B := \{(x - b_i)^{d_i} \mid 1 \leq i \leq l\}$, the set of all the solutions of the SDE of the form $(x - b)^e$ with $(t+1)^2/2 \leq e \leq \deg(f) + \frac{(\deg(f)+2)^2}{8}$ and assume that $d_1 \geq d_2 \geq \cdots \geq d_l \geq d_{l+1} := (t+1)^2/2$.*

Step 3. *We take $r \in \{1, \ldots, l\}$ such that $d_r - d_{r+1} > r^2/2$ and $d_{r+1} < \deg(f)$.*

Step 4. *We set $j := d_r - (r^2/2)$ and express $f^{(j)}$ as $f^{(j)} = \sum_{i=1}^{r} \beta_i \frac{d_i!}{(d_i - j)!}(x - b_i)^{d_i - j}$ with $\beta_1, \ldots, \beta_r \in \mathbb{F}$. We set $I := \{i \mid \beta_i \neq 0\}$.*

Step 5. *We set $\widetilde{f} := \sum_{i=1}^{r} \beta_i (x - b_i)^{d_i}$ and $h := f - \widetilde{f}$.*
If $h = 0$, then $C(f) = (\beta_i \mid i \in I)$, $N(f) = (b_i \mid i \in I)$ and $E(f) = (d_i \mid i \in I)$.
Otherwise, we set $h := f - \widetilde{f}$ and we have that $C(f) = (\beta_i \mid i \in I) \cup C(h)$, $N(f) = (b_i \mid i \in I) \cup N(h)$ and $E(f) = (d_i \mid i \in I) \cup E(h)$, where the triplet $(C(h), N(h), E(h))$ is the output of $\mathtt{Build}(h)$.

Proof. By Corollary 2.2.8 we have that $\mathrm{AffPow}_{\mathbb{F}}(f) = s$. Concerning the algorithm, first we observe that the value t computed in **Step 1** is $\leq 2s - 1$ by Proposition 3.1.3. Moreover, we claim that the set B computed in **Step 2** has $l \leq t$ elements. Otherwise, by Lemma 3.1.4, there exists a set $I \subseteq [\![1, l]\!]$ of size $\leq t + 1$ and there exist $\{\gamma_i \mid i \in I\} \subseteq \mathbb{F} \setminus \{0\}$ such that we have $\sum_{i \in I} \gamma_i (x - b_i)^{d_i} = 0$. Setting $m := \max\{d_i \mid i \in I\} \geq (t+1)^2/2$, Theorem 2.2.5 yields that $t + 1 \geq |I| > \sqrt{2(m+1)} > t + 1$, a contradiction.
Now we set $L := 5s^2/2$ and consider the set $C := \{(x - a_i)^{e_i} \mid e_i \geq L\}$ where the a_i's are the nodes in the optimal expression of f. We have that $C \neq \emptyset$; indeed, if we set $e_{\max} := \max\{e_i \mid 1 \leq i \leq s\}$, then $s = n_{e_{\max}+1} \leq (3e_{\max}/4)^{1/3} - 1$ and therefore $L \leq 4(s+1)^3/3 \leq e_{\max}$.
By Corollary 2.2.10 we know that $e_i \leq \deg(f) + \frac{(\deg(f)+2)^2}{8}$ for all $i \in [\![1, s]\!]$, and Proposition 3.2.4 yields that all the elements of C are solution of the SDE since

$$ts + \binom{s}{2} \leq (2s - 1)s + \binom{s}{2} \leq 5s^2/2.$$

Therefore, we have $C \subseteq B$ and in particular, there exists $\tau \in [\![1, l]\!]$ such that $d_1 \geq d_\tau = e_{\max} \geq \frac{4}{3}(s+1)^3$.
Now we take $k := \max\{i \mid d_i > L\}$ (we have that $1 \leq k \leq l \leq t \leq 2s-1$) and we are going to prove that

- there exists $r \in \{\tau, \ldots, k-1\}$ such that $d_r - d_{r+1} > r^2/2$, or
- $d_k - L > k^2/2$.

Indeed, if this is not the case, then we get the following contradiction:

$$\begin{aligned}\tfrac{4s^3}{3} \leq\ & \tfrac{4(s+1)^3}{3} - L \leq e_{\max} - L = d_\tau - L = \textstyle\sum_{i=\tau}^{k-1}(d_i - d_{i+1}) + d_k - L \leq \\ \leq\ & \tfrac{1}{2}\textstyle\sum_{i=\tau}^{k} i^2 \leq \tfrac{1}{2}\sum_{i=1}^{k} i^2 = \tfrac{k(k+1)(2k+1)}{12} \leq \tfrac{(2s-1)2s(4s-1)}{12} < \tfrac{4s^3}{3}.\end{aligned}$$

We take $r \in \{1, \ldots, k-1\}$ such that $d_r - d_{r+1} > r^2/2$, or $r = k$ if such a value does not exist (and $d_k - L > k^2/2$). We claim that $e_{\max} \geq d_r$ if and only if $d_{r+1} < \deg(f)$ and, thus, the r described in **Step 3** always exists. If $d_{r+1} < \deg(f)$, since $\deg(f) \leq e$ and $C \subseteq B$, then $d_r \leq e_{\max}$ (since $e_{\max} = d_\tau$, it cannot be sandwiched between two consecutive elements d_r, d_{r+1} of this sequence).
Conversely, assume now that $e_{\max} \geq d_r$ and let us prove that $d_{r+1} < \deg(f)$. To prove this we first observe that setting $j := d_r - (r^2/2)$, then $f^{(j)}$ can be uniquely expressed as a linear combination of $B' := \{(x-b_i)^{d_i - j} \mid 1 \leq j \leq r\}$. Indeed, $f^{(j)} = \sum_{e_i \geq j} \alpha_i \frac{e_i!}{(e_i-j)!}(x-a_i)^{e_i-j}$ with $\alpha_i \neq 0$ and $(x-a_i)^{e_i-j} \in B'$ for all $e_i \geq j$, and if there is another way of expressing $f^{(j)}$ as a linear combination of B', then by Theorem 2.2.5 we get that $r > \sqrt{2(\min\{d_i \mid 1 \leq i \leq r\} - j + 1)} \geq \sqrt{r^2+2} > r$, a contradiction. So, if $d_{r+1} \geq \deg(f)$, then $f^{(j)} = 0$ and the only expression of $f^{(j)}$ as a linear combination of B' would be the one in which every coefficient is 0, a contradiction. Hence, the value r computed in **Step 3** exists.
We have seen that $f^{(j)}$ can be uniquely expressed as a linear combination of B' as $f^{(j)} = \sum_{e_i \geq j} \alpha_i \frac{e_i!}{(e_i-j)!}(x-a_i)^{e_i-j}$. Hence, in **Step 4**, one finds all the (α_i, a_i, e_i) such that $e_i \geq j$. In **Step 5**, either the polynomial h is 0 and we have finished or $h = \sum_{e_i < j} \alpha_i (x-a_i)^{e_i}$ is written as a linear combination of strictly less than s terms and satisfies the hypotheses of the Theorem, so by induction we are done. □

Note that in Step 2 of this algorithm we need to compute polynomial roots, just as in the corresponding step of Theorem 4.1.2 (see the proof of Theorem 4.1.2 for details). One difference, however, is that we do not use the roots b_i only to output the coefficients of the optimal decomposition: we also use the b_i in the subsequent iterations of the algorithm since the polynomials $\tilde{f}$ and h of Step 5 are defined in terms of the b_i, and we call the algorithm recursively on input h. From this discussion one might be lead to think that if f has its coefficients in a subfield $\mathbb{F}$ of $\mathbb{K}$, the coefficients of $\tilde{f}$ and h may lie outside $\mathbb{F}$. Proposition 2.2.9 implies that this is not the case: in the last paragraph of the proof of Theorem 4.1.4, we proved that for all $e \in \mathbb{N}$, either all the affine powers with exponents e of the optimal expression are in $\tilde{f}$ or none of them are, proving that $\tilde{f}$ and therefore $h = f - \tilde{f}$ always lie in $\mathbb{F}[x]$.

However, we do not know if $\tilde{f}$ can be computed from f with a polynomial number of arithmetic operations and comparisons.

We define the size of the set of triplets $\{(\alpha_i, a_i, e_i) \mid 1 \leq i \leq s\} \subset \mathbb{Z} \times \mathbb{Z} \times \mathbb{N}$ as $\sum_{i=1}^{s}[1 + \log_2(1 + |a_i|) + \log_2(1 + |\alpha_i|) + e_i]$. It is not clear that the size of the output of the algorithm proposed in Theorem 4.1.4 is polynomially bounded in the input size (i.e., in the bit size of f given as a sum of monomials) because it is an iterative algorithm. Indeed, at each step of the iteration, we have to compute roots of polynomials (which may lie outside $\mathbb{F}$), and we keep computing with these roots in the subsequent iterations. Since the number of iterations can be a priori as large as s, a naive analysis will only give an exponential upper bound on the bit size of the output. It is in fact not clear that there exists a solution of size polynomially bounded in the input size (i.e., in the bit size of f given as a sum of monomials). More precisely, we ask the following question.

Question 4.1.5. *We define the dense size of a polynomial $f = \sum_{i=0}^{d} f_i x^i \in \mathbb{Z}[x]$ as $\sum_{i=0}^{d}[1 + \log_2(1 + |f_i|)]$. Assume that f can be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i},$$

with $a_i \in \mathbb{Z}$, $\alpha_i \in \mathbb{Z} \setminus \{0\}$, and that this decomposition satisfies the conditions of Theorem 4.0.1: the constants a_i are all distinct, and $n_{i+1} \leq (3i/4)^{1/3} - 1$ for all $i \geq 2$.
Is it possible to bound the bit size of the constants α_i, a_i by a polynomial function of the dense size of f ?

Yet, it is straightforward to check that the input size is polynomially bounded by the output size. Indeed, the degree of f is upper bounded by the maximum value of the e_i and every coefficient of f can be seen as the evaluation of a small polynomial in the α_i, a_i's. In the following result we prove that the algorithm works in polynomial time in the size of the output. Hence, a positive answer to Question 4.1.5 together with Corollary 2.2.10 would directly yield that the algorithm works in polynomial time (in the size of the input).

Proposition 4.1.6. *Let $f \in \mathbb{Z}[x]$ be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i},$$

with $a_i \in \mathbb{Z}$, $\alpha_i \in \mathbb{Z} \setminus \{0\}$, $e_i \in \mathbb{N}$ and assume that this decomposition satisfies the conditions of Theorem 4.1.4: the constants a_i are all distinct, and $n_{i+1} \leq (3i/4)^{1/3} - 1$ for all $i \geq 2$.
Then, the algorithm in Theorem 4.1.4 works in polynomial time in the size of the output.

Proof. We write $f = \sum_{j=0}^{d} f_j x^j$ with $f_j \in \mathbb{Z}$ and $d = \deg(f) \leq \max\{e_1, \ldots, e_s\}$. We have that $f_j = \sum_{e_i \geq j} \alpha_i \binom{e_i}{j} a_i^{e_i - j}$ for all $j \in \{0, \ldots, d\}$. Thus, the size of f is polynomially bounded by the size of the output. To perform **Step 1** we follow Remark 3.1.2. We note that the coefficients of the polynomials appearing in the SDE are polynomially bounded by the size of f. In **Step 2** we have to compute the integral roots of polynomials of degree $t \leq s$ with integral coefficients, which can also be done in polynomial time (see, e.g., [56]). **Step 4** can also be performed in polynomial time by solving a linear system of equations (see, e.g., [65, Corollary 3.3a]). The result follows from the fact that the polynomial h defined in **Step 5** can be written as $h = \sum_{j \in J} \alpha_j (x - a_j)^{e_j}$ for some set $J \subset \{1, \ldots, s\}$ of at most $s - 1$ elements. After the first iteration, the algorithm is therefore called recursively on polynomials h with an output size bounded by the output size of the original f. □

4.2 Algorithms for repeated nodes

This section concerns the case where the nodes a_i in the optimal expression of f in Model 2 are not necessarily different. This case is more involved since we can no longer apply Proposition 3.2.4. As a consequence, we don't know if an analogue of Corollary 4.1.1 holds for repeated nodes, even with a bigger lower bound on the exponents. To overcome this difficulty, we consider two scenarios with additional constraints, whose study naturally divides this section in two parts. In the first subsection we provide algorithms when all the exponents corresponding to a repeated node appear in a small interval. The second one handles the case where the difference between two consecutive exponents corresponding to the same node is always large.

4.2.1 Small intervals

The following scenario is motivated by Jordan's Lemma (Lemma 2.2.16): for each distinct node a, we allow several affine powers with this node but all the corresponding exponents must lie in a "small" interval. In the first version of this work [31], we studied the more constrained case where we have a "uniform" bound on the size of each interval, but in the following we will rather impose an upper bound on the sum of the degrees, which is more similar to Jordan's Lemma hypothesis. More precisely, we consider polynomials f having an expression in Model 2 of the following form:

Model 8.

$$f = \sum_{i=1}^{r} Q_i(x)\,(x - a_i)^{e_i} = \sum_{i=1}^{r} \sum_{j=1}^{s_i} \alpha_{i,j} (x - a_i)^{e_i + \varepsilon_{i,j}} \tag{4.1}$$

with distinct $a_i \in \mathbb{F}$, $\alpha_{i,j} \in \mathbb{F}$, $Q_i \in \mathbb{F}[x]$, $e_i, \varepsilon_{i,j} \in \mathbb{N}$.
We will usually set $\delta_i \overset{def}{=} \deg(Q_i) = \max_j\{\varepsilon_{i,j}\}$, *and* $\Delta \overset{def}{=} \sum_{i=1}^{r} \delta_i$, *and we will say that Equation* (4.1) *is a* Δ-decomposition *of* f. *We will also set* $s \overset{def}{=} \sum_{i=1}^{r} s_i \leq r + \Delta$ *so that if* f *admits a* Δ*-decomposition, then we have* $\mathit{AffPow}_{\mathbb{F}}(f) \leq s$.

Notice that the results of Section 4.1 correspond to the case where $\Delta = 0$ and indeed most of them can be re-obtained as particular cases of the results of this section. The definition of Δ-decompositions is also motivated by an analysis of the proof of Proposition 3.1.3 in the case of repeated nodes: two affine powers $(x-a)^e$ and $(x-a)^d$ with same node can be gathered to decrease the number of terms, with an extra-cost of $|d-e|$ in the left-hand side of the inequality. More precisely, we prove the following refinement of Proposition 3.1.3 that is more suitable for Δ-decompositions.

Proposition 4.2.1. *Let $F = \bigcup_{i=1}^{r} \{(x-a_i)^{e_i+\varepsilon_{i,j}} : j \in [\![1, s_i]\!]\}$ with $a_i \in \mathbb{F}$, and $e_i, \varepsilon_{i,j} \in \mathbb{N}$. Define $\delta_i \stackrel{def}{=} \max\{\varepsilon_{i,j} : j \in [\![1, s_i]\!]\}$ and $\Delta \stackrel{def}{=} \sum_{i=1}^{r} \delta_i$. Then for any choice of parameters (t,k,l) such that*

$$r(l+k+1) + \Delta < (k+1)(l+1) + \binom{t}{2} = |D(t,k,l)| \tag{4.2}$$

there exists a SDE$_{\leq}(t,k,l)$ satisfied simultaneously by all the polynomials $f_{i,j}(x) = (x-a_i)^{e_i+\varepsilon_{i,j}}$ for $i = 1, \ldots, s, j = 1, \ldots, s_i$.

Proof. Again, the existence of this common SDE is equivalent to the existence of a common nonzero solution for a system of linear equations with $(k+1)(l+1) + \binom{t}{2}$ unknowns, so we need to show that the rank of the corresponding matrix is smaller. For all $u \in [\![1, r]\!]$, we will show that the subspace V_u has dimension less than $l + k + 1 + \delta_u$, where V_u is the linear space spanned by the polynomials $x^j f_{u,v}^{(i)}(x)$, with $(i,j) \in D(t,k,l), v \in [\![1, s_u]\!]$. Notice again that V_u is included in the subspace spanned by the polynomials

$$\{(x-a_u)^{e_u+j};\ -k \leq j \leq l+\delta_u,\ e_u + j \geq 0\}.$$

Hence, we have that $\dim V_r \leq l + k + 1 + \delta_u$ and, since the rank is subadditive, the whole system has rank $\leq r(l+k+1) + \sum_{i=1}^{r} \delta_i$, proving the result. □

Remark 4.2.2. *If we have $s_i = 1$ for all $i \in [\![1, r]\!]$, then $\Delta = 0$ and we obtain the same result as Proposition 3.1.3, proving that Proposition 4.2.1 is indeed a refinement. Notice also that the following choice of parameters satisfies Equation* (4.2)*: $t = k+1$, $k = 2r-1$ and $l = \Delta/r$; which is again a generalization of the choice of parameters of Section 4.1 ($k = 2s-1, l = 0$).*

In order to prove the correctness of our algorithms, we will need to generalize the ingredients we used in Section 4.1 to Model 8. We begin by proving that Propositions 3.2.4 and 3.1.5 can be extended for Δ-decompositions.

Proposition 4.2.3. *Let $f \in \mathbb{F}[x]$ be a polynomial that admits the following Δ-decomposition:*

$$f = \sum_{i=1}^{r} Q_i(x)\,(x-a_i)^{e_i}.$$

Whenever f satisfies a SDE(t,k,l), then for all $e_i \geq k + (k+l)(r-1) + \binom{r}{2} + (\Delta - \delta_i)$, we have that $Q_i(x)\,(x-a_i)^{e_i}$ satisfies the same SDE.

Proof. We suppose that $Q_1(x)(x-a_1)^{e_1}$ does not satisfy this equation, and we are going to prove that it implies that e_1 is small. For every $j \in [\![1, r]\!]$, we denote by g_j and R_j the polynomials such that

$$g_j = \sum_{i=0}^{k} P_i(x) \left(Q_j(x)(x-a_j)^{e_j}\right)^{(i)} = R_j(x)(x-a_j)^{d_j}$$

where $d_j := \max\{0, e_j - k\}$ for all j, and with $\deg R_j \leq k + l + \delta_j$; so that $\sum_{j=1}^{r} g_j = 0$ with $g_1 \neq 0$ by hypothesis. We consider a linearly independent subfamily of $g_2, \ldots, g_r$, namely $\{g_j \,|\, j \in J\}$ with $J = \{j_1, \ldots, j_p\} \subseteq \{2, \ldots, r\}$. There exists $(\alpha_i) \in \mathbb{F}^p$ such that $g_1 = \sum_{i=1}^{p} \alpha_j g_{j_i}$ and we can assume without loss of generality that $\alpha_1 \neq 0$. By properties of the determinant, we have

$$0 \neq \mathrm{Wr}(\alpha_1 g_{j_1}, g_{j_2}, \ldots, g_{j_p}) = \mathrm{Wr}(g_1, g_{j_2}, \ldots, g_{j_p}).$$

Following Proposition 2.2.4, we factor the Wronskians to obtain

$$\begin{cases} \mathrm{Wr}(\alpha_1 g_{j_1}, g_{j_2}, \ldots, g_{j_p}) & = \prod_{d_{j_i} \geq p-1} (x-a_{j_i})^{d_{j_i}-(p-1)} \cdot W_1 \\ \mathrm{Wr}(g_1, \quad g_{j_2}, \ldots, g_{j_p}) & = (x-a_1)^{d_1-(p-1)} \cdot W_2 \end{cases}$$

with $\deg(W_1) \leq \sum_{i=1}^{p} [\deg(R_i) + p - 1] - \binom{p}{2}$. Since a_1 is distinct from $a_{j_1}, \ldots, a_{j_p}$, then $(x-a_1)^{d_1-(p-1)}$ must divide W_1 and therefore we have

$$e_1 - k - (p-1) \leq \deg(W_1) \leq (k+l)p + \binom{p}{2} + (\Delta - \delta_1).$$

Since $p \leq r-1$, we get that $e_1 \leq k - 1 + (k+l)(r-1) + \binom{r}{2} + (\Delta - \delta_1)$, proving the result. □

Proposition 4.2.4. *Consider the following differential equation with polynomial coefficients:*

$$\sum_{i=0}^{k} P_i(x) f^{(i)}(x) = 0.$$

Assume that $Q(x)(x-a)^e$ satisfies this equation, with $e \geq k$. Then we have $P_k(a) = 0$.

Proof. We take $d \geq e$ and $R \in \mathbb{F}[x]$ such that $(x-a)^d R(x) = (x-a)^e Q(x)$ and $R(a) \neq 0$. Since $(x-a)^d R(x)$ is a solution of the SDE, we have that:

$$\sum_{i=0}^{k} P_i(x) \left((x-a)^d R(x)\right)^{(i)} = 0,$$

we deduce that there exists $q \in \mathbb{F}[x]$ such that $P_k(x)(x-a)^{d-k} h(x) = (x-a)^{d-k+1} q(x)$, from where we deduce that $P_k(a) = 0$. □

From Proposition 4.2.3 we shall now derive Corollary 4.2.5 and Theorem 4.2.6. Under reasonable hypotheses, a Δ-decomposition of f is also the optimal decomposition of f in Model 2 and we have $\mathrm{AffPow}_{\mathbb{F}}(f) = s \stackrel{\text{def}}{=} \sum_{i=1}^{r} s_i$. Since our algorithms only work in the regime of uniqueness, the following results will provide sufficient conditions to ensure this is the case.

Corollary 4.2.5. *Let $f \in \mathbb{F}[x]$ be a polynomial with the following Δ-decomposition:*

$$f = \sum_{i=1}^{r} Q_i(x)\,(x-a_i)^{e_i},$$

with $e_i \geq \frac{5}{2}r^2 + 2\Delta r$ for all i. Then,

a) the set of polynomials $\{Q_i(x)\,(x-a_i)^{e_i} \mid 1 \leq i \leq r\}$ is linearly independent,

b) $\mathrm{AffPow}_{\mathbb{F}}(f) = \sum_{i=1}^{r} s_i$ (see 8 for the definition of the s_i's) and the optimal representation of f is unique,

c) f satisfies a SDE$(2r-1, \Delta/r)$,

d) if f satisfies the SDE$(k,\, 2\Delta/(k+1))$

$$\sum_{i=0}^{k} P_i(x) f^{(i)}(x) = 0$$

and $k \leq 2r-1$; then $Q_i(x)(x-a_i)^{e_i}$ also satisfies it and $P_k(a_i) = 0$ for all $i \in [\![1, r]\!]$, and

e) f does not satisfy any SDE$(k,\, 2\Delta/(k+1))$ with $k < r$.

Proof. Notice that (b) implies (a). To prove (b), we observe that a Δ-decomposition of f gives an expression in Model 2 with $s = \sum_{i=1}^{r} s_i$ terms. Now assume that f can also be expressed as $f = \sum_{i=1}^{u} \beta_i (x-b_i)^{d_i}$ with $\beta_i \in \mathbb{F}$ and $u \leq s \leq r + \Delta$. By Theorem 2.2.5 we get that either both expressions are the same, or

$$2(r+\Delta) \geq u + s \geq \sqrt{2(\min\{e_1, \ldots, e_r\} + 1)} > \sqrt{5(r+\Delta)^2},$$

which is not possible. Thus $\mathrm{AffPow}_{\mathbb{F}}(f) = s$ and the optimal representation of f is unique.
From Proposition 4.2.1 we get (c) (see Remark 4.2.2). Assume that f satisfies a SDE$(k,\, 2\Delta/(k+1))$ with $k \leq 2r-1$. For all $i \in [\![1, r]\!]$ we have that

$$\begin{aligned} e_i &\geq \tfrac{5}{2}r^2 + 2\Delta r > \tfrac{5}{2}r^2 + (2\Delta - \tfrac{3}{2})r - \Delta \\ &= 2r - 1 + (r-1)(2r - 1 + 2\Delta) + \tbinom{r}{2} + \Delta \\ &\geq k + (r-1)\left(k + \tfrac{2\Delta}{k+1}\right) + \tbinom{r}{2} + \Delta. \end{aligned} \tag{4.3}$$

Hence, Proposition 4.2.3 yields that $Q_i(x)(x-a_i)^{e_i}$ is also a solution of this equation for all i, proving (d) by Proposition 4.2.4. Finally, notice that f cannot satisfy a SDE$(k,\, 2\Delta/(k+1))$ with $k < r$; otherwise by (a) and (d), the vector space of solutions to this equation has dimension $\geq r$, which contradicts Lemma 3.1.4. $\square$

Theorem 4.2.6. *Let $f \in \mathbb{F}[x]$ be a polynomial with the following Δ-decomposition:*

$$f = \sum_{i=1}^{r} Q_i(x)\,(x-a_i)^{e_i},$$

with $e_i \geq \frac{5}{2}r^2 + 2\Delta r$ for all i. Then AffPow$_{\mathbb{F}}(f) = \sum_{i=1}^{r} s_i$. Moreover, there is a polynomial time algorithm `Build`(f,Δ) *that receives $f = \sum_{i=0}^{d} f_i x^i \in \mathbb{F}[x]$ and Δ as input and computes the r-tuples of nodes $N(f) = (a_1,\ldots,a_r)$, the values $s_1,\ldots,s_r$ and the tuple of coefficients $C(f) = (\gamma_{i,j} : 1 \leq i \leq r,\ 1 \leq j \leq s_i)$, and exponents $E(f) = (e_i + \epsilon_{i,j} : 1 \leq i \leq r,\ 1 \leq j \leq s_i)$. The algorithm* `Build`$(f,\Delta)$ *works as follows:*

Step 1. *Take u the minimum value such that f satisfies a SDE$(u, 2\Delta/(u+1))$. Compute explicitly one of these SDE, i.e., compute $P_0,\ldots,P_u \in \mathbb{F}[x]$ such that $\sum_{i=0}^{u} P_i(x) f^{(i)}(x) = 0$ and $\deg(P_i) \leq i + 2\Delta/(u+1)$.*

Step 2. *Compute $\mathcal{R} = \{c_1,\ldots,c_p\} \subseteq \mathbb{F}$ the set of roots of P_u. For each $i \in \{1,\ldots,p\}$, consider the $\mathbb{F}$-vector space V_i spanned by the solutions of the SDE of the form $R(x)(x-c_i)^e$, with $\frac{5}{8}(u+1)^2 + \Delta(u+1) < e < d + \frac{(u+\Delta)^2}{2}$ and $R(x)$ a polynomial of degree $\leq \Delta$.*

We take $B_i = \{g_{i,1},\ldots,g_{i,l_i}\}$ a base of V_i, where $g_{i,j} = R(x)(x-c_i)^e$. with $\frac{5}{8}(u+1)^2 + \Delta(u+1) < e < d + \frac{(u+\Delta)^2}{2}$ and $R(x)$ a polynomial of degree $\leq \Delta$. We set $B := \cup_{i=1}^{p} B_i$.

Step 3. *Express f as a linear combination of the elements of B, namely, $f = \sum_{i=1}^{p}\sum_{j=1}^{l_i} \lambda_{i,j}\, g_{i,j}$ with $\lambda_{i,j} \in \mathbb{F}$.*

Step 4. *Denote $f_i = \sum_{j=1}^{l_i} \lambda_{i,j}\, g_{i,j}$, for all $i \in \{1,\ldots,p\}$. Write f_i in the shift c_i, i.e., $f_i = \sum_{j=1}^{r_i} \beta_{i,j}(x-c_i)^{\mu_{i,j}}$ with $\beta_{i,j} \in \mathbb{F}\setminus\{0\}$.*

Step 5. *Output $N(f) = (c_1,\ldots,c_p)$, $r_1,\ldots,r_p \in \mathbb{N}$, $C(f) = (\beta_{i,j} \,|\, 1 \leq i \leq p, 1 \leq j \leq r_i)$ and $E(f) = (\mu_{i,j} \,|\, 1 \leq i \leq p, 1 \leq j \leq r_i)$.*

Proof. We observe that f satisfies the hypotheses of Corollary 4.2.5; then, by Corollary 4.2.5.(b), we have that AffPow$_{\mathbb{F}}(f) = \sum_{i=1}^{r} s_i$ and that there is a unique optimal expression of f in the AffPow model.
Let us prove the correctness of the algorithm `Build`(f,Δ). By Corollary 4.2.5.(c) and (e), the value u computed in **Step 1** satisfies that $r \leq u \leq 2r-1$. By Corollary 4.2.5.(d), for all $i \in [\![1,r]\!]$ we have that

- $a_i \in \mathcal{R}$, and
- $Q_i(x)\,(x-a_i)^{e_i}$ is a solution of the SDE computed in **Step 1**

Moreover, the input polynomial f can be expressed as a linear combination of the elements of B, because:

- f can be written as a combination of $Q_i(x)\,(x-a_i)^{e_i}$.
- Since $\mathrm{AffPow}_{\mathbb{F}}(f) = \sum_{i=1}^{r} s_i \leq r + \Delta \leq u + \Delta$, using Corollary 2.2.10 we have $\max\{e_i\} < d + \frac{(u+\Delta)^2}{2}$. On the other hand, we have $e_i \geq \frac{5}{2}r^2 + 2\Delta r \geq \frac{5}{2}(\frac{u+1}{2})^2 + 2\Delta\frac{u+1}{2}$. This implies that $Q_i(x)\,(x-a_i)^{e_i}$ belongs to V_i and thus can be written as a linear combination of the elements of B_i.

So, let us assume (we will prove it later) that all the elements of B are linearly independent. Then, in **Step 3** there is a unique way of writing of f as a linear combination of the elements of B. Finally, it suffices to write $f_i = R_i(x)(x-c_i)^{d_i}$ and consider the Taylor expansion of $R_i(x)$ with respect to c_i for every $i \in [\![1, p]\!]$ as in **Step 4** to get the desired sets of nodes, coefficients and exponents.
To prove the correctness of the algorithm, it only remains to prove that the elements of B are linearly independent. To prove this we will follow a similar argument to that of Theorem 2.2.5. Assume that the elements of B are not linearly independent, and take $W = \{P_i(x)(x-b_i)^{d_i} \mid 1 \leq i \leq w\} \subset B$ a minimal $\mathbb{F}$-linearly dependent set: there exist $\lambda_2, \ldots, \lambda_w \in \mathbb{F}^*$ such that

$$f_1(x) = \sum_{i=2}^{w} \lambda_i f_i(x)$$

with $f_i(x) := P_i(x)(x-b_i)^{d_i}$. By Lemma 3.1.4, the size of this set is $w \leq u+1 \leq 2r$. Moreover, since the set B_i is linearly independent for all i, we can assume without loss of generality that $b_1 \neq b_2$. By properties of the determinant:

$$0 \neq \mathrm{Wr}(f_1, f_3, \ldots, f_w) = \mathrm{Wr}(\lambda_2 f_2, f_3, \ldots, f_w)$$

We define $Z = \{i : 3 \leq i \leq w, d_i \geq w\}$ and, following Proposition 2.2.4, we factorise the Wronskians:

$$\begin{cases} \mathrm{Wr}(f_1, \quad f_3, \ldots, f_w) & = (x-b_1)^{d_1-(w-1)} \prod_{i\in Z}(x-b_i)^{\,d_i-(w-1)} \cdot W_1 \\ \mathrm{Wr}(\lambda_2 f_2, f_3, \ldots, f_w) & = (x-b_2)^{d_2-(w-1)} \prod_{i\in Z}(x-b_i)^{\,d_i-(w-1)} \cdot W_2 \end{cases}$$

where $W_1, W_2 \in \mathbb{F}[x]$ are the remaining determinants whose degrees are upper bounded by $(w-1)\Delta + \binom{w-1}{2}$ according to Proposition 2.2.4. Since $b_1 \neq b_2$, then $(x-b_1)^{d_1-(w-1)}$ must divide W_2 and therefore we should have

$$d_1 - (w-1) \leq \frac{(w-1)(w-2)}{2} + (w-1)\Delta$$

By hypothesis, we have $d_1 \geq \frac{5}{8}(u+1)^2 + \Delta(u+1)$ and since $w \leq u+1$, we therefore have

$$\frac{5}{8}(u+1)^2 + \Delta(u+1) \leq \frac{u(u-1)}{2} + \Delta u + u,$$

a contradiction. □

Remark 4.2.7. *The algorithm* `Build`(f, Δ) *described above can be slightly modified to not receive* Δ *as input as long as* f *satisfies the hypotheses of Theorem 4.2.6 for some* $r, \Delta \in \mathbb{N}$*. That is, we only need to assume that there exists a* Δ*-decomposition of* f *with all the exponents greater than* $\frac{5}{2}r^2 + 2\Delta r$*. Indeed, it suffices to start with* $\Delta = 0$ *and execute* `Build`(f, Δ) *with increasing values of* Δ *until the reconstruction of* f *succeeds. The correctness of this algorithm is justified by Corollary 4.2.5.(b). In fact, once we find* Δ *such that the reconstruction is possible, we obtain the optimal expression of* f *in the Affine Power model.*

4.2.2 Big gaps

This subsection deals with polynomials f such that whenever the terms $(x-a)^e$ and $(x-a)^d$ appear in the optimal expression of f in the Affine Power model, then the difference between d and e is "large". Similarly to Section 4.2.1, we begin with the proof of an extension of Proposition 3.2.4 for this specific scenario. The desired algorithm then follows as a consequence.

Proposition 4.2.8. *Let* $f \in \mathbb{F}[x]$ *be written as*

$$f = (x-a)^m g(x) + \sum_{i=1}^{s} \alpha_i (x-a)^{e_i} + \sum_{i=1}^{p} \beta_i (x-a_i)^{d_i},$$

with $g \in \mathbb{F}[x], a, a_i, \alpha_i, \beta_i \in \mathbb{F}$*,* $m, e_i, d_i \in \mathbb{N}$ *and* $a_i \neq a$ *for all* i*. We set* $e := \max\{e_1, \ldots, e_s\}$ *if* $s \geq 1$ *or* $e := -1$ *if* $s = 0$*. Whenever* f *satisfies a SDE*(k, l) *with* $m - e > (k+l)(p+1) + \binom{p+1}{2}$*, then* $(x-a)^m g$ *satisfies the same SDE.*

Proof. Assume that f satisfies a SDE(k, l)

$$\sum_{i=1}^{k} P_i(x) f^{(i)}(x) = 0$$

By contradiction, we assume that $(x-a)^m g(x)$ does not satisfy this equation. Thus, there exists $T(x) \in \mathbb{F}[x]$ nonzero such that $\sum_{i=0}^{k} P_i(x) \left((x-a)^m g\right)^{(i)} = T(x)(x-a)^{m-k}$. For every $j \in [\![1, s]\!]$ and every $j \in [\![1, p]\!]$, we denote by h_j and g_j the polynomials such that

$$h_j = \sum_{i=0}^{k} P_i(x) \left((x-a)^{e_j}\right)^{(i)} \quad \text{and} \quad g_j = \sum_{i=0}^{k} P_i(x) \left((x-a_j)^{d_j}\right)^{(i)}.$$

We observe that $\deg(h_j) \leq e_j + l \leq e + l$ and $\deg(g_j) \leq d_j + l$. Since f satisfies the already mentioned SDE, we get that

$$T(x)(x-a)^{m-k} = \sum_{i=1}^{s} \alpha_i h_i + \sum_{i=1}^{p} \beta_i g_i.$$

If we differentiate $(e+l+1)$ times on both sides of the previous equation, we obtain an equality of the following form

$$U(x)(x-a)^{m-k-e-l-1} = \sum_{i=1}^{p} \beta_i g_i^{(e+l+1)} = \sum_{i=1}^{p} U_i(x)(x-a_i)^{r_i}$$

with $r_i := \max\{0, d_i - k - e - l - 1\}$ and $\deg(U_i(x)) \leq k+l$. If we take a linearly independent family $\{g_i^{(e+l+1)} : i \in I\} \subseteq \{g_i^{(e+l+1)} : i \in [\![1,p]\!]\}$, then factorize the corresponding Wronskians following Proposition 2.2.4 as before, we obtain the following inequality:

$$m - k - e - l - 1 - (p-1) \leq (k+l)p + (p-1)p - \binom{p}{2},$$

which yields that

$$m - e \leq (k+l)(p+1) + \binom{p+1}{2},$$

a contradiction. □

The following result is a generalization of Proposition 3.2.4 where we allow repeated nodes provided their corresponding exponents are far enough.

Corollary 4.2.9. *Let $f \in \mathbb{F}[x]$ be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x-a)^{e_i} + \sum_{i=1}^{p} \beta_i (x-a_i)^{d_i},$$

with $a, a_i, \alpha_i, \beta_i \in \mathbb{F}$, $m, e_i, d_i \in \mathbb{N}$, $a_i \neq a$ for all i and $e_s > \cdots > e_1 > e_0 := -1$. Assume that f satisfies a SDE(k,l) and that $e_{i+1} - e_i > (k+l)(p+1) + \binom{p+1}{2}$ for all i, then $(x-a)^{e_i}$ satisfies the same SDE for all $i \in \{1, \ldots, s\}$.

Proof. Assume that there exists an e_i such that $(x-a)^{e_i}$ does not satisfy the SDE(k,l) and we take e the maximum of such e_i. Then, we can write $f(x) = g(x)(x-a)^e + \sum_{e_i<e} \alpha_i (x-a)^{e_i} + \sum_{i=1}^{p} \beta_i (x-a_i)^{d_i}$. By means of Proposition 4.2.8 we have that $g(x)(x-a)^e$ is a solution of the same SDE. Moreover, for all $e_i > e$, then $(x-a)^{e_i}$ is also a solution of the SDE. But this is not possible since the set of solutions is a vector space, and, hence, $(x-a)^e$ would also be a solution to the same SDE. □

The proof of the following Corollary is similar to that of Corollary 4.1.1 but makes use of Corollary 4.2.9 instead of Proposition 3.2.4.

Corollary 4.2.10. *Let $f \in \mathbb{F}[x]$ be a polynomial that can be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x-a_i)^{e_i}$$

with $a_i, \alpha_i \in \mathbb{F}$, $e_i > 5s^2/2$ and, whenever $a_i = a_j$ for some $1 \leq i < j \leq s$, then $|e_i - e_j| > 5s^2/2$.

a) $\{(x-a_i)^{e_i} \mid 1 \leq i \leq s\}$ *are linearly independent,*

b) If $f = \sum_{i=1}^{t} \beta_i (x-b_i)^{d_i}$ *with* $t \leq s$*, then* $t = s$ *and we have the equality* $\{(\alpha_i, a_i, e_i) \mid 1 \leq i \leq s\} = \{(\beta_i, b_i, d_i) \mid 1 \leq i \leq s\}$*; in particular,* $\mathit{AffPow}_{\mathbb{F}}(f) = s$*,*

c) f *satisfies a SDE*$(2s-1, 0)$*,*

d) if f *satisfies a SDE*$(k, 0)$ *with* $k \leq 2s-1$*, then* $(x-a_i)^{e_i}$ *also satisfies it for all* $i \in \{1, \ldots, s\}$*, and*

e) f *does not satisfy any SDE*$(k, 0)$ *with* $k < s$*.*

From this corollary we get the following result whose proof is similar to that of Theorem 4.1.2.

Theorem 4.2.11 (Big gaps)**.** *Let* $f \in \mathbb{F}[x]$ *be a polynomial that can be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x-a_i)^{e_i}$$

with $a_i, \alpha_i \in \mathbb{F}$*,* $e_i > 5s^2/2$ *and whenever* $a_i = a_j$ *for some* $1 \leq i < j \leq s$*, then* $|e_i - e_j| > 5s^2/2$*. Then,* $\mathit{AffPow}_{\mathbb{F}}(f) = s$*. Moreover, there is a polynomial time algorithm* $\texttt{Build}(f)$ *that receives* $f = \sum_{i=0}^{d} f_i x^i \in \mathbb{F}[x]$ *as input and computes the* s*-tuples of nodes* $N(f) = (a_1, \ldots, a_s)$*, coefficients* $C(f) = (\alpha_1, \ldots, \alpha_s)$ *and exponents* $E(f) = (e_1, \ldots, e_s)$*. The algorithm* $\texttt{Build}(f)$ *works as follows:*

Step 1. *Take* r *the minimum value such that* f *satisfies a SDE*$(r, 0)$ *and compute explicitly one of these SDE.*

Step 2. *Compute* $B = \{(x-b_i)^{d_i} \mid 1 \leq i \leq t\}$*, the set of all the solutions of the SDE of the form* $(x-b)^d$ *with* $(r+1)^2/2 \leq e \leq \deg(f) + (r^2/2)$*.*

Step 3. *Determine* $\alpha_1, \ldots, \alpha_r$ *such that* $f = \sum_{i=1}^{r} \alpha_i (x-b_i)^{d_i}$

Step 4. *Output the sets* $C(f) = (\alpha_1, \ldots, \alpha_r)$*,* $N(f) = (b_1, \ldots, b_r)$ *and* $E(f) = (d_1, \ldots, d_r)$*.*

5

Multivariate reconstruction algorithms

Let $\mathbb{F}[X] = \mathbb{F}[x_1, \ldots, x_n]$ be a ring of polynomials in n variables over a characteristic 0 field. This chapter concerns Model 1 (the multivariate analogue of Model 2), i.e., we study expressions of a polynomial $f \in \mathbb{F}[X]$ as

$$f = \sum_{i=1}^{s} \alpha_i \ell_i^{e_i}, \tag{5.1}$$

where $e_i \in \mathbb{N}$, $\alpha_i \in \mathbb{F}$ and ℓ_i is a (non constant) affine form for all i. Whenever $\mathbb{F}$ is an algebraically closed field, we may assume without loss of generality that all the α_i's equal 1. For the sake of conciseness, we assume this is the case. However, one can restate all the results in this chapter for a non algebraically closed field by just adding the α_i's.

The main goal of this chapter is to design algorithms that reconstruct the optimal representation of polynomials in this model, i.e., algorithms that receive as input $f \in \mathbb{F}[X]$ and compute the exact value $s = \text{AffPow}_{\mathbb{F}}(f)$ and a set of triplets of coefficients, affine forms and exponents $\{(\alpha_i, \ell_i, e_i) \,|\, 1 \leq i \leq s\} \subseteq \mathbb{F} \times \mathbb{F}[X] \times \mathbb{N}$ such that $f = \sum_{i=1}^{s} \alpha_i \ell_i^{e_i}$. In the following, we devise algorithms for finding optimal representations of a polynomial $f \in \mathbb{F}[X]$ in Model 1, provided the value of $\text{AffPow}(f)$ is small compared to the number of variables or to the degree of f.

Let us denote by $\text{EssVar}(f)$ the number of *essential variables* of f. This is roughly speaking the number of variables on which f "truly depends" up to a linear change of variables [18, 43]. A first easy remark is that the value $\text{AffPow}(f)$ is at least equal to $\text{EssVar}(f)$. In Section 5.2 we investigate when this is an equality and provide an algorithm that decides whether $\text{AffPow}(f) = \text{EssVar}(f)$ and, if so, provides an optimal expression in the model.

In Section 5.3, we generalize the previous results to characterize by means of an algorithm when a polynomial $f \in \mathbb{F}[X]$ can be written as a sum of univariates after an affine change of coordinates. It is plausible that when this is a case, an optimal expression of f can be built by putting together the optimal expressions of all the univariate polynomials involved. We believe this is true and we give a proof for $n = 2$. The general case ($n \geq 3$) is left as an open problem.

In Section 5.4, we focus on the reconstruction problem when $\text{AffPow}(f) \leq \binom{n+1}{2}$. In the main result of this section, we provide a randomized algorithm that works when, in the optimal decomposition, all the e_i's are ≥ 5 and the coefficients of the ℓ_i's are taken uniformly at random from a finite set. In particular, this provides a new algorithm for computing Waring decompositions of "generic polynomials with $\text{Waring}(f) \leq \binom{n+1}{2}$". For comparison, note that the algorithm from [43] can only find Waring decompositions up to size n, and that the Waring decomposition algorithm from [44] is only interesting when d is relatively large compared to s (see Theorem 5 and Remark 6 in that paper). Our main tool in this section is a "4th order Hessian" inspired from the ordinary Hessian determinant used in [43].

Finally, in Section 5.5 we propose an algorithm that performs random univariate projections, calls our univariate algorithms for sums of affine powers from Chapter 4 and reconstructs f from this univariate information.

5.1 Preliminaries

5.1.1 Algorithmic preliminaries

In the rest of this chapter, we will design algorithms that work in the "blackbox" setting: they have access to the input polynomial only through an oracle so that for any point $a \in \mathbb{F}^n$, we can obtain $f(a)$ in a single step by querying this oracle. This representation of an input polynomial is in some sense the weakest representation for which one can hope to have efficient algorithms and it subsumes all other representations such as arithmetic circuits. This very general model is standard for the study of many problems about multivariate polynomials such as, e.g., factorization [40], sparse interpolation [6, 59], sparsest shift [60] or Waring decomposition [41]. In this section, we describe some useful blackbox subroutines that our algorithms will use.

Change of basis

In the following, given a polynomial $f(X) \in \mathbb{F}[X]$, we will often want to consider $h = f(A \cdot X + b)$ with $A \in \mathcal{M}_n(\mathbb{F})$ and $b \in \mathbb{F}^n$. It is straightforward to obtain a blackbox access to h given a blackbox access to f.

Solving linear systems

Using evaluation at random points, one could prove the following proposition using similar ideas to [44, Lemma 14]

Proposition 5.1.1. *Let $f, h_1, \ldots, h_p \in \mathbb{F}[X]$ be polynomials. Given blackbox accesses to $f, h_1, \ldots, h_p$, we can test if f can be written as a linear combination of the h_i's, and output one such combination if it exists, in randomized polynomial time.*

Polynomial Identity Testing

Given a blackbox access to a polynomial $f \in \mathbb{F}[X]$ of degree d, the Zippel-Schwartz lemma [73, 66] ensures that evaluating f at random points yields a randomized polynomial time algorithm that tests whether f is equal to the zero polynomial.

Obtaining the derivatives

Proposition 5.1.2. *[44, Proposition 18] Let $f(X) \in \mathbb{F}[X]$ be an n-variate polynomial of degree d. Given blackbox access to f, in time poly(dn), we obtain blackbox access to any derivative $\frac{\partial f}{\partial x}$ of f.*

Obtaining the homogeneous components

For a polynomial $f(X) \in \mathbb{F}[X]$ we will denote by $[f]_k$ its homogeneous component of degree k. We will also sometimes use the notation $[f]_{\geq k}$ defined as $[f]_{\geq k} := \sum_{i \geq k} [f]_i$.

Proposition 5.1.3. *[44, Proposition 19] Let $f(X) = \sum_{i=0}^{d} [f]_i(X)$ be a polynomial of degree d. Given blackbox access to f and a point $a \in \mathbb{F}^n$, we can compute $[f]_i(a)$ for each $i \in [\![0, d]\!]$ in polynomial time.*

Factorization

To factorize a polynomial, we will use the randomized polynomial time algorithm described in [40]: given a blackbox access to $f(X) = \prod_{i=1}^{r} h_i(X)^{e_i}$, it outputs the e_i's and yields a blackbox access to the h_i's. This algorithm needs an effective polynomial factorization algorithm for $\mathbb{F}[x]$ in order to work, so we will assume to have such an algorithm throughout this chapter. In the following, given a polynomial f, we will often need to reconstruct the coefficients of a factor h of degree 1 of f. This can be done using an additional randomized step in $\text{poly}(n)$ time: we evaluate h in $n+1$ random points, interpolate the coefficients and then evaluate in one more point to ensure with high probability that the other coefficients (of higher degree monomials) are zero.

5.1.2 Essential variables

We first clarify what we mean by "small in term of n" because a polynomial f with n variables $x_1, \ldots, x_n$ can also be seen as a polynomial with $n+1$ variables even though it does not depend on the last one. We say that a polynomial $f(X) \in \mathbb{F}[X]$ *depends on a variable* if it appears in at least one of the monomials of $f(X)$. The *number of essential variables* of a polynomial $f(X)$, denoted by $\text{EssVar}(f)$, is the least integer $t \in [\![0, n]\!]$ such that there exists an invertible linear transformation $A \in \text{GL}_n(\mathbb{F})$ such that $f(A \cdot X)$ depends on t variables. The number of essential variables of a polynomial is given by the following result due to Carlini [18, Proposition 1].

Proposition 5.1.4. *The number of essential variables of $f(X) \in \mathbb{F}[X]$ equals the dimension of the $\mathbb{F}$-vector space spanned by the first partial derivatives of $f(X)$. In other words,*

$$\textit{EssVar}(f) = \dim_{\mathbb{F}} \left\langle \frac{\partial f}{\partial x_i} \,\middle|\, 1 \leq i \leq n \right\rangle.$$

A first easy observation is that if $f(X) = \sum_{i=1}^{s} \ell_i^{e_i}$ with ℓ_i affine forms and $e_i \in \mathbb{N}$; then $\left\langle \frac{\partial f}{\partial x_i} \,|\, 1 \leq i \leq n \right\rangle \subseteq \left\langle \ell_i^{e_i - 1} \,|\, 1 \leq i \leq s \right\rangle$ and, hence, $\text{EssVar}(f) \leq s$. In particular, $\text{EssVar}(f) \leq \text{AffPow}(f)$.

A polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ is said *regular* if it has n essential variables. From now on we always assume that the input polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ is regular. This can be achieved through a preliminary step consisting of eliminating the redundant variables using a randomized polynomial time algorithm (see, e.g., [44, Lemma 17] and [43, Theorem 4.1]).

5.2 From reconstruction to polynomial equivalence

Let us first consider $f \in \mathbb{F}[X]$ a regular polynomial such that $\mathrm{AffPow}(f) = n$, i.e. there exists a decomposition $f(X) = \sum_{i=1}^{n} \ell_i^{e_i}$. We construct the matrix A from the linear parts of the ℓ_i's and the vector b from the constant terms. Since f is regular, we have that $A \in \mathrm{GL}_n(\mathbb{F})$. This implies that $f(A^{-1}X - A^{-1}b) = \sum_{i=1}^{n} x_i^{e_i}$ and motivates the following definitions.

Definition 5.2.1. *[44] We will say that two n-variate polynomials f and g are equivalent, denoted $f \sim g$, if there exists an invertible linear transformation $A \in \mathrm{GL}_n(\mathbb{F})$ such that $f(X) = g(A \cdot X)$. Moreover, we will say that f and g are affine equivalent, denoted $f \equiv g$ if there exists a vector $b \in \mathbb{F}^n$ such that $f(X+b) \sim g$, or equivalently if $f = g(A \cdot X + c)$ with $A \in \mathrm{GL}_n(\mathbb{F}), c \in \mathbb{F}^n$.*

With these notations, for a regular polynomial f, we have that $\mathrm{AffPow}(f) = n$ if and only if $f \equiv g$ where $g = \sum_{i=1}^{n} x_i^{e_i}$ for some $(e_i) \in \mathbb{N}^n$. This restates the problem of checking whether $\mathrm{AffPow}(f) = n$ into a problem of testing affine equivalence. The affine equivalence problem was already investigated in [44]. One major difference of our situation with respect to [44] is that instead of testing affine equivalence to one target polynomial g, we test affine equivalence to a family of polynomials. Another difference is that its author used [44, Theorem 28] as a preliminary step to reduce the affine equivalence problem to an equivalence problem, which cannot be used here since the polynomials we consider are not homogeneous in general. Yet, the techniques used to solve some special cases of the equivalence problem in [43] have been a source of inspiration to design the algorithms of this chapter.

5.2.1 Algorithm overview

Let us fix some notations: unless stated otherwise, f will always denote the input polynomial and g one target polynomial. Whenever $f \equiv g$, we will usually denote by A and b the matrices such that $f(X) = g(A \cdot X + b)$, with $A \in \mathrm{GL}_n(\mathbb{F})$. Moreover, we will define the associated affine and linear forms: $\ell_i = \sum_{j=1}^{n} A_{i,j} x_j + b_i$ and $[\ell_i] = \ell_i - b_i$. The main tool of the algorithms is the Hessian Matrix, whose entries are the second order derivatives of the polynomial.

Definition 5.2.2. *For a polynomial $f \in \mathbb{F}[X]$, the Hessian Matrix $H_f(X)$ is defined as follows.*

$$H_f(X) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix} \in \mathcal{M}_n(\mathbb{F}[X]).$$

In the following, the most useful property of the Hessian matrix is how affine transformations change the matrix. This lemma is an affine analogue of [43, Lemma 5.1] and can be proved similarly.

Lemma 5.2.3. *Let $g \in \mathbb{F}[X]$ be an n-variate polynomial. Let $A \in \mathcal{M}_n(\mathbb{F})$ be a linear transformation, and let $b \in \mathbb{F}^n$. Let $f(X) = g(A \cdot X + b)$. Then,*

$$H_f(X) = A^T \cdot H_g(A \cdot X + b) \cdot A.$$

In particular,

$$\det(H_f(X)) = \det(A)^2 \det(H_g(A \cdot X + b)).$$

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

$$\frac{\partial f}{\partial x_i} = \sum_{k=1}^{n} a_{ki} \frac{\partial g}{\partial x_k}(A \cdot X + b).$$

Therefore, for second-order derivatives, we can write

$$\begin{aligned}\frac{\partial^2 f}{\partial x_i \cdot \partial x_j} &= \sum_{k=1}^{n} a_{ki} \left(\sum_{\ell=1}^{n} a_{\ell j} \frac{\partial^2 g}{\partial x_k \cdot \partial x_\ell}(A \cdot X + b) \right) \\ &= \sum_{k=1}^{n} \sum_{\ell=1}^{n} a_{ki} \frac{\partial^2 g}{\partial x_k \cdot \partial x_\ell}(A \cdot X + b)\, a_{\ell j}.\end{aligned}$$

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

In particular when $f \equiv g$, the matrix A is invertible and hence the determinant of the Hessian matrix of f can be understood by studying an affine transformation of the determinant of the Hessian matrix of g. For instance, when $g = \sum_{i=1}^{n} x_i^{e_i}$, observe that

$$\frac{\partial^2 g}{\partial x_i \cdot \partial x_j} = \begin{cases} 0 & \text{if } i \neq j, \\ e_i(e_i - 1)x_i^{e_i - 2} & \text{if } i = j \end{cases}$$

The matrix H_g is therefore diagonal and we have

$$\det(H_g(X)) = \prod_{i=1}^{n} e_i(e_i - 1)x_i^{e_i - 2}.$$

In particular, Lemma 5.2.3 implies that

Lemma 5.2.4. *Let f be a regular polynomial such that $f(X) = \sum_{i=1}^{n} \ell_i(X)^{e_i}$ where $\ell_1(X), \ldots, \ell_n(X)$ are affine forms and $e_i \geq 2$. Then we have*

$$\det(H_f(X)) = c \cdot \prod_{i=1}^{n} \ell_i(X)^{e_i - 2}$$

where $c \in \mathbb{F}$ is a nonzero constant.

This result yields a blueprint for an algorithm to find a decomposition of f when $\mathrm{AffPow}(f) = n$: factorize $\det(H_f(X))$ to obtain candidates for the affine forms and associated exponents, then try to express f as a linear combination of these affine powers. However, if $\mathrm{AffPow}_{\mathbb{F}}(f) = n$ and one e_i is ≤ 1 then $\det(H_f(x)) = 0$; and if some of the e_i's are equal to 2 then ℓ_i is not a factor of $\det(H_f(X))$. This makes this idea fail on such scenarios. Therefore, in order to have an algorithm that decides whether $\mathrm{AffPow}_{\mathbb{F}}(f) = \mathrm{EssVar}(f)$, one also needs to handle the case when some of the (e_i)'s are smaller than 3. In the next section, we start tackling this problem by studying the case where f is a quadratic polynomial.

5.2.2 Quadratic polynomials

The goal of this subsection is to describe how to obtain an optimal expression in the Affine Powers model for every polynomial of degree 2. In particular, we are going to generalize the following classical result concerning homogeneous polynomials of degree 2.

Proposition 5.2.5. *Let $\mathbb{F}$ be an algebraically closed field of characteristic different from 2 and let $f, g \in \mathbb{F}[X]$ be homogeneous quadratic polynomials. Then,*

$$f \sim g \iff \mathit{EssVar}(f) = \mathit{EssVar}(g).$$

In particular, from this result we deduce that if f is a quadratic homogeneous polynomial, then $f \sim \sum_{i=1}^{t} x_i^2$ with $t = \mathrm{EssVar}(f)$. Thus, for every quadratic homogeneous polynomial we have $\mathrm{AffPow}(f) = \mathrm{EssVar}(f) = n$. Now we can proceed with the classification of degree 2 polynomials.

Theorem 5.2.6. *Let $\mathbb{F}$ be an algebraically closed field of characteristic different from 2 and let $f \in \mathbb{F}[X]$ be a polynomial of degree at most 2. Then, there exists a unique $r \in [\![0, n]\!]$ such that*

i) $f \equiv \sum_{i=1}^{r} x_i^2$,

ii) $f \equiv \sum_{i=1}^{r} x_i^2 + c$ *with* $c \in \mathbb{F} \setminus \{0\}$, *or*

iii) $f \equiv \sum_{i=1}^{r-1} x_i^2 + x_r$.

Moreover, only one of these three scenarios can hold and $r = \mathit{EssVar}(f)$.

Proof. After a linear change of coordinates we may assume that f has n essential variables. We begin by proposing a greedy algorithm that shows how to write any polynomial f of degree at most 2 as either

(a) $\sum_{i=1}^{s} \ell_i^2$,

(b) $\sum_{i=1}^{s} \ell_i^2 + c$ with $c \in \mathbb{F} \setminus \{0\}$, or

(c) $\sum_{i=1}^{s-1} \ell_i^2 + \ell_s$,

for some $s \leq n$, where $\ell_i = l_i + c_i$ are affine forms whose linear parts $l_1, \ldots, l_s$ are linearly independent, and $c_1, \ldots, c_s \in \mathbb{F}$.
We proceed by induction on the number of variables of f. If f has 0 or 1 variables or f has degree one, it is trivial to write f in one of the desired forms. Assume now that f has $n \geq 2$ variables and that f has degree 2. If there exists a variable x such that the monomial x^2 appears in f, then after multiplying f by a constant if necessary, we write $f = x^2 + xt + g$, where t is a linear form in $n-1$ variables and g is a polynomial of degree ≤ 2 in $n-1$ variables. Thus setting $\ell_1 = x + (t/2)$, we have that $f = \ell_1^2 + g - (t^2/4)$ and proceeding by induction on $g - (t^2/4)$ we are done. If for every variable there is no monomial of the form x^2 in f, then we take two variables x, y such that the monomial xy appears in f. After multiplying f by a constant if necessary, we have that $f = xy + xt_1 + yt_2 + g$, where t_1, t_2 are linear forms in $n-2$ variables and g is a polynomial of degree ≤ 2 in $n-2$ variables. So we set $\ell_1 = (x + y + t_1 + t_2)/2$ and $\ell_2 = (x - y - t_1 + t_2)/2$ and we have that $f = \ell_1^2 - \ell_2^2 + g - t_1 t_2$ and we proceed by induction on $g - t_1 t_2$. We also observe that by construction the linear parts of the affine forms $\ell_1, \ldots, \ell_s$ we obtain are linearly independent. If f has n essential variables, then the integer s in the above construction has to be equal to n.
To prove now that these scenario are disjoint we assume that f is regular and, thus, $s = n$. First, if f can be written as in (a) or (b), then $\det(H_f(X)) \neq 0$ whereas if f can be written as in (c), then $\det(H_f(X)) = 0$. As a consequence, the case (c) is disjoint from the cases (a) and (b). For any polynomial g, we denote by $g^h \in \mathbb{F}[X, z]$ the homogenized polynomial of g with respect to a new variable z. If f can be written as in (a), then $f^h = \sum_{i=1}^{n} (\ell_i^h)^2$, whereas if f can be written as in (b), then $\sum_{i=1}^{n-1} \ell_i^2 + cz^2$. By Proposition 5.2.5, in (a) we have that $\mathrm{EssVar}(f^h) = n$ whereas in (b) we have that $\mathrm{EssVar}(f^h) = n+1$; thus both cannot happen at the same time. □

As a consequence of this result we have the following one, which provides an effective method to compute the exact value of $\mathrm{AffPow}(f)$ for any degree 2 polynomial f. In particular, it implies that a quadratic polynomial always has an optimal decomposition in the AffPow model with exponents at most 2, showing that cancellations don't help for degree 2.

Corollary 5.2.7. *Let $f \in \mathbb{F}[X]$ be a regular polynomial of degree at most 2. Then, $\mathrm{AffPow}(f) = n+1$ if we have $f \equiv \sum_{i=1}^{n} x_i^2 + c$; and $\mathrm{AffPow}(f) = n$ otherwise.*

Proof. If $f \equiv \sum_{i=1}^{n} x_i^2$ or $f \equiv \sum_{i=1}^{n-1} x_i^2 + x_n$, then $\mathrm{AffPow}(f) \leq n$ and equality holds because $\mathrm{AffPow}(f) \geq \mathrm{EssVar}(f) = n$.
By Theorem 5.2.6, it only remains to consider the case when $f \equiv \sum_{i=1}^{n} x_i^2 + c$ with $c \in \mathbb{F}^*$. In this case we clearly have that $\mathrm{AffPow}(f) \leq n+1$, hence to prove equality we just need to prove that $\mathrm{AffPow}(f) \neq n$. Assume for contradiction that $f = \sum_{i=1}^{n} \ell_i^{e_i}$ for some affine forms ℓ_i and some $e_i \in \mathbb{N}$. Since we have neither $f \equiv \sum_{i=1}^{n} x_i^2$ nor $f \equiv \sum_{i=1}^{n-1} x_i^2 + x_n$, there exists some exponent $e_i \geq 3$. By Lemma 5.2.4, we have that $\det(H_f)$ is a non-constant polynomial or zero, a contradiction. □

5.2.3 Linear terms in an optimal expression

We now investigate the case where $f \equiv g$ with $g = \sum_{i=1}^{n} x_i^{e_i}$ and $\min(e_i) = 1$. The algorithm presented in Section 5.2.1 fails on this scenario since $\det(H_g) = 0$ and therefore $\det(H_f) = 0$ by Lemma 5.2.3. Notice first that $e_i = 1$ can only hold for one $i \in [\![1, n]\!]$ since otherwise $\text{EssVar}(f) = \text{EssVar}(g) < n$. Up to renaming the variables, we can therefore write g as $g = \sum_{i=1}^{n-1} x_i^{e_i} + x_n$. Notice that the Hessian of g can be written as

$$H_g(X) = \begin{pmatrix} H_h(X) & 0 \\ 0 & 0 \end{pmatrix} \quad \text{with } h = g - x_n.$$

Let $A \in \mathrm{GL}_n(\mathbb{F})$ and $b \in \mathbb{F}^n$ be such that $f(X) = g(A \cdot X + b)$ and decompose A along its last line $A = \begin{pmatrix} B \\ l \end{pmatrix}$, so that the equality of Lemma 5.2.3 can be rewritten as

$$H_f(X) = (B^T \; l^T) \cdot \begin{pmatrix} H_h(A \cdot X + b) & 0 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} B \\ l \end{pmatrix}$$

If we denote by $[H_f]_{k,k}$ the submatrix of H_f obtained by deleting the k^{th} row and the k^{th} column of H_f, and by $[B]_k$ the square submatrix of B obtained by deleting the k^{th} column, then we have:

$$[H_f(X)]_{k,k} = ([B]_k)^T \cdot H_h(A \cdot X + b) \cdot [B]_k$$

Since $A \in \mathrm{GL}_n(\mathbb{F})$, we have $\operatorname{rank} B = n - 1$ and therefore there exists $k \in [\![1, n]\!]$ such that $[B]_k \in \mathrm{GL}_{n-1}(\mathbb{F})$. In particular, for such a k, we have

$$\det([H_f(X)]_{k,k}) = (\det([B]_k))^2 \cdot \det(H_h(A \cdot X + b))$$

Finally, since $\det(H_h(X)) = \prod_{i=1}^{n-1} e_i(e_i - 1)x_i^{e_i - 2}$, we get the following result.

Lemma 5.2.8. *Let f be a regular polynomial such that $f(X) = \sum_{i=1}^{n-1} \ell_i(X)^{e_i} + \ell_n(X)$ where $\ell_1, \ldots, \ell_n$ are affine forms. Then there exists an integer $k \in [\![1, n]\!]$ and such that*

$$\det([H_f(X)]_{k,k}) = c \cdot \prod_{i=1}^{n-1} \ell_i(X)^{e_i - 2}$$

where $c \in \mathbb{F}$ is a nonzero constant.

5.2.4 Wrapping up : the algorithm

The goal of this subsection is to design a polynomial-time randomized algorithm that receives as input a blackbox access to a polynomial $f \in \mathbb{F}[X]$, and decides whether $\text{AffPow}_{\mathbb{F}}(f) = \text{EssVar}(f) = n$ and, in such a case, provides an optimal expression of f in Model 1. Before presenting the algorithm, we are going to prove some results about uniqueness that follow from previous sections.

Let $s \in \mathbb{N}^*$ and denote by $E_n := \{\underline{e} = (e_1, \ldots, e_n) \in (\mathbb{N}^*)^n \mid e_1 \geq \cdots \geq e_n\}$ the set of non increasing integer sequences of size n. For each sequence $\underline{e} \in E_n$, we consider the associated polynomial $p_{\underline{e}} := \sum_{i=1}^n x_i^{e_i}$. It is easy to check by means of Proposition 5.1.4 that $p_{\underline{e}}$ has n essential variables if and only if $e_{n-1} > 1$. If $\underline{e} \in E_n$ with $e_{n-1} > 1$ and $f \equiv p_{\underline{e}}$, then it is clear that $\mathrm{AffPow}_{\mathbb{F}}(f) = n$.

Proposition 5.2.9. *Let $f \in \mathbb{F}[X]$ be a regular polynomial. If $\mathrm{AffPow}_{\mathbb{F}}(f) = n$, then there exists a unique $\underline{e} = (e_1, \ldots, e_n) \in E_n$ with $e_{n-1} > 1$ such that $f \equiv p_{\underline{e}}$.*

Proof. If $\mathrm{AffPow}_{\mathbb{F}}(f) = n$, then $f = \sum_{i=1}^n \ell_i^{e_i}$ for some $\underline{e} = (e_1, \ldots, e_n) \in E_n$. Since f has n essential variables, then $e_{n-1} > 1$ and $\ell_1, \ldots, \ell_n$ are linearly independent. Hence, $f \equiv p_{\underline{e}}$. To conclude the result it suffices to prove that if $p_{\underline{e}} \equiv p_{\underline{d}}$ for some $\underline{d}, \underline{e} \in E_n$, then $\underline{d} = \underline{e}$. Assume that there exist linearly independent linear forms $\ell_1, \ldots, \ell_n$ such that $p_{\underline{e}} = \sum_{i=1}^n \ell_i^{d_i}$. First we observe that $\det(H_{p_{\underline{e}}}) = 0$ if and only if $e_n = 1$ and, by Lemma 5.2.4, that $\det(H_{\sum_{i=1}^s \ell_i^{d_i}}) = 0$ if and only if $d_n = 1$. Hence, $e_n = 1$ if and only if $d_n = 1$.
Assume first that $e_n > 1$. By Lemma 5.2.4 we have that

$$\det(H_{p_{\underline{e}}}) = c \prod_{i=1}^n x_i^{e_i - 2} = b \prod_{i=1}^n \ell_i^{d_i - 2},$$

with $b, c \in \mathbb{F}$ nonzero constants. This implies in particular $\underline{e} = \underline{d}$ and that whenever $e_i \geq 3$ there exists ℓ_j proportional with x_i such that $d_j = e_i$.
If $e_s = 1$ we obtain the same result using Lemma 5.2.8 instead of Lemma 5.2.4. □

It is easy to obtain examples of degree 2 polynomials such that $\mathrm{AffPow}_{\mathbb{F}}(f) = n$ and the optimal expression in Model 1 is not essentially unique (see Proposition 5.2.5). However, as we have seen in Proposition 5.2.9, when $\mathrm{AffPow}_{\mathbb{F}}(f) = n$ every optimal expression in the Affine Powers model uses the same sequence of exponents. In the following result we are going to prove that the terms corresponding to exponents greater or equal to 3 are also essentially unique.

Proposition 5.2.10. *Let $f \in \mathbb{F}[X]$ be a regular polynomial. If*

$$f = \sum_{i=1}^n \alpha_i \ell_i^{e_i} = \sum_{i=1}^n \beta_i t_i^{d_i}$$

with ℓ_i, t_i linear forms and $\underline{e} = (e_1, \ldots, e_n), \underline{d} = (d_1, \ldots, d_n) \in E_n$, then, $e_i = d_i$ for all i, and there exists a permutation $\sigma \in \mathfrak{S}_n$ such that $\alpha_i \ell_i^{e_i} = \beta_{\sigma(i)} t_{\sigma(i)}^{d_{\sigma(i)}}$ if $e_i \geq 3$.

Proof. By Proposition 5.2.9 we have that $e_i = d_i$ for all i. Assume first that $e_s > 1$. By Lemma 5.2.4 we have that $\det(H_f) = c \prod_{e_i \geq 3} \ell_i^{e_i - 2} = b \prod_{e_i \geq 3} t_i^{e_i - 2}$ and the result follows. If $e_n = 1$ we obtain the same result by using Lemma 5.2.8 instead of Lemma 5.2.4. □

Theorem 5.2.11. *There exists a polynomial-time randomized algorithm* `Build1` *that receives as input a blackbox access to a regular polynomial $f \in \mathbb{F}[X]$ and finds an optimal decomposition of f in the Affine Powers model if AffPow$(f) = n$, or rejects otherwise.*

Proof. We compute a blackbox for $D(X) = \det(H_g(X))$ and distinguish two cases depending on whether it vanishes or not.
Case $D \neq 0$: if D does not split into degree 1 factors, we reject. Otherwise we write $D = c \cdot \prod_{i=1}^{t} \ell_i^{m_i}$ with $c \in \mathbb{F}^*$ and $\ell_1, \dots, \ell_t$ affine forms. If $t > n$, we reject. Consider the $t \times s$ matrix A whose rows are the $[\ell_i]$'s, and the matrix b whose entries are the constant terms. If the system $A \cdot X = -b$ has no solution, we reject. Otherwise, let X_0 be one solution, and consider $h(X) = g(X + X_0)$ so that $(\ell_i(X - X_0))^{m_i+2} = [\ell_i]^{m_i+2}$ is a homogeneous polynomial of degree $m_i + 2 \geq 3$. By Lemma 5.2.4, these are the only terms of degree ≥ 3 in an expression of f as a combination of n affine powers. Therefore, if $[h]_{\geq 3} \notin \left\langle [\ell_i]^{m_i+2} \right\rangle$, then we reject. Otherwise, let (α_i) be such that $h = \sum_{i=1}^{t} \alpha_i [\ell_i]^{m_i+2} + [h]_{\leq 2}$. We express $[h]_{\leq 2} = \sum_{i=1}^{r} \beta_i t_i^{e_i}$ as in Theorem 5.2.6. If $r + t \neq n$, then reject. Otherwise output the optimal expression of $f(X) = g(B^{-1} \cdot X) = h(B^{-1} \cdot X - X_0)$.
Case $D = 0$: for all k such that $\det([H_f(X)]_{k,k}) \neq 0$, we repeat the previous procedure. If no such k exists, or if we reject for all such k, then we reject; otherwise we output the optimal expression.

Correctness of the algorithm is justified by Lemma 5.2.4 and Lemma 5.2.8. □

In Sections 5.3 and 5.4, we try to generalize this algorithm in two natural ways: by allowing the affine forms to be repeated, or by allowing more than EssVar(f) different affine forms.

5.3 Repeated affine forms

In this section, we investigate the case where there exists a decomposition of a regular polynomial f with n different affine forms that can be used possibly several times in the decomposition. Since f is regular, the n affine forms are necessarily linearly independent. In other words, we want to test if $f \equiv g$ with $g = \sum_{i=1}^{n} \sum_{j=1}^{t_i} \alpha_{i,j} \, x_i^{e_{i,j}}$. In such a scenario, we can write f as a sum of univariate polynomials: $f = \sum_{i=1}^{n} g_i(\ell_i(X))$ with $g_i(x) = \sum_{j=1}^{t_i} \alpha_{i,j} \, x^{e_{i,j}}$ and ℓ_i an affine form. Conversely, if f can be written in this way, we can obtain a decomposition with n linearly independent affine forms by taking a decomposition for each univariate polynomial g_i. This motivates the study of the following problem of *univariate decomposition*:

Problem 5.3.1. *Given $f \in \mathbb{F}[X]$, is $f \equiv g$ with $g = \sum_{i=1}^{n} g_i(x_i)$?*

Yet, this problem does not completely capture the problem of finding an optimal decomposition in the AffPow model: indeed, even if a polynomial has a univariate

decomposition $f(X) = \sum_{i=1}^{n} g_i(\ell_i(X))$, we have no guarantee that taking an optimal AffPow decomposition for each g_i will yield an optimal decomposition of f in Model 1. In the following, we first study Problem 5.3.1 on its own, and then solve the bivariate case by proving that indeed an optimal univariate decomposition is optimal in Model 1.

5.3.1 Decomposing a polynomial as sum of univariates

The goal of this section is to design an algorithm in Theorem 5.3.4 that receives as input a regular polynomial f and computes a univariate decomposition if there is one. Notice first that Problem 5.3.1 is equivalent to testing if there exist univariate polynomials $(g_i(x))$ such that $f \sim g_1(x_1) + \cdots + g_n(x_n)$. A more general version of this problem has been already studied in Appendix C of [43] where the following result is proved:

Theorem 5.3.2. *[43, Theorem C.2] Given an n-variate polynomial $f(X) \in \mathbb{F}[X]$, there exists an algorithm that finds a decomposition of f as*

$$f(A \cdot X) = p(x_1, \ldots, x_t) + q(x_{t+1}, \ldots, x_n),$$

with A invertible, if it exists, in randomized polynomial time provided $\det(H_f)$ is a regular polynomial, i.e. it has n essential variables.

In the following, we will see how to find a univariate decomposition even if the determinant of the Hessian is not regular. The following result both provides the main ideas and justifies the correctness of the algorithm we propose.

Proposition 5.3.3. *Let $f \in \mathbb{F}[X]$, and let the g_i's be univariate polynomials sorted by decreasing degree. Let $d_i := \deg(g_i)$ and $k := \max\{i : d_i \geq 3\}$. Let $\ell_1, \ldots, \ell_n$ be linear forms such that $f = \sum_{i=1}^{n} g_i(\ell_i)$. Then,*

$$\det(H_f(X)) = c \cdot \prod_{i=1}^{k} \prod_{j=1}^{d_i - 2} (\ell_i - \alpha_{i,j}),$$

where $c \in \mathbb{F}$, and $\alpha_{i,j}$ are the roots of $g_i''(x)$ for $1 \leq i \leq k$.
Moreover, if $\ell_1, \ldots, \ell_n$ are linearly independent, for any solution $X_0 \in \mathbb{F}^n$ to the system $B \cdot X_0 = (\alpha_{1,1}, \ldots, \alpha_{k,1})^T$, where B is the $k \times n$ matrix whose rows are the coefficients of the $\ell_1, \ldots, \ell_k$, we have that

(a) $[f(X + X_0)]_{\geq 3} = \sum_{i=1}^{k} h_i(\ell_i)$ for some unique $h_i \in \mathbb{F}[x]$, and

(b) EssVar$([f(X + X_0)]_2) = |\{i \,|\, \deg(g_i) = 2\}|$.

Proof. By Lemma 5.2.3, $\det(H_f(X)) = (\det(A))^2 \prod_{i=1}^{n} g_i''(\ell_i)$, where A is the matrix whose i-th row corresponds to the coefficients of ℓ_i. It suffices to write $g_i''(x) = c_i \prod_{i=1}^{d_i - 2}(x - \alpha_{i,j})$ for all $i \in [\![1, k]\!]$ to get the first part of the result.

We assume now that $\ell_1, \dots, \ell_n$ are linearly independent. To prove (a), we observe that

$$[f(X + X_0)]_{\geq 3} = \sum_{i=1}^{k} [g_i(\ell_i(X + X_0))]_{\geq 3} = \sum_{i=1}^{k} [g_i(\ell_i + \alpha_{i,1})]_{\geq 3};$$

so it suffices to take $h_i(x) := [g_i(x + \alpha_{i,1})]_{\geq 3}$ for $i = 1, \dots, k$. Uniqueness of h_i comes directly from the fact that $\ell_1, \dots, \ell_k$ are linearly independent.
To prove (b) we observe first that $[g_i(x + \alpha_{i,1})]_2 = 0$ because $g_i''(\alpha_{i,1}) = 0$ for $i = 1, \dots k$. Since ℓ_i is a linear form this implies that $[g_i(\ell_i + \alpha_{i,1})]_2 = 0$ for all $i \in [\![1, k]\!]$, and then

$$[f(X + X_0)]_2 = \sum_{d_i=2} [g_i(\ell_i(X + X_0))]_2 = \sum_{d_i=2} \gamma_i \ell_i^2,$$

for some $\gamma_i \neq 0$ and, thus, (b) follows from Proposition 5.2.5. □

Theorem 5.3.4. *There exists a polynomial-time randomized algorithm that receives as input a blackbox access to a regular polynomial $f \in \mathbb{F}[X]$ and finds a univariate decomposition of f if such a decomposition exists, or rejects otherwise.*

Proof. The algorithm works as follows. We first compute $D(X) = \det(H_f(X))$ and separate two cases.
Case $D \neq 0$: if $D(X)$ does not split into polynomials of degree 1, we reject. Otherwise we take $\ell_1, \dots, \ell_k$ all the non-proportional linear parts of the factors and build the associated $k \times n$ matrix B. If $\operatorname{rank}(B) \neq k$, we reject. Otherwise, we gather the factors to write

$$D(X) = c' \cdot \prod_{i=1}^{k} p_i(\ell_i(X)) \quad \text{with} \quad p_i(x) = c_i \prod_{j=1}^{d_i} (x - \alpha_{ij})$$

where $c_1, \dots, c_k, c'$ are nonzero constants. Now we take X_0 a solution of $B \cdot X_0 = (\alpha_{1,1}, \dots, \alpha_{k,1})^T$ and consider $g(X) = f(X + X_0)$. Let $h_1, \dots, h_k \in \mathbb{F}[x]$ be the only polynomials so that $[g]_{\geq 3} = \sum_{i=1}^{k} h_i(\ell_i)$ (or reject if they do not exist). Then we use the greedy algorithm of Section 5.2.2 to write $[g]_2$ as $\sum_{i=k+1}^{m} \gamma_i \ell_i^2$ for some new linear forms $\ell_{k+1}, \dots, \ell_m$. If $m \neq n$ or $\ell_1, \dots, \ell_n$ are not linearly independent, we reject. Otherwise, we express $[g]_{\leq 1} = \sum_{i=1}^{n} \delta_i \ell_i + b$ for some $\delta_1, \dots, \delta_n, b \in \mathbb{F}$. Putting all together we have that g can be written as $\sum_{i=1}^{k} (h_i(\ell_i) + \delta_i \ell_i) + \sum_{i=k+1}^{n} \left(\gamma_i \ell_i^2 + \delta_i \ell_i\right) + b$, and we finally get an univariate decomposition of f as $f(X) = \sum_{i=1}^{n} q_i(t_i)$ with

$$q_i(x) := \begin{cases} h_1(x) + \delta_1 x + b & \text{for } i = 1 \\ h_i(x) + \delta_i x & \text{for } i = 2, \dots, k \\ \gamma_i x^2 + \delta_i x & \text{for } i = k+1, \dots, n \end{cases}$$

and $t_i(X) := \ell_i(X - X_0)$ an affine form for all $i \in [\![1, n]\!]$.

Case $D = 0$: this case happens whenever f is a sum of univariate polynomials where one of the g_i's is of degree 1. To handle this situation we proceed similarly to Section 5.2.3. Again we can have at most one g_i of degree 1 since otherwise the number of essential variables of f would not be n. In this case we use the following more general version of Lemma 5.2.8 which can be proved using similar techniques.

Lemma 5.3.5. *Let $f \in \mathbb{F}[X]$ be a regular polynomial that can be written as $f(X) = \sum_{i=1}^{n-1} g_i(\ell_i(X)) + \ell_n(X)$ where $\ell_1, \ldots, \ell_n$ are affine forms, and g_i is a univariate polynomial of degree ≥ 2 for all i. Then there exists an integer $k \in [\![1, n]\!]$ and $c \neq 0$ such that*

$$\det([H_f(X)]_{k,k}) = c \cdot \prod_{i=1}^{n-1} g_i''(\ell_i(X))$$

Hence, for all k such that $D_k := \det([H_f(X)]_{k,k}) \neq 0$, we proceed as before with D_k and try to express $[f]_{\geq 2}$ as $[\sum_{i=1}^{n-1} q_i(t_i)]_{\geq 2}$. If we succeed, we set $t_n := f - \sum_{i=1}^{n-1} q_i(t_i)$, $q_n := x$ and output the optimal expression. If there is no k with $D_k \neq 0$, or if we reject for all such k, then we reject. □

5.3.2 The bivariate case

Let $f \in \mathbb{F}[x_1, x_2]$ be a bivariate polynomial that admits a univariate decomposition $f = f_1(\ell_1) + f_2(\ell_2)$. In this case, we are going to describe how the optimal expression of f can be obtained from the (univariate) optimal expressions of f_1 and f_2, by putting together, if possible, the terms of degree ≤ 1 in one bivariate polynomial. To prove that this gives the optimal expression we are going to define the UnivAffPow(f) as the number of terms obtained in this procedure and we are going to prove that UnivAffPow$(f) =$ AffPow(f).
More precisely, set $s_i \stackrel{\text{def}}{=}$ AffPow(f_i) and write

$$f_i = \sum_{j=1}^{s_i} \alpha_{i,j}(x_i + a_{i,j})^{e_{i,j}},$$

with $e_{i,1} \leq \cdots \leq e_{i,s_i}$. We separate two cases: if there exist optimal expressions of f_1 and f_2 with $e_{1,1} \leq 1$ and $e_{2,1} \leq 1$; we consider them, we set $\ell := \alpha_{1,1}(x_1 + a_{1,1})^{e_{1,1}} + \alpha_{2,1}(x_2 + a_{2,1})^{e_{2,1}}$, and observe that

$$f = \sum_{j=2}^{s_1} \alpha_{1,j}(x_1 + a_{1,j})^{e_{1,j}} + \sum_{j=2}^{s_2} \alpha_{2,j}(x_2 + a_{2,j})^{e_{2,j}} + \ell.$$

In this case we define UnivAffPow$(f) := s_1 + s_2 - 1$. Otherwise, we define UnivAffPow$(f) := s_1 + s_2$.
We prove in Proposition 5.3.9 that AffPow$(f) =$ UnivAffPow(f). This result is a consequence of the some lemmas. Before proving the first of this lemmas, recall

from Proposition 1.1.17 that every univariate polynomial g of degree d satisfies that $\mathrm{AffPow}(g) \leq r \stackrel{\text{def}}{=} \lceil \frac{d+1}{2} \rceil$. Moreover, if $\mathrm{AffPow}(g) = r$, then g admits and expression as $\sum_{i=1}^{r} \alpha_i (x + a_i)^{e_i}$ with $d = e_1 > e_2 > \cdots > e_r$ with $e_i - e_{i+1} \geq 2$ for all i and, thus, $e_r \in \{0, 1\}$ (see [29, Proposition 18] and its proof).

Lemma 5.3.6. *Let $f_i \in \mathbb{F}[x_i]$ polynomials of degree d_i for $i = 1, 2$. Then,*

$$\mathit{UnivAffPow}(f_1 + f_2) \leq \left\lceil \frac{d_1 + 1}{2} \right\rceil + \left\lceil \frac{d_2 + 1}{2} \right\rceil - 1.$$

Proof. Let $s_i := \mathrm{AffPow}(f_i)$ for $i = 1, 2$. If $s_i < \left\lceil \frac{d_1+1}{2} \right\rceil$ for some i, the result follows directly since $\mathrm{UnivAffPow}(f_1 + f_2) \leq s_1 + s_2$. Otherwise, $s_i = \left\lceil \frac{d_1+1}{2} \right\rceil$ for $i = 1, 2$; in this case both f_i can be written in an optimal way by using terms of degree ≤ 1; hence, $\mathrm{UnivAffPow}(f_1 + f_2) = s_1 + s_2 - 1$, proving the result □

Lemma 5.3.7. *Let $s, d \in \mathbb{Z}^+$ and $b_1, \ldots, b_s$ different nonzero elements of $\mathbb{F}$. If*

$$\lambda_1 x_1^d + \lambda_2 x_2^d = \sum_{i=1}^{s} \gamma_i (x_1 + b_i x_2)^d,$$

with $\lambda_1, \lambda_2 \in \mathbb{F}$ and $\gamma_i \in \mathbb{F}$ not all zero, then $s \geq d$. Moreover, if $\lambda_1 = 0$ or $\lambda_2 = 0$ then $s \geq d + 1$; and if $\lambda_1 = \lambda_2 = 0$, then $s \geq d + 2$.

Proof. Consider the evaluation map φ induced by $x_1 \mapsto x$, $x_2 \mapsto x + 1$. Then, $\varphi(x_1 + b_i\, x_2) = (1 + b_i)x + b_i = (1 + b_i)(x + \frac{b_i}{1+b_i})$. Thus, the linear dependency of $\{x_1^d, x_2^d, (x_1 + b_i x_2)^d \,|\, 1 \leq i \leq s\}$ implies linear dependency of $\{x^d, (x + 1)^d, (x + \frac{b_i}{1+b_i})^d \,|\, 1 \leq i \leq s\}$. Hence, the result follows from Proposition 2.2.3. □

Lemma 5.3.8. *Let $f = \sum_{i=1}^{s} \alpha_i (x_1 + b_i x_2 + c_i)^{e_i} \in \mathbb{F}[x_1, x_2]$ be a polynomial of degree ≥ 2, with $\alpha_i \in \mathbb{F}$ and $b_i \neq 0$ for all $i \in [\![1, s]\!]$. If $f = f_1 + f_2$ with $f_i \in \mathbb{F}[x_i]$, then $s \geq \mathit{UnivAffPow}(f_1 + f_2)$.*

Proof. We assume without loss of generality that $\alpha_i \neq 0$ for all i. Let $\ell_i := x_1 + b_i x_2 + c_i$ for $i = 1, \ldots, s$, and $d_j := \deg(f_j)$ for $j = 1, 2$. We know that $s_i := \mathrm{AffPow}(f_i) \leq \lceil (d_i + 1)/2 \rceil$. For all $e \in \mathbb{N}$, consider $[f]_e$ the homogeneous component of degree e of f. We have:

$$[f_1]_e + [f_2]_e = \sum_{e_i \geq e} \alpha_i [\ell_i^{e_i}]_e \in \langle (x_1 + b_i x_2)^e \,|\, e_i \geq e \rangle.$$

We assume that $d_1 \geq d_2$ and separate two cases:
Case 1: $d_1 > d_2$. We have that $0 \neq [f_1]_{d_1} = \sum_{e_i \geq d_1} \gamma_i (x_1 + b_i x_2)^{d_1}$ for some $\gamma_i \in \mathbb{F}$, hence by Lemma 5.3.7 there are at least $d_1 + 1$ exponents e_i that are $\geq d_1$. As a consequence, by Lemma 5.3.6 we get that if $d_2 = d_1 - 1$, then

$$s \geq d_1 + 1 = \frac{d_1 + d_2 + 3}{2} = \left\lceil \frac{d_1 + 1}{2} \right\rceil + \left\lceil \frac{d_2 + 1}{2} \right\rceil > \mathrm{UnivAffPow}(f_1 + f_2);$$

and if $d_2 < d_1 - 1$, then

$$s \geq d_1 + 1 \geq \frac{d_1 + d_2 + 4}{2} \geq \left\lceil \frac{d_1 + 1}{2} \right\rceil + \left\lceil \frac{d_2 + 1}{2} \right\rceil > \text{UnivAffPow}(f_1 + f_2).$$

Case 2: $d_1 = d_2$. We have that $[f_1]_{d_1} + [f_2]_{d_2} = \sum_{e_i \geq d_1} \gamma_i (x_1 + b_i x_2)^{d_1}$ for some $\gamma_i \in \mathbb{F}$, hence by Lemma 5.3.7 we have that there are at least d_1 exponents bigger than or equal to d_1. As a consequence, by Lemma 5.3.6:

$$s \geq |\{b_i \,|\, e_i \geq d_1\}| \geq d_1 \geq 2\left\lceil \frac{d_1 + 1}{2} \right\rceil - 2 \geq \text{UnivAffPow}(f_1 + f_2) - 1.$$

If one of these inequalities is strict, the result is proved; so it only remains to prove that they all cannot be equalities at the same time.
Let us assume by contradiction that they are all equalities. In particular, we have that then $b_i \neq b_j$ for all $1 \leq i < j \leq s$. We claim that $e_i = d_1$ for all i. Otherwise, taking $e := \max(e_i) > d_1$ and observing the homogeneous component of degree e, we get that

$$0 = \sum_{e_i = e} \alpha_i \, (x_1 + b_i x_2)^e;$$

but again by Lemma 5.3.7, this implies that the number of ℓ_i with $e_i = e$ is at least $e + 2 \geq d_1 + 3 > s$, a contradiction. Hence,

$$f_1 + f_2 = \sum_{i=1}^{s} \alpha_i (x_1 + b_i x_2 + c_i)^{d_1}.$$

Now set $\beta_i \in \mathbb{F}$ the (only) root of the derivative of order $d_1 - 1$ of f_i and consider $g_i(x_i) := f_i(x_i + \beta_i)$. We have that $g_1 + g_2 = \sum_{i=1}^{s} \alpha_i (x_1 + b_i x_2 + c_i')^{d_1}$. However, the homogeneous component of degree $d_1 - 1$ of g_1 and g_2 is zero; therefore if we observe the homogeneous component of degree $d_1 - 1$ in this expression we get that

$$0 = \sum_{i=1}^{s} d_1 \alpha_i c_i' (x_1 + b_i x_2)^{d_1 - 1}.$$

Since $s < d_1 + 2$, Lemma 5.3.7 yields that $c_i' = 0$ for all i. Since $f_1(x_1 + \beta_1)$, $f_2(x_2 + \beta_2)$ are univariate polynomials, we have that $f_1(x_1 + \beta_1) + f_2(x_2 + \beta_2) = \gamma_1 x_1^{d_1} + \gamma_2 x_2^{d_2}$. However, this implies that $\text{AffPow}(f_1) = \text{AffPow}(f_2) = 1$ and, then, $1 \geq \text{UnivAffPow}(f) - 1 = d_1 = d_2$, a contradiction. □

As a consequence of Lemma 5.3.8, we obtain the main result of this subsection:

Proposition 5.3.9. *Let $f_1 \in \mathbb{F}[x_1]$ and $f_2 \in \mathbb{F}[x_2]$, then*

$$\textit{AffPow}(f_1 + f_2) = \textit{UnivAffPow}(f_1 + f_2).$$

Proof. It is obvious that $\text{AffPow}(f_1+f_2) \leq \text{UnivAffPow}(f_1+f_2)$. Let $s := \text{AffPow}(f_1+f_2)$ and consider $f_1+f_2 = \sum_{i=1}^{s} \ell_i^{e_i}$ an optimal expression of f_1+f_2 in Model 1. We write $\ell_i = a_i x_1 + b_i x_2 + c_i$ with $a_i, b_i, c_i \in \mathbb{F}$. Set $g := f_1 + f_2 - \sum_{\substack{b_i=0 \\ \text{or } c_i=0}} \ell_i^{e_i}$. Clearly, g is a sum of two univariate polynomials and it can be written as

$$g = \sum_{\substack{a_i \neq 0 \\ b_i \neq 0}} \ell_i^{e_i} = \sum_{\substack{a_i \neq 0 \\ b_i \neq 0}} a_i^{e_i} \left(x_1 + \frac{b_i}{a_i} x_2 + \frac{c_i}{b_i} \right)^{e_i}.$$

Setting $r := |\{i \,|\, a_i \neq 0 \text{ and } b_i \neq 0\}|$, we have that $\text{UnivAffPow}(g) \leq r$ by Lemma 5.3.8. Hence we can rewrite g as $g = \sum_{i=1}^{r'} (\alpha_i x + \beta_i y + \gamma_i)^{d_i}$ with either $\alpha_i = 0$, $\beta_i = 0$ or $d_i = 1$, and $r' \leq r$. As a consequence, $f = \sum_{i=1}^{r'} (\alpha_i x + \beta_i y + \gamma_i)^{d_i} + \sum_{\substack{b_i=0 \\ \text{or } c_i=0}} \ell_i^{e_i}$ is an expression of f with $s - r + r'$ terms. Since $s - r + r' \leq s$, this shows that $\text{UnivAffPow}(f_1+f_2) \leq s = \text{AffPow}(f_1+f_2)$. □

5.4 Allowing more affine forms

In what follows we investigate the case where the number of affine forms used to express f in Model 1 is greater than the number of essential variables. The most basic such case is when $f \equiv g$ with $g = \sum_{i=1}^{n} x_i^{e_i} + \ell^e$, where ℓ is an affine form and $e \in \mathbb{N}^*$. Let us first see why the algorithm of Section 5.2 cannot be straightforwardly generalised to recover the optimal expression of f. We set $h := g - \ell^e$ so that we have $H_g = H_h + H_{\ell^e}$ by linearity of differentiation. Notice that $H_{\ell^e} = e^{\underline{2}}\, \ell^{e-2} \beta\beta^T$, where β is the column vector associated to the coefficients of ℓ and $e^{\underline{i}} := e \cdots (e-i+1)$. In order to compute $\det(H_g)$, we use the *matrix determinant lemma* that we first recall.

Lemma 5.4.1. *Let $A \in \mathcal{M}_n(\mathbb{F})$ and $u, v \in \mathbb{F}^n$ two column vectors. Then,*

$$\det(A + uv^T) = \det(A) + v^T \operatorname{adj}(A) u,$$

where $\operatorname{adj}(A)$ denotes the adjugate matrix of A.

We therefore have $\det(H_g) = \det(H_h) + e^{\underline{2}}\, \ell^{e-2} \beta^T \operatorname{adj}(H_h) \beta$. Hence, if $f = g(A \cdot X + b)$, Lemma 5.2.4 implies that

$$\det(H_f) = \det(A)^2 \left(\prod_{i=1}^{n} e_i^{\underline{2}}\, \ell_i(X)^{e_i - 2} + e^{\underline{2}}\, \ell(A \cdot X + b)^{e-2}\, P(X) \right)$$

with $P(X) = \sum_{i=1}^{n} \beta_i^2 \left(\prod_{j \neq i} e_j^{\underline{2}}\, \ell_j(X)^{e_i - 2} \right) \in \mathbb{F}[X]$. In most cases neither the ℓ_i's nor ℓ are factors of $\det(H_f)$, which makes the (straightforward generalization of) algorithm of Section 5.2 fail.

5.4.1 Higher order Hessian

The main idea we propose to generalize the algorithm is to consider an extension of the Hessian by looking at higher order derivatives. We therefore consider all monomials $x_i x_j$ of degree 2, and we build the *4-th order Hessian* $\tilde{H}_f \in \mathcal{M}_{n^2}(\mathbb{F}[X])$ whose entries are:

$$(\tilde{H}_f)_{(a,b),(i,j)} = \frac{\partial^4 f}{\partial x_a \partial x_b \partial x_i \partial x_j}$$

This extension is quite natural and the following analogue of Lemma 5.2.3 shows that it also behaves well with a change of basis. For two matrices $A = (A_{ij}), B = (B_{kl})$ of sizes $m_1 \times n_1$ and $m_2 \times n_2$, respectively, we denote by $A \otimes B$ its *Kronecker product*. That is, $A \otimes B = (C_{(i,k),(j,l)})$ is the $m_1 m_2 \times n_1 n_2$ matrix with $C_{(i,k),(j,l)} = A_{i,j} B_{k,l}$ (see [37] for a further study of Kronecker product).

Lemma 5.4.2. *Let $g \in \mathbb{F}[X]$ be an n-variate polynomial. Let $A \in \mathcal{M}_n(\mathbb{F})$ be a linear transformation, and let $b \in \mathbb{F}^n$. Let $f(X) = g(A \cdot X + b)$. Then,*

$$\tilde{H}_f(X) = (A \otimes A)^T \cdot \tilde{H}_g(A \cdot X + b) \cdot (A \otimes A)$$

Proof. By the chain rule for differentiation we have for all $1 \leq a, b, c, d \leq n$:

$$\frac{\partial^4 f}{\partial x_a \partial x_b \partial x_c \partial x_d} = \sum_{i,j,k,l \in [n]} A_{i,a} A_{j,b} A_{k,c} A_{l,d} \cdot \frac{\partial^4 g}{\partial x_i \partial x_j \partial x_k \partial x_l}(A \cdot X + b)$$

We set $E = \tilde{H}_f$ and $H = \tilde{H}_g(A \cdot X + b)$ so that we have

$$E_{(a,b),(c,d)} = \sum_{(i,j),(k,l) \in [n]^2} ((A \otimes A)^T)_{(a,b),(i,j)} H_{(i,j),(k,l)} (A \otimes A)_{(k,l),(c,d)}.$$

□

This result, along with the fact that $\det(A \otimes A) = \det(A)^{2n}$, could seem promising at first, but in fact we always have $\det(\tilde{H}_f) = 0$ since Schwarz theorem (symmetry of second derivatives) implies that lines (a, b) and (b, a) of $\tilde{H}_f$ are equal. We will therefore consider the *symmetric 4-th order Hessian* $\overline{H}_f$, which is the submatrix of $\tilde{H}_f$ where we remove the rows (a, b) with $a > b$ and the columns (i, j) with $i > j$. In other words, $\overline{H}_f \in \mathcal{M}_{\binom{n+1}{2}}(\mathbb{F}[X])$ and its entries are:

$$\forall a \leq b, i \leq j, \quad (\overline{H}_f)_{(a,b),(i,j)} = \frac{\partial^4 f}{\partial x_a \partial x_b \partial x_i \partial x_j}$$

Again, we can prove an analogue of Lemma 5.2.3 regarding a change of basis.

Lemma 5.4.3. *Let $g \in \mathbb{F}[X]$ be an n-variate polynomial. Let $A \in \mathcal{M}_n(\mathbb{F})$ be a linear transformation, and let $b \in \mathbb{F}^n$. Let $f(X) = g(A \cdot X + b)$. Then,*

$$\overline{H}_f(X) = (A \oslash A)^T \cdot \overline{H}_g(A \cdot X + b) \cdot (A \oslash A)$$

where the matrix $A \oslash A$ is defined as follows: for all $a \leq b$ and $i \leq j$

$$(A \oslash A)_{(a,b),(i,j)} = \begin{cases} A_{a,i}A_{b,j} + A_{a,j}A_{b,i} & \text{if } a \neq b \\ A_{a,i}A_{a,j} & \text{otherwise} \end{cases}$$

Proof. We set $E = \overline{H}_f$ and $H = \overline{H}_g(A \cdot X + b)$, and we use again the chain rule for differentiation for $a \leq b, c \leq d$:

$$\begin{aligned} E_{(a,b),(c,d)} &= \sum_{i,j,k,l \in [n]} A_{i,a}A_{j,b}A_{k,c}A_{l,d} \cdot H_{(i,j),(k,l)} \\ &= \sum_{i,j,k<l \in [n]} A_{i,a}A_{j,b} \cdot H_{(i,j),(k,l)} \cdot (A_{k,c}A_{l,d} + A_{l,c}A_{k,d}) \\ &\qquad + \sum_{i,j,k=l \in [n]} A_{i,a}A_{j,b} \cdot H_{(i,j),(k,k)} \cdot A_{k,c}A_{k,d} \\ &= \sum_{i,j,k \leq l \in [n]} A_{i,a}A_{j,b} \cdot H_{(i,j),(k,l)} \cdot (A \oslash A)_{(k,l),(c,d)} \\ &= \sum_{i \leq j, k \leq l \in [n]} (A \oslash A)_{(i,j),(a,b)} \cdot H_{(i,j),(k,l)} \cdot (A \oslash A)_{(k,l),(c,d)} \end{aligned}$$

□

We now prove that given a regular matrix A, the matrix $A \oslash A$ is also regular. As a consequence, if $f = g(A \cdot X + b)$ then $\det(\overline{H}_f(X)) = c \cdot \det(\overline{H}_g(A \cdot X + b))$ with c a nonzero constant. To do so, we will relate $A \oslash A$ with the *symmetric Kronecker product* (see e.g. [22]) which is a $\binom{n+1}{2} \times \binom{n+1}{2}$ matrix defined as $A \otimes_S B = \frac{1}{2}Q(A \otimes B + B \otimes A)Q^T$ with

$$\forall i \leq j, k, l, \quad Q_{(i,j),(k,l)} = \begin{cases} 1 & \text{if } i = j = k = l \\ \frac{1}{\sqrt{2}} & \text{if } i = k \neq j = l, \text{or } i = l \neq j = k \\ 0 & \text{otherwise} \end{cases}$$

In particular, we will use the following result that can be easily derived from the properties of $\otimes_S$ described in [3].

Lemma 5.4.4. *We have* $\det(A \otimes_S A) = \det(A)^{n+1}$.

Proof. We will prove this result for a diagonalizable matrix and conclude by density. Let (λ_i) be the eigenvalues of A, repeated with multiplicity. Then the eigenvalues of $A \otimes_S A$ are given by $\frac{1}{2}(\lambda_i\lambda_j + \lambda_j\lambda_i) = \lambda_i\lambda_j$ for $i \leq j$. We therefore have $\det(A \otimes_S A) = \prod_{i \leq j} \lambda_i\lambda_j = \prod_{i=1}^n \lambda_i^{n+1} = \det(A)^{n+1}$. □

Lemma 5.4.5. *We have* $\det(A \oslash A) = \det(A)^{n+1}$.

Proof. Consider the matrix $\overline{Q} := D \cdot Q$ where D is a diagonal matrix defined as

$$D_{(i,j),(i,j)} = \begin{cases} \sqrt{2} & \text{if } i = j \\ 1 & \text{otherwise} \end{cases}$$

Then the coefficients of the matrix $B = \overline{Q}(A \otimes A)\overline{Q}^T$ are given by $B_{(i,j),(k,l)} = A_{i,k}A_{j,l} + A_{i,l}A_{j,k}$. In particular, we have $\det(B) = 2^n \det(A \oslash A)$. Moreover, since $\overline{Q} = D \cdot Q$, we also have $\det(B) = \det(D)^2 \det(Q(A \otimes A)Q^T) = 2^n \det(A \otimes_S A)$. Finally, we have $\det(A \oslash A) = \det(A \otimes_S A) = \det(A)^{n+1}$. □

Corollary 5.4.6. *Let* $g \in \mathbb{F}[X]$ *be an* n*-variate polynomial. Let* $A \in \mathrm{GL}_n(FF)$ *be a linear transformation, and let* $b \in \mathbb{F}^n$*. Let* $f(X) = g(A \cdot X + b)$*. Then,*

$$\det(\overline{H}_f(X)) = c \cdot \det(\overline{H}_g(A \cdot X + b))$$

where $c \in \mathbb{F}$ *is a nonzero constant.*

5.4.2 The bivariate case

For $f \in \mathbb{F}[x, y]$, the preceding results directly allow us to detect if $f(x, y) \equiv g(x, y) = x^{e_1} + y^{e_2} + \ell^{e_3}$ where $\ell = \alpha_1 x + \alpha_2 y + \alpha_0$ is an affine form with $\alpha_1, \alpha_2 \neq 0$, and $e_i \geq 5$ for all i. For this purpose, we build the symmetric 4-th order Hessian $\overline{H}_f$:

$$\overline{H}_f = \begin{array}{c} \\ xy \\ x^2 \\ y^2 \end{array} \begin{array}{c} \begin{array}{ccc} xy & x^2 & y^2 \end{array} \\ \begin{pmatrix} \frac{\partial^4 f}{\partial x^2 \partial y^2} & \frac{\partial^4 f}{\partial x^3 \partial y} & \frac{\partial^4 f}{\partial x \partial y^3} \\ \frac{\partial^4 f}{\partial x^3 \partial y} & \frac{\partial^4 f}{\partial x^4} & \frac{\partial^4 f}{\partial x^2 \partial y^2} \\ \frac{\partial^4 f}{\partial x \partial y^3} & \frac{\partial^4 f}{\partial x^2 \partial y^2} & \frac{\partial^4 f}{\partial y^4} \end{pmatrix} \end{array}$$

Let us compute the determinant of $\overline{H}_g$:

$$\overline{H}_g = \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & e_1^4 x^{e_1-4} & 0 \\ 0 & 0 & e_2^4 y^{e_2-4} \end{pmatrix}}_{=B} + e_3^4 \ell^{e_3-4} u u^T \quad \text{where } u = \begin{pmatrix} \alpha_1\alpha_2 \\ \alpha_1^2 \\ \alpha_2^2 \end{pmatrix}$$

By Lemma 5.4.1, we therefore have $\det(\overline{H}_g) = e_3^4 \ell^{e_3-4} u^T \operatorname{adj}(B) u$. In this case, the adjugate matrix of B is easy to compute:

$$\operatorname{adj}(B) = \begin{pmatrix} e_1^4 e_2^4 x^{e_1-4} y^{e_2-4} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

We thus have $\det(\overline{H}_g)(x, y) = c \cdot x^{e_1-4} y^{e_2-4} \ell^{e_3-4}$ with $c = e_1^4 e_2^4 e_3^4 \alpha_1^2 \alpha_2^2 \neq 0$. By Corollary 5.4.6, we obtain the following result.

Lemma 5.4.7. *Let $f(x, y)$ be a polynomial such that $f = \ell_1^{e_1} + \ell_2^{e_2} + \ell_3^{e_3}$ where ℓ_1, ℓ_2, ℓ_3 are affine forms whose linear parts are not proportional, and $e_i \geq 4$ for all i. Then we have*

$$\det(\overline{H}_f(X)) = c \cdot \prod_{i=1}^{3} \ell_i(x, y)^{e_i - 4}$$

where $c \in \mathbb{F}$ is a nonzero constant.

This result directly yields an algorithm for the case when all the e_i's are greater than 4, as one just has to factorize the symmetric 4-th order Hessian to recover the e_i's and the ℓ_i's.

5.4.3 The general case

In this section, we will design a randomized algorithm that can reconstruct a decomposition in Model 1 that uses up to $\binom{n+1}{2}$ distinct affine forms. However, it will not work for all input polynomials of such type. Indeed, it will work whenever all the exponents involved in the optimal expression of f are ≥ 5 and a certain matrix U, which depends on the affine forms involved, is invertible. We will conduct a randomized analysis to show that our method is correct with high probability (over the choice of the input polynomial and of the internal coin tosses of the algorithm). We begin by proving an analogue of Lemma 5.2.4 for the symmetric 4-th order Hessian.

Proposition 5.4.8. *Let $n \in \mathbb{N}^*$, $m := \binom{n+1}{2}$, and let $f = \sum_{i=1}^{m} \ell_i^{e_i}$ with $\ell_i = \sum_{j=1}^{n} b_{i,j} x_j + b_{i,0}$ affine forms and $e_i \geq 4$ for all i. Let U be the square $m \times m$ matrix with entries $U_{(i,j),k} := b_{k,i}\, b_{k,j}$ for all $1 \leq k \leq m$, $1 \leq i \leq j \leq n$. If $\det(U) \neq 0$, there exists $c \neq 0$ such that*

$$\det(\overline{H}_f(X)) = c \cdot \prod_{i=1}^{m} \ell_i^{e_i - 4},$$

Proof. By linearity of the symmetric 4-th order Hessian, we have

$$\overline{H}_f(X) = \sum_{k=1}^{m} \overline{H}_{\ell_k}(X) = \sum_{k=1}^{m} e_k^4\, \ell_k^{e_k - 4} (u_k \cdot u_k^T) = U \cdot D \cdot U^T,$$

where $D = \mathrm{Diag}(e_1^4\, \ell_1^{e_1 - 4}, \ldots, e_m^4\, \ell_m^{e_m - 4})$, and u_k is the column vector whose (i, j)-th entry is $b_{k,i} b_{k,j}$ with $1 \leq i \leq j \leq n$. Thus,

$$\det(\overline{H}_f(X)) = \det(U)^2 \prod_{k=1}^{m} e_k^4\, \ell_k^{e_k - 4}.$$

□

Now, we are going to prove that if the coefficients of the ℓ_i are chosen uniformly at random, then with a high probability we have $\det(U) \neq 0$. Thus, whenever $e_i \geq 5$ for all i, one can find ℓ_i as a factor of $\det(\overline{H}_f(X))$ of multiplicity $e_i - 4$.

Lemma 5.4.9. *Let $n \in \mathbb{N}^*$ and $m := \binom{n+1}{2}$, and consider the set of variables $\mathcal{V} := \{y_{(k,l),i} \mid 1 \leq k \leq l \leq n,\ 1 \leq i \leq n\}$. Let U be the $m \times m$ square matrix with entries $U_{(i,j),(k,l)} := y_{(k,l),i}\, y_{(k,l),j}$, where $1 \leq i \leq j \leq n$, $1 \leq k \leq l \leq n$. Then, $\det(U) \in \mathbb{Z}[\mathcal{V}]$ is a nonzero polynomial of degree $2m$.*

Proof. Since all the entries of the matrix are homogeneous polynomials of degree 2, it is clear that $\det(U)$ is either zero or a polynomial of degree $2m$. To prove that $\det(U) \neq 0$ it suffices to exhibit a nonzero evaluation of $\det(U)$. We consider the matrix $\tilde{U}$ given by the evaluation $y_{(k,l),i} \mapsto 1$ if $i \in \{k, l\}$; or $y_{(k,l),i} \mapsto 0$ otherwise. By ordering pairs (i, j) with $i = j$ first, we obtain the following shape

$$\tilde{U} = \begin{array}{c} \\ i=j \\ i<j \end{array} \begin{array}{c} \begin{array}{cc} k=l & k<l \end{array} \\ \begin{pmatrix} \mathrm{Id}_n & (*) \\ 0 & \mathrm{Id}_{m-n} \end{pmatrix} \end{array},$$

proving that $\det(\tilde{U}) = 1$ and therefore that $\det(U) \neq 0$. □

Theorem 5.4.10. *Let $n \geq 2$ and $m := \binom{n+1}{2}$. Let $\ell_i = \sum_{j=1}^{n} b_{i,j}x_j + b_{i,0} : 1 \leq i \leq m$ whose coefficients $b_{i,j}$ are taken uniformly at random from a finite set S and take $f := \sum_{i=1}^{m} \ell_i^{e_i} \in \mathbb{F}[X]$ with $e_i \geq 4$ for all i. Then, $\det(\overline{H}_f(X)) \neq 0$ with probability at least $1 - \frac{2m}{|S|}$.*

Proof. By Proposition 5.4.8, it is enough to show that $\det(U) \neq 0$, where U is the matrix defined by $U_{(i,j),k} = b_{k,i}b_{k,j}$. By Lemma 5.4.9 and the Schwartz-Zippel lemma, the probability that $\det(U) \neq 0$ is at least $1 - \frac{2m}{|S|}$. □

This theorem suggests a polynomial time algorithm for finding an optimal expression of a polynomial f with high probability when $\mathrm{AffPow}(f) \leq m = \binom{n+1}{2}$, the affine forms in optimal expression of f are chosen at random from a finite set and all the exponents involved are ≥ 5. It is enough to start with $k = m - 1$, choose randomly k affine forms $t_1, \ldots, t_k$ with exponents $d_i = 4$ and denote $g := f + \sum_{i=1}^{k} t_i^{d_i}$. If $D := \det(\overline{H}_f(X)) = 0$, we decrease the value of k by one unit and repeat the argument, or we reject if $k = 0$. If $D \neq 0$, we factorize it. If D splits into linear factors $l_1, \ldots, l_{m-k}$ of multiplicities $r_1, \ldots, r_{m-k}$ and $f \in \langle l_i^{r_i+4} \mid 1 \leq i \leq m - k \rangle$, then $\mathrm{AffPow}(f) = m - k$ and we output the optimal expression. Otherwise, we reject.

5.5 Univariate projections

In this section, we will proceed by reduction to the univariate case: we solve n univariate projections of the multivariate problem using algorithms from Chapter 4, and then "lift" them to a solution of the multivariate problem. The main result of this section will be an algorithm that finds the optimal reconstruction (it works only in the regime of uniqueness) under the condition on n_e being small, where n_e denotes the number of exponents smaller than e as in previous chapters.

5.5.1 Uniqueness

Strictly speaking the optimal expressions in Model 1 are never unique since for all $e \geq 2$ and ℓ an affine form, we have that $\ell^e = (\lambda\ell)^e$ for λ an e-th root of unity. To deal with this ambiguity, we use the notion of *essentially equal* expressions, as introduced in [44]. Given $f \in \mathbb{F}[X]$, we say that two expressions of $f = \sum_{i=1}^{s} \ell_i^{e_i} = \sum_{i=1}^{r} t_i^{d_i}$ are essentially equal if $r = s$ and there exists a permutation σ of $\{1, \ldots, s\}$ such that $\ell_i^{e_i} = t_{\sigma(i)}^{d_{\sigma(i)}}$ for all $i \in [\![1, s]\!]$. Likewise, we say that f has an *essentially unique* optimal decomposition in Model 1 if two optimal decompositions of f are always essentially equal.
The following result provides a sufficient condition for f to have an essentially unique optimal decomposition. It is an extension to the multivariate setting of Corollary 2.2.8.

Proposition 5.5.1. *Let $f \in \mathbb{F}[X]$ be a polynomial of the form:*

$$f = \sum_{i=1}^{s} \ell_i^{e_i}$$

where the ℓ_i are non constant affine forms, and ℓ_i is not proportional to ℓ_j whenever $e_i = e_j$. If $n_e \leq \sqrt{\frac{e}{2}}$ for all $e \in \mathbb{N}$, then AffPow$_\mathbb{F}(f) = s$ and the optimal representation of f is essentially unique.

Proof. Let $r := \text{AffPow}_\mathbb{F}(f) \leq s$ and let $f = \sum_{i=s+1}^{s+r} \ell_i^{e_i}$ be an optimal representation of f. We write $\ell_i = \sum_{j=1}^{n} a_{ij}x_j + a_{i0}$ for all $i \in [\![1, s+r]\!]$. Consider the ring homomorphism $\varphi : \mathbb{F}[X] \to \mathbb{F}[x]$ induced by $x_i \mapsto \omega_i x + \lambda_i$ where $\omega = (\omega_1, \ldots, \omega_n), \lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{F}^n$. We choose ω and λ so that

(1.a) $\varphi(\ell_i)^{e_i} = \varphi(\ell_j)^{e_j}$ if and only if $\ell_i^{e_i} = \ell_j^{e_j}$, and

(1.b) whenever $e_i = e_j$ with $1 \leq i < j \leq s$, then $c_i/b_i \neq c_j/b_j$

It is important to observe that a generic choice of $\omega, \lambda \in \mathbb{F}^n$ fulfills these two conditions. Then

$$\begin{array}{rcccl} \varphi(f) & = & \sum_{i=1}^{s} \varphi(\ell_i)^{e_i} & = & \sum_{i=1}^{s} b_i^{e_i}(x + c_i/b_i)^{e_i} \\ & = & \sum_{i=s+1}^{s+r} \varphi(\ell_i)^{e_i} & = & \sum_{i=s+1}^{s+r} b_i^{e_i}(x + c_i/b_i)^{e_i}. \end{array}$$

We consider the expression $\varphi(f) = \sum_{i=1}^{s} b_i^{e_i}(x + c_i/b_i)^{e_i}$ in the univariate Affine Power model. Since (1.b) holds and $n_e \leq \sqrt{\frac{e}{2}}$ for all $e \in \mathbb{N}$, by Corollary 2.2.8 we get that $r \geq \text{AffPow}_\mathbb{F}(\varphi(f)) = s \geq r$ and that both expressions for $\varphi(f)$ are the same. By (1.a) we conclude the result. □

Notice that this method of random univariate projections also allow to obtain the following upper bound on the maximum exponent in a expression, using Corollary 2.2.10.

Proposition 5.5.2. *Let $f \in \mathbb{F}[X]$ be a polynomial of degree d written as*

$$f = \sum_{i=1}^{s} \ell_i^{e_i}$$

with ℓ_i affine form, $e_i \in \mathbb{N}$. We set $e \stackrel{def}{=} \max\{e_i : i \in [\![1, s]\!]\}$. Then we have that

$$e < d + \frac{s^2}{2}.$$

In particular, since we have the upper bound $\mathrm{AffPow}_{\mathbb{F}}(f) \leq \binom{d+n-1}{d-1}$ (see Proposition 1.2.1), this yields an upper bound on the maximum exponent in an optimal expression. However, this bound on the maximum exponent is way too large to be used in efficient algorithms. We do not know if the bound is optimal, and it would be interesting to try to improve it by using other techniques or to find families of polynomials that requires large exponents in their optimal expressions.

5.5.2 Univariate projections

Our goal is to provide an algorithm that, given blackbox access to $f \in \mathbb{F}[X]$, computes $s = \mathrm{AffPow}(f)$ and an optimal expression for f. It is a multivariate analogue of Theorem 4.1.4 where the condition of "distinct nodes" is replaced by "the ℓ_i's in the decomposition are not proportional". The idea of the algorithm is to perform a random change of coordinates and then project to n univariate problems that we solve using Theorem 4.1.4. The same strategy applies to obtain similar results for all the algorithms of Chapter 4. One minor difficulty is that the univariate algorithms of [30] are presented for polynomials given in dense representation rather than in blackbox representation, but we can obtain the dense representation of a univariate polynomial by random evaluations and, then, interpolation.

Theorem 5.5.3. *Let $f \in \mathbb{F}[X]$ be a polynomial that can be written as*

$$f = \sum_{i=1}^{s} \ell_i^{e_i},$$

where the ℓ_i's are pairwise non-proportional linear forms, and $e_i \in \mathbb{N}$. Assume that $n_{i+1} \leq (3i/4)^{1/3} - 1$ for all $i \geq 2$. Then, AffPow$(f) = s$. Moreover, there is a randomized algorithm `MultiBuild`(f) *that, given access to a blackbox for f, computes the set of terms $T(f) = \{\ell_i^{e_i} : 1 \leq i \leq s\}$. The algorithm* `MultiBuild`$(f)$ *runs in time polynomial in n and d, and works as follows:*

Step 1. *We define $g := \phi(f)$ where ϕ is a random affine change of coordinates ($x_i \mapsto \sum_{j=1}^{n} \lambda_{ij} x_j + \lambda_i$ for all i).*

Step 2. *For each $j \in [\![1, n]\!]$, we set $g_j := \pi_j(g)$ where $\pi_j : \mathbb{F}[X] \longrightarrow \mathbb{F}[x]$ is induced by $x_k \mapsto 0$ if $k \neq j$ and $x_j \mapsto x$.*

Step 3. *Apply the algorithm* $\texttt{Build}(g_j)$ *from Theorem 4.1.4 to obtain the value* $s_j := \mathit{AffPow}(g_j)$ *and the triplets* $(\beta_{ij}, b_{ij}, e_{ij}) \in \mathbb{F} \times \mathbb{F} \times \mathbb{N}$ *such that* $g_j = \sum_{i=1}^{s_j} \beta_{ij}(x + b_{ij})^{e_{ij}}$.

Step 4. *We define* $P_j := \{(c_{ij}, 1/b_{ij}, e_{ij}) \,|\, c_{ij} := \beta_{ij} b_{ij}^{e_{ij}},\ 1 \leq i \leq s_i\}$.

Step 5. *We reorder the elements of* $P_2, \ldots, P_n$ *so that* $c_i := c_{i1} = c_{i2} = \cdots = c_{in}$ *and* $e_i := e_{i1} = e_{i2} = \cdots = e_{in}$ *for all* $i \in [\![1, s_1]\!]$.

Step 6. *If* $g = \sum_{i=1}^{s} c_i(1 + \sum_{j=1}^{n} x_j/b_{ij})^{e_i}$, *we output* $f = \phi^{-1}(g)$.

Or we reject if any of these steps is not feasible.
If the λ_i*'s and the* λ_{ij}*'s needed to define* ϕ *are chosen uniformly at random from a finite set* S*, then the probability of success of the algorithm is at least*

$$1 - \frac{d^{2/3}(2n + d)}{|S|}.$$

Proof. Since the input polynomial f satisfies the hypotheses of Proposition 5.5.1, we therefore have that $\mathrm{AffPow}_{\mathbb{F}}(f) = s$ and the optimal representation of f is essentially unique.
After applying a random ϕ as described in **Step 1**, with high probability[1] we have that ϕ is invertible and $g = \sum_{i=1}^{s} t_i^{e_i}$ with $t_i = \sum_{j=1}^{n} a_{ij}x_j + a_{i0}$ satisfies the following properties:

(i) $a_{ij} \neq 0$ for all i, j.

(ii) for all $j \neq 0$, then $a_{ij}/a_{i0} \neq a_{i'j}/a_{i'0}$ for all i, i', and

(iii) $a_{i0}^{e_i} \neq a_{i'0}^{e_{i'}}$ for all $i \neq i'$.

It is important to observe that for a generic choice of the λ_i's and λ_{ij}'s involved in the definition of ϕ, these conditions will be fulfilled, as this will guarantee in the probabilistic analysis that the polynomial encoding these conditions is nonzero. The goal of the algorithm is to recover f via the following expression of g:

$$g = \sum_{i=1}^{s} a_{i0}^{e_i} \left(1 + \sum_{j=1}^{n} \frac{a_{ij}}{a_{i0}} x_j\right)^{e_i};$$

so we are interested in computing the values

- $a_{i0}^{e_i}$ for all i
- a_{ij}/a_{i0} for all i, j
- e_i for all i

[1] A detailed probabilistic analysis is performed at the end of this proof.

In **Step 2**, for all $j \in [\![1, n]\!]$ we consider

$$\pi_j(g) = \sum_{i=1}^{s} a_{i0}^{e_i} \left(1 + \frac{a_{ij}}{a_{i0}}\, x\right)^{e_i} = \sum_{i=1}^{s} a_{ij}^{e_i} \left(x + \frac{a_{i0}}{a_{ij}}\right)^{e_i}.$$

Since $\pi_j(g)$ satisfies the hypotheses of Theorem 4.1.4 $\texttt{Build}(\pi_j(g))$ outputs the values

$$\left\{(a_{ij}^{e_i}, \frac{a_{i0}}{a_{ij}}, e_i) \,:\, 1 \le i \le s\right\}.$$

From these values we obtain in the sets

$$P_j = \left\{(a_{i0}^{e_i}, \frac{a_{ij}}{a_{i0}}, e_i) \,:\, 1 \le i \le s\right\}.$$

The uniqueness of the expression of g_j for all j and to (iii) guarantee that we recover g in **Step 6**.

We now give a probabilistic analysis of the algorithm. If we see the values of λ_i, λ_{ij} as variables, the invertibility of ϕ is equivalent to the nonvanishing of a degree n polynomial. Moreover, the a_{ij} are degree one polynomials in these variables. Thus, the conditions $a_{ij} \neq 0$ consist in the nonvanishing of $s(n+1)$ polynomials of degree 1. The conditions $a_{ij}/a_{i0} \neq a_{i'j}/a_{i'0}$ for all i, i', j with $j \neq 0$ can be seen as the nonvanishing of $s(s-1)n/2$ polynomials of degree 2. The conditions $a_{i0}^{e_i} \neq a_{i'0}^{e_{i'}}$ can be seen as the nonvanishing of $s(s-1)/2$ polynomials of degree at most $e := \max(e_i)$, which, by Corollary 2.2.10, is upper bounded by $d + (s^2/2)$. Hence, all the conditions to be satisfied can be codified in a nonzero polynomial ψ of degree

$$n + s(n+1) + s(s-1)n + (s(s-1)(2d+s^2)/4) \le \frac{8s^2n + 2s^2d + s^4}{4}.$$

Moreover, $e \le d + (s^2/2)$, and $s = n_e \le (3e/4)^{1/3}$; from where we deduce that $s \le d^{1/3}$ and the degree of ψ is upper bounded by $d^{2/3}(2n+d)$. The result follows from the Schwartz-Zippel lemma. □

6 Conclusion

In this final chapter, we give a brief overview of the results of this work and give several directions in which one could try to extend them.

Lower bounds. In Chapter 3, we exhibited two families of polynomials with large AffPow rank. More precisely, for both

$$f_n = \sum_{i=1}^{n} (x - a_i)^d \quad \text{and} \quad g_n = \prod_{i=1}^{n} (x - a_i)^{d/n},$$

with distinct a_i's, we proved in Propositions 3.2.7 and 3.2.10 that for some value of $c \in [0; 1]$, taking $n = c \cdot \sqrt{d}$ yields $\text{AffPow}_{\mathbb{F}}(f_n) = \Omega(\sqrt{d})$ and $\text{AffPow}_{\mathbb{F}}(g_n) = \Omega(\sqrt{d})$. As explained in Chapter 1, the goal is to obtain a linear lower bound, that is, an explicit polynomial f such that $\text{AffPow}_{\mathbb{F}}(f) = \Omega(d)$, with $d = \deg(f)$. For $\mathbb{F} = \mathbb{R}$, this problem is solved in [29] with one of the two families above: we have $\text{AffPow}_{\mathbb{R}}(f_{d/4}) = d/4$. However, for $\mathbb{F} = \mathbb{C}$ this problem is left open. Yet, we know that this family of polynomials f_n will not directly yield a linear lower bound as Remark 2.2.6 shows that Theorem 2.2.5 is optimal, that is, we know a choice of $n = \sqrt{d+1} + 1$ distinct a_i's such that the corresponding polynomial f_n is such that $\text{AffPow}(f_n) < n$. This does not completely eliminate the family f_n, but rather shows that in order to obtain a linear lower bound, one will have to choose precise values for the a_i's. For the other family of polynomials, we do not have any non-trivial upper bound for $n \geq 2$ and hence one could try to further investigate this family to determine its AffPow rank.

Linear independence. We have proposed Conjecture 3.3.3 concerning linear independence of shifted powers; we have proved in Proposition 3.3.4 its counterpart for the field of real numbers and have given some steps towards potential proofs over $\mathbb{C}$ by proving a weaker version in Proposition 3.3.8. Studying linear independence of shifted powers, apart from being interesting by itself, has nice consequences in

arithmetic complexity as it can imply a lower bound on the number of shifted powers needed to represent a polynomial of degree d. We believe that the conjecture is true; however, we have the feeling that a tool different from shifted differential equations should be used to prove them.

We have also provided bounds on the dimension of the vector space spanned by a family F of shifted powers that satisfy the Pólya condition. The lower bounds for the field of complex numbers given in Section 3.4.2 are nonconstructive in the sense that they do not pinpoint a linearly independent subset of F of cardinality equal to our lower bound on $\dim F$ (but they of course imply the existence of such a subset). It would be interesting to obtain a constructive proof. This may be related to the problem of obtaining a "good" sufficient condition for the linear independence of shifted powers over $\mathbb{C}$.

To our knowledge, the family F of real or complex polynomials that satisfies the Pólya condition and that spans a vector space with the least dimension is the one we provide in Lemma 3.4.5 It would be interesting to improve the bounds we provide or to show that they are tight.

Univariate algorithms. We designed univariate algorithms that, given a polynomial f in its dense form, find the optimal representation of f in Model 2. We achieved this goal in several cases, but we do not solve the problem in its full generality. The algorithm described in Theorem 4.1.4 whenever f admits an expression with distinct nodes and such that $n_{i+1} \leq (3i/4)^{1/3} - 1$ for all i, where n_i denotes the number of exponents $< i$ in the expression. As already pointed out, it is quite natural to assume that $n_i \leq i$ (Polya's condition) from the point of view of the optimality, but it would be interesting to relax the assumption $n_{i+1} \leq (3i/4)^{1/3} - 1$ in this theorem. When the nodes are not distinct, we provided algorithms for two special cases: when all the exponents with common node lie in a small interval (Section 4.2.1), and when two consecutive exponents with same node are far apart (Section 4.2.2). It would be very interesting to weaken these assumptions, or even to remove them entirely. All these algorithms only work in the regime of uniqueness of the optimal decomposition, and almost nothing is known when this is not the case.

Another issue that we have only begun to address is the analysis of the bit complexity of our algorithms. We give an explicit polynomial bound on the bit complexity of the algorithm of Theorem 4.1.2, but this issue seems to be more subtle for Theorem 4.1.4 due to the iterative nature of our algorithm. It is in fact not clear that there exists a solution of size polynomially bounded in the input size (i.e., in the bit size of f given as a sum of monomials). More precisely, we ask the following question.

Question 6.0.1. *We define the dense size of a polynomial* $f = \sum_{i=0}^{d} f_i x^i \in \mathbb{Z}[X]$ *as* $\sum_{i=0}^{d} [1 + \log_2(1 + |f_i|)]$. *Assume that* f *can be written as*

$$f = \sum_{i=1}^{s} \alpha_i (x - a_i)^{e_i}$$

with $a_i \in \mathbb{Z}$, $\alpha_i \in \mathbb{Z} \setminus \{0\}$, and that this decomposition satisfies the conditions of Theorem 4.1.4: the constants a_i are all distinct, and $n_{i+1} \leq (3i/4)^{1/3} - 1$.
Is it possible to bound the bit size of the constants α_i, a_i by a polynomial function of the dense size of f ?

As explained in Proposition 4.1.6, under the same conditions we have a decomposition algorithm that runs in time polynomial in the bit size of the *output*. It follows that the above question has a positive answer if and only if there is a decomposition algorithm that runs in time polynomial in the bit size of the input (i.e., in time polynomial in the dense size of f).
One could also ask similar questions in the case where the conditions of Theorem 4.1.4 do not hold. For instance, assuming that f has an optimal decomposition with integer coefficients, is there such a decomposition where the coefficients α_i, a_i are of size polynomial in the size of f ?

Multivariate algorithms. For the problem of multivariate reconstruction, we described two strategies in Chapter 5: univariate projections when AffPow(f) is small in terms of the degree, and Hessian methods when AffPow(f) is small in term of the number of variables. In Section 5.4, we provided an algorithm that can reconstruct decomposition in Model 1 with up to $\binom{n+1}{2}$ distinct affine forms. This algorithm relies on a random choice of the affine forms involved in the optimal expression and on the assumption that all the exponents are greater than 4. It would be very interesting to weaken these assumptions, or even to remove them entirely, even though the recent NP-hardness result of Waring decomposition [67] indicates that it might be hard to do so.
Even if we do not state it explicitly, whenever our algorithms succeed then we have some sort of uniqueness for all the affine forms involved with exponent ≥ 3. It would be interesting to characterize under which conditions the forms with exponent ≥ 3 in the optimal expressions are unique.
When f is a univariate polynomial of degree d, then AffPow$(f) \leq \lceil \frac{d+1}{2} \rceil$ and we the inequality is strict for a generic f. In the multivariate setting, it would be interesting to obtain upper bounds for AffPow(f) (different from those that can be directly derived from upper bounds for Waring(f) as in Proposition 1.2.1) and to determine the value(s) of AffPow(f) for generic polynomials.
We prove in Proposition 5.3.9 that whenever a bivariate polynomial $f(x_1, x_2)$ is a sum of two univariate ones $g_1(x_1), g_2(x_2)$, one can construct an optimal expression of f in Model 1 by gathering the (univariate) optimal expressions of f_1 and f_2 and putting together the terms of degree 1. We do not know if this phenomenon is also true for polynomials in more than two variables that can be written as a sum of univariates. Even more generally, we wonder if whenever $f(X) \in \mathbb{F}[X]$ can be expressed as a sum of two polynomials g_1, g_2 in disjoints set of variables, then an optimal expression for f in Model 1 can be built up from the optimal expressions of g_1 and g_2 by just putting together the terms of degree 1. This could be seen as an analog of Strassen's conjecture for the symmetric tensor rank, which can be stated as

follows: rank is additive on the sum of forms in different sets of variables (see [68]). Our result should be compared with [19, Theorem 5.6], where the authors prove the conjecture for homogeneous polynomials in four variables that can be written as a sum of two bivariate ones.

Intermediate models. As pointed out in Section 1.2, one could study simpler models as an intermediate step. Although there are already open problems for the Waring and Sparsest Shift models, it could be interesting to study intermediate models between these models and the affine power model. In Remark 1.2.2, we introduced the notions of *generalized Waring expression* and gave several interesting open questions that we did not investigate in this work. Similarly, one could define an intermediate model by placing an upper bound k on the number of distinct shifts. This would provide a smooth interpolation between the Sparsest Shift model (where $k = 1$) and Model 2, where $k = s$.

7 Bibliography

[1] Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. *In Foundations of Computer Science (FOCS)*, pages 67–75, 2008.

[2] James Alexander and André Hirschowitz. Polynomial interpolation in several variables. *Journal of Algebraic Geometry*, 4(2):201–222, 1995.

[3] Farid Alizadeh, Jean-Pierre A. Haeberly, and Michael L. Overton. Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results. *SIAM Journal on Optimization*, 8(3):746–768, 1998.

[4] Vladimir Arnold. *Lectures on Partial Differential Equations*. Springer, 2004.

[5] K Atkinson and A Sharma. A partial characterization of poised Hermite-Birkhoff interpolation problems. *SIAM Journal on Numerical Analysis*, 6(2):230–235, 1969.

[6] M. Ben-Or and P. Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proc. 20th annual ACM Symposium on Theory of Computing*, pages 301–309, 1988.

[7] A. Białynicki-Birula and A. Schinzel. Representations of multivariate polynomials by sums of univariate polynomials in linear forms. *Colloquium Mathematicum*, 112(2):201–233, 2008.

[8] Mats Boij, Enrico Carlini, and A Geramita. Monomials as sums of powers: the real binary case. *Proceedings of the American Mathematical Society*, 139(9):3039–3043, 2011.

[9] A. Borodin and P. Tiwari. On the decidability of sparse univariate polynomial interpolation. In *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*, STOC '90, pages 535–545, New York, NY, USA, 1990. ACM.

[10] Alin Bostan, Frédéric Chyzak, Marc Giusti, Romain Lebreton, Grégoire Lecerf, Bruno Salvy, and Eric Schost. *Algorithmes Efficaces en Calcul Formel*. published by the Authors, 2017. Open access book on `https://hal.archives-ouvertes.fr/AECF/`.

[11] Alin Bostan, Frédéric Chyzak, Grégoire Lecerf, Bruno Salvy, and Éric Schost. Differential equations for algebraic functions. In *Proceedings of the 2007 International Symposium on Symbolic and Algebraic Computation*, ISSAC '07, pages 25–32, New York, NY, USA, 2007. ACM.

[12] Alin Bostan and Philippe Dumas. Wronskians and linear independence. *The American Mathematical Monthly*, 117(8):722–727, 2010.

[13] Jerome Brachat, Pierre Comon, Bernard Mourrain, and Elias Tsigaridas. Symmetric tensor decomposition. *Linear Algebra and its Applications*, 433(11):1851 – 1872, 2010.

[14] Maria Chiara Brambilla and Giorgio Ottaviani. On the Alexander –Hirschowitz theorem. *Journal of Pure and Applied Algebra*, 212(5):1229–1251, 2008.

[15] M. Bôcher. The theory of linear dependence. *The Annals of Mathematics*, 2:81–96, 1900.

[16] P. Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*. Springer, 2000.

[17] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory*. Springer, 1997.

[18] Enrico Carlini. Reducing the number of variables of a polynomial. In *Algebraic geometry and geometric modeling*, Math. Vis., pages 237–247. Springer, Berlin, 2006.

[19] Enrico Carlini, Maria Virginia Catalisano, and Luca Chiantini. Progress on the symmetric Strassen conjecture. *J. Pure Appl. Algebra*, 219(8):3149–3157, 2015.

[20] Gonzalo Comas and Malena Seiguer. On the rank of a binary form. *Foundations of Computational Mathematics*, 11(1):65–78, 2011.

[21] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Undergraduate Texts in Mathematics. Springer, 1992.

[22] E. de Klerk. *Aspects of Semidefinite Programming*. Kluwer Academic Publishers, The Netherlands, 2002.

[23] K. Efremenko, A. Garg R. de Oliveira, and A. Wigderson. Barriers for rank methods in arithmetic complexity. *arXiv preprint arXiv:1710.09502 [cs.CC]*, 2017.

[24] Klim Efremenko, J M. Landsberg, Hal Schenck, and Jerzy Weyman. The method of shifted partial derivatives cannot separate the permanent from the determinant. *Mathematics of Computation*, 87(312):2037–2045, 2018.

[25] S.M. Engdahl and A.E. Parker. Peano on wronskians: A translation. http://www.maa.org/publications/periodicals/convergence/peano-on-wronskians-a-translation-introduction.

[26] I. Fischer. *Mathematics Magazine*, volume 67, chapter Sums of like powers of multivariate linear forms, pages 59–61. Taylor & Francis, Ltd. on behalf of the Mathematical Association of America, 1994.

[27] Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower bounds for depth 4 formulas computing iterated matrix multiplication. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 128–135. ACM, 2014.

[28] Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinievasan. Lower bounds for depth 4 formulas computing iterated matrix multiplication. *SIAM J. Comput*, 44(5):1173–1201, 2015.

[29] Ignacio García-Marco and Pascal Koiran. Lower bounds by Birkhoff interpolation. *Journal of Complexity*, 39, 07 2015.

[30] Ignacio García-Marco, Pascal Koiran, and Timothée Pecatte. Reconstruction algorithms for sums of affine powers. *arXiv preprint arXiv:1607.05420*, 2016. Conference version in: *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC '17, pages 317–324, 2017.

[31] Ignacio García-Marco, Pascal Koiran, and Timothée Pecatte. On the linear independence of shifted powers. *Journal of Complexity*, 45:67–82, 04 2018.

[32] Ignacio García-Marco, Pascal Koiran, and Timothée Pecatte. Polynomial equivalence problems for sums of affine powers. *accepted at ISSAC' 18*, 2018.

[33] Mark Giesbrecht and Daniel S. Roche. Interpolation of shifted-lacunary polynomials. *Comput. Complex.*, 19(3):333–354, September 2010.

[34] John Hilton Grace and Alfred Young. *The algebra of invariants*. Cambridge Library Collection. Cambridge University Press, Cambridge, 2010. Reprint of the 1903 original.

[35] Dima Grigoriev and Marek Karpinski. A zero-test and an interpolation algorithm for the shifted sparse polynomials. In Gérard Cohen, Teo Mora, and Oscar Moreno, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, pages 162–169, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[36] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the chasm at depth four. In *Proceedings of the Conference on Computational Complexity (CCC)*, 2013.

[37] Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[38] Anthony Iarrobino and Vassil Kanev. *Power Sums, Gorenstein Algebras, and Determinantal Loci*. Springer, 1999.

[39] Joachim Jelisiejew. An upper bound for the Waring rank of a form. *Archiv der Mathematik*, 102(4):329–336, Apr 2014.

[40] E. Kaltofen and B. Trager. Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *Journal of Symbolic Computation*, 9(3):301–320, 1990.

[41] N. Kayal. An exponential lower bound for the sum of powers of bounded degree polynomials. *Electronic Colloquium on Computational Complexity (ECCC)*, 2012.

[42] N. Kayal, P. Koiran, T. Pecatte, and C. Saha. Lower bounds for sums of powers of low degree univariates. In *Proc. 42nd International Colloquium on Automata, Languages and Programming (ICALP 2015), part I*, LNCS 9134, pages 810–821. Springer, 2015. Available from `http://perso.ens-lyon.fr/pascal.koiran`.

[43] Neeraj Kayal. Efficient algorithms for some special cases of the polynomial equivalence problem. In *Symposium on Discrete Algorithms (SODA)*, pages 1409–1421. Society for Industrial and Applied Mathematics, January 2011.

[44] Neeraj Kayal. Affine projections of polynomials. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pages 643–662. ACM, 2012.

[45] Neeraj Kayal and Chandan Saha. Lower bounds for depth three arithmetic circuits with small bottom fanin. In *Proceedings of the 30th Conference on Computational Complexity*, pages 158–182, 2015.

[46] Neeraj Kayal, Chandan Saha, and Ramprasad Saptharishi. A super-polynomial lower bound for regular arithmetic formulas. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 146–153, 2014. STOC '14.

[47] Neeraj Kayal and Ramprasad Saptharishi. *A Selection of Lower Bounds for Arithmetic Circuits*, pages 77–115. Springer International Publishing, Cham, 2014.

[48] J. Kleppe. Representing a homogeneous polynomial as a sum of powers of linear forms. Thesis for the degree of Candidatus Scientarum (University of Oslo), 1999. Available at `http://folk.uio.no/johannkl/kleppe-master.pdf`.

[49] Donald E. Knuth. *The Art of Computer Programming, Volume 4, Fascicle 4: Generating All Trees–History of Combinatorial Generation (Art of Computer Programming)*. Addison-Wesley Professional, 2006.

[50] P. Koiran. Arithmetic circuits: the chasm at depth four gets wider. *Theoretical Computer Science*, 448:56–65, 2012. arxiv.org/abs/1006.4700.

[51] Pascal Koiran. Shallow circuits with high-powered inputs. *CoRR*, abs/1004.4960, 2010. Innovations in Computer Science.

[52] Pascal Koiran, Natacha Portier, and Sébastien Tavenas. A Wronskian approach to the real τ-conjecture. *Journal of Symbolic Computation*, 68(2):195–214, 05 2015.

[53] Mrinal Kumar and Shubhangi Saraf. The limits of depth reduction for arithmetic formulas: It's all about the top fan-in. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 136–145. ACM, 2014.

[54] Y. N. Lakshman and B. David Saunders. Sparse shifts for univariate polynomials. *Applicable Algebra in Engineering, Communication and Computing*, 7(5):351–364, 1996.

[55] Joseph M Landsberg and Zach Teitler. On the ranks and border ranks of symmetric tensors. *Foundations of Computational Mathematics*, 10(3):339–366, 2010.

[56] H.W.. Lenstra, A.K. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[57] G. Lorentz and S. Riemenschneider. Probabilistic approach to Schoenberg's problem in Birkhoff interpolation. *Acta Mathematica Hungarica*, 33(1-2):127–135, 1979.

[58] George G Lorentz, Kurt Jetter, and Sherman D Riemenschneider. *Birkhoff interpolation*, volume 19 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1984.

[59] G. Labahn M. Giesbrecht and W.-S. Lee. Symbolic-numeric sparse interpolation of multivariate polynomials. *Journal of Symbolic Computation*, 44(8):943–959, 2009.

[60] W. Lee M. Giesbrecht, E. Kaltofen. Algorithms for computing sparsest shifts of polynomials in power, chebyshev and pochhammer bases. *Journal of Symbolic Computation*, 36(3-4):401–424, 2003.

[61] N. Nisan and A. Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1996. Conference version in FOCS'95.

[62] Luke Oeding and Giorgio Ottaviani. Eigenvectors of tensors and algorithms for waring decomposition. *Journal of Symbolic Computation*, 54:9 – 35, 2013.

[63] G. Polya and G. Szego. *Problems and Theorems in Analysis*, volume II. Springer, 1976.

[64] Berkowitz S. On computing the determinant in small parallel time using a small number of processors. *Information Processing Letters*, 18:147–150, 1984.

[65] Alexander Schrijver. *Theory of linear and integer programming*. Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons, Ltd., Chichester, 1986. A Wiley-Interscience Publication.

[66] J. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4):701–717, 1980.

[67] Yaroslav Shitov. How hard is the tensor rank? *arXiv preprint arXiv:1611.01559*, 2016.

[68] Volker Strassen. Vermeidung von Divisionen. *J. Reine Angew. Math.*, 264:184–202, 1973.

[69] Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Mathematical Foundations of Computer Science (MFCS)*, pages 813–824, 2013.

[70] L. G. Valiant. Completeness classes in algebra. In *Proceedings of the 11th Annual STOC*, pages 249–261, 1979.

[71] L. G. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff. Fast parallel computation of polynomials using few processors. *SIAM*, 12(4):641—644, 1983.

[72] M Voorhoeve and A.J Van Der Poorten. Wronskian determinants and the zeros of certain functions. *Indagationes Mathematicae (Proceedings)*, 78(5):417 – 424, 1975.

[73] R. Zippel. Probabilistic algorithms for sparse polynomials. *Symbolic and Algebraic Computation*, pages 216–226, 1979.