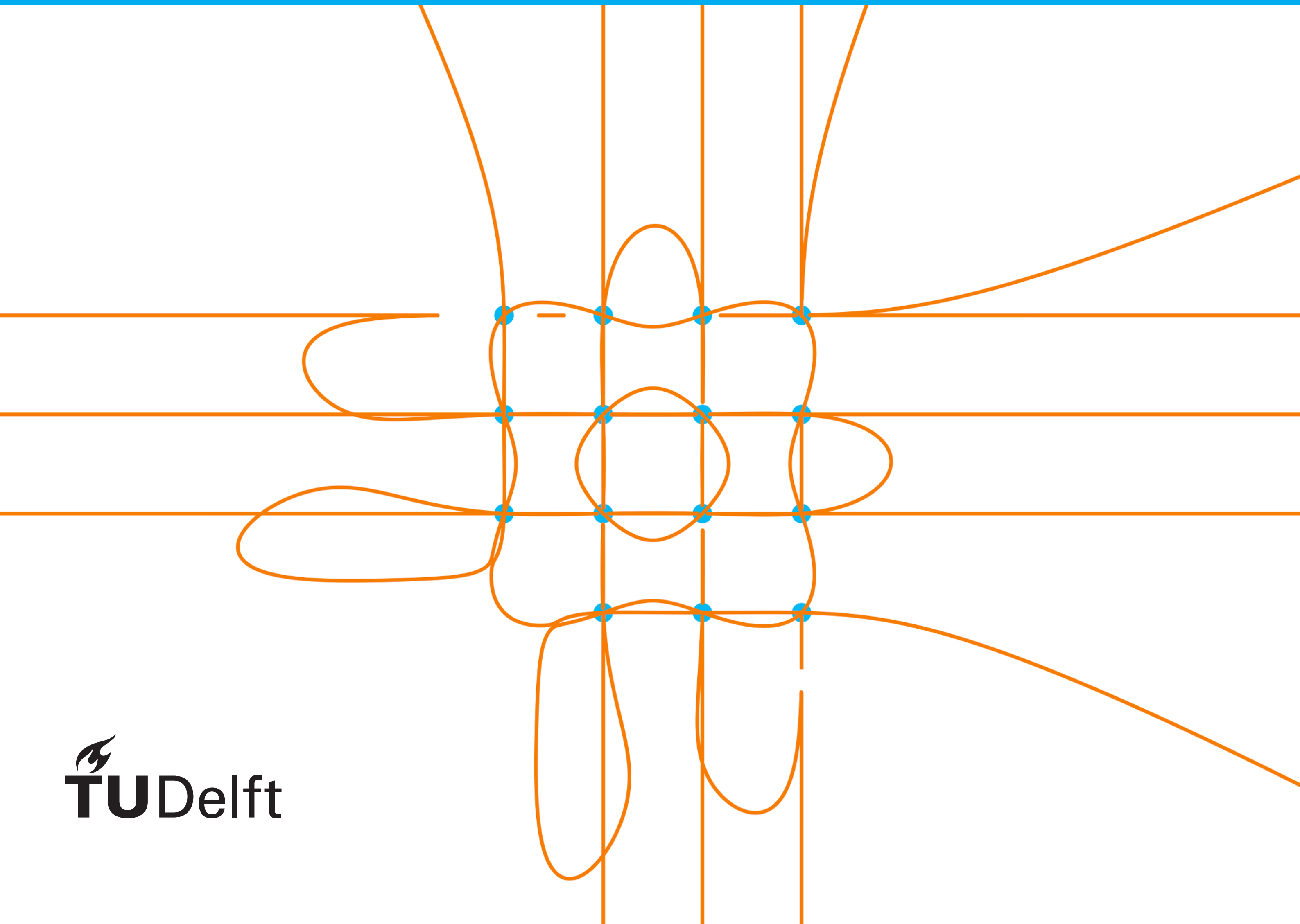
Polynomial Methods for Grid Covering Problems

Why curves beat lines

Lander Verlinde
Msc. Applied Mathematics



TU Delft

Polynomial Methods for Grid Covering Problems

Why curves beat lines

by

Lander Verlinde

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday June 21, 2024 at 2:00 PM.

Student number:	5109353	
Project duration:	December 11, 2023 – June 21, 2024	
Thesis committee:	Prof. Dr. D.C. Gijswijt,	TU Delft, Chair
	Dr. C.E. Groenland	TU Delft, Daily Supervisor
	Dr. A. Bishnoi	TU Delft, Daily Supervisor
	Prof. Dr. J.M.A.M van Neerven,	TU Delft

TU Delft

Abstract

Consider an arbitrary finite grid in some field $\mathbb{F}$. How many hyperplanes are required so that every point is contained in at least k hyperplanes, except for one point that is not allowed to be contained in any hyperplane? This problem is called the *hyperplane grid covering problem* and has a rich history, throughout which it has been studied for multiple types of grids. In many of these settings, the polynomial method has proven to be extremely useful in finding bounds on the optimal hyperplane covering number. This has given rise to a second problem: the *polynomial grid covering problem*. This problem considers the minimum degree of a polynomial such that every grid point is a root with multiplicity k, except for one point where the polynomial does not vanish.

This thesis provides a thorough investigation of these two related problems. The first chapter derives and analyses the most important polynomial bounds for grid coverings: the Alon-Füredi Bound and the Ball-Serra Bound. Furthermore, we explore the link between grid coverings and two results from algebraic geometry: the Footprint Bound and the Cayley-Bacharach Theorems. The second chapter continues to focus on the Ball-Serra Bound. We apply the established theoretical framework to examine the difference between hyperplane covers and polynomial covers for different grids and compare both covers to the bound. Firstly, we consider the hypercube and show that the Ball-Serra Bound is never tight. Secondly, we look at the binary field. In a new result, we prove its polynomial covering number for multiplicity four. We also look at the distinction between polynomials and hyperplanes for a relaxed version of the covering problem. Lastly, we investigate covers of grids in the Cartesian plane. In this setting, a polynomial corresponds to a curve, and a hyperplane corresponds to a line. Since curve covers in the Cartesian plane have not been studied before, we provide algorithms and techniques to explore these covers. All proposed methods and findings suggest that curves provide more efficient grid coverings than lines.

Acknowledgement

This thesis marks the ending of my academic journey at the TU Delft, a journey that would not have been possible without the support and guidance of many individuals.
First and foremost, I am immensely grateful to my committee members Anurag, Carla, Jan and Dion for supporting me throughout this project. I am so lucky to have a graduation committee filled with four people whom I regard as mentors.
My two amazing daily supervisors, Anurag and Carla, did not only provide me with mathematical advice. They have always pushed me academically and personally, encouraging me to go to conferences and seminars, and gave me a lot of advice when choosing where to do a PhD. Jan has been there for me from the very beginning of the bachelor as mentor of the Excellence Programme and has always provided me with guidance on every step of the way, whether it was about summer schools, going on exchange, or choosing a PhD. And last but not least, Dion was appointed as the mentor of my first year bachelor instruction group. Ever since, his door was always open and I have always felt very welcome to discuss anything with him.
I also would like to express my gratitude towards Alessandro Neri. He recommended me to look into the Cayley-Bacharach Theorems and symmetric grids in the plane. Both recommendations became important topics of this thesis. In addition, I want to extend my thanks to Relinde Jurrius. Her guidance during my internship has made it a great experience and was a major factor in my decision to do a PhD after my studies. And, of course, I would like to thank all friends and family that have been there for me. From Den Helder to Brussels and from Delft to Lausanne, I have met so many amazing people in the past five years. If you are taking the time to read this, you are probably part of this group, so a very heartfelt thank you to you.

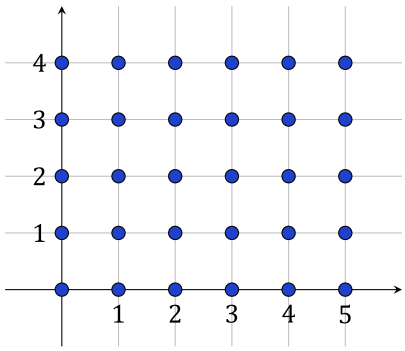
Lander Verlinde
Delft, June 2024

Contents

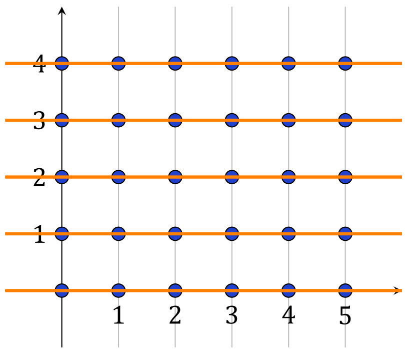
1

Introduction

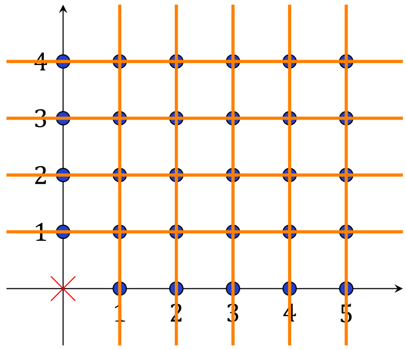
This thesis considers the broad topic of grid covering problems. Mathematical problems have a reputation as being very difficult to understand, let alone to solve. But at the core of grid covering problems lies a very easy question, which we will solve to embark on our journey in this topic. Consider the (6×5)-grid, obtained by taking $\{0, 1, 2, 3, 4, 5\} \times \{0, 1, 2, 3, 4\}$, as drawn below. How many lines do we minimally need such that every point is covered at least once by a line?



It does not take long to see that this can be done using 5 horizontal lines, while it is impossible to use fewer lines.



The question becomes a little more challenging when we add the constraint that the origin has to remain uncovered. That is, there cannot be a line that passes through the origin. An intuitive way to cover the grid now is to include every non-zero horizontal and vertical line once. It turns out that this is optimal.



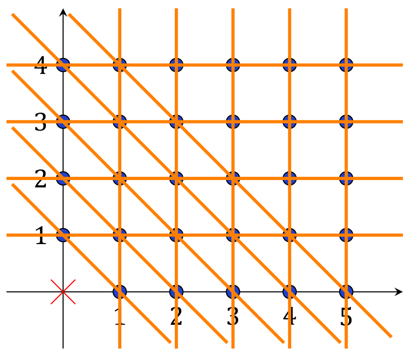
But how do we know that this is optimal and that we could not have covered the grid with fewer lines, while still avoiding the origin? This is where matters become more complicated, but also beautiful and perhaps surprising. To prove the optimality of the cover, we will not use geometrical arguments. Instead, we translate our cover to a problem involving polynomials. Each line in our cover can be seen as a first-order polynomial. For instance, the vertical lines are of the form $x = i$ for $i \in [5]$ and the horizontal lines are of the form $y = j$ for $j \in [4]$, where we let $[n]$ denote the set $\{1, 2, \ldots, n\}$. With these polynomials, finding a minimal line cover of the grid corresponds to finding a polynomial of minimal degree that can be factorised into polynomials of degree 1, such that every non-zero point in the grid is a root of the polynomial and the origin is not. The polynomial in our cover is

$$f(x, y) = \prod_{i\in[5]}(x - i) \cdot \prod_{j\in[4]}(y - j),$$

and $\deg f = 9$, equal to the number of lines in the cover. The reason why we formulated the cover as a polynomial is that there exist bounds on the degree of polynomials that vanish on grids. One of these bounds is the Alon-Füredi Bound [2], stating that any polynomial f that vanishes once on all points of some grid $S_1 \times S_2 \times \cdots \times S_n$, except one point where it is non-zero, has degree at least $\deg f \geq \sum_i(|S_i| - 1)$. In this case, $S_1 = \{0, 1, 2, 3, 4, 5\}$ and $S_2 = \{0, 1, 2, 3, 4\}$, so we could not have used fewer lines for the cover. This small example shows that changing one condition in the problem that we are considering can have a considerable impact on the number of lines we require. What happens if we require that every point is contained in at least two lines, rather than in just one? Then we are looking for a so-called *hyperplane* 2*-cover* of the grid. In general, we define *hyperplane k-covers* as follows.

Definition. *Let n, k be natural numbers and consider an arbitrary grid $\Gamma = S_1 \times S_2 \times \cdots \times S_n$ such that $\overline{0} \in \Gamma$. A hyperplane k-cover of Γ is a set of hyperplanes such that every non-zero point of Γ is contained in at least k hyperplanes, while there is no hyperplane that goes through the origin. The minimum size of such a cover is called the hyperplane k-covering number of Γ.*

Returning to the example, most points of the grids are already covered twice when we reuse the previous cover. We only need to add five more lines to find a hyperplane 2-cover, or since in this case a hyperplane corresponds to a line, a line 2-cover.



As before, to verify that there does not exist a line 2-cover of the grid with fewer lines, we turn to polynomials. This time, our cover corresponds to the polynomial

$$f(x,y) = \prod_{i\in[5]}(x-i)\cdot\prod_{j\in[4]}(y-j)\cdot\prod_{k\in[5]}(x+y-k).$$

The degree of f is again equal to the number of lines. While the Alon-Füredi Bound still holds, the lower bound can be increased by making use of the fact that the polynomial now has to vanish with multiplicity 2 at every non-zero point. Thus, for covers with higher multiplicity we will use a different lower bound, namely the Ball-Serra Bound [4]. This bound states that if f vanishes at all points of an $(n\times m)$-grid, $n \geq m$, with multiplicity $k \geq 2$, except at one point where it is non-zero, f has degree at least $k(n-1)+(m-1)$. We can always assume that $n \geq m$ because of symmetry. For $k=2$, this shows that our line cover is optimal. Note that the Ball-Serra Bound does not assume anything on the form of the polynomial. So, the next question that arises is whether there is a difference in the minimal degree of polynomials that vanish with the right multiplicity on the grid and that are a hyperplane cover – and hence can be factorised into polynomials of degree 1 – and the minimal degree of polynomials that do vanish with the right multiplicity, but are not a hyperplane cover. The latter type of cover is called a *polynomial 2-cover*.

Definition. *Let n, k be natural numbers and consider an arbitrary grid $\Gamma = S_1 \times S_2 \times \cdots \times S_n$ such that $\overline{0} \in \Gamma$. A polynomial k-cover of Γ is a polynomial f such that every non-zero point of Γ is a root of f with multiplicity at least k, while $f(\overline{0}) \neq 0$. The minimum degree of such a cover is called the polynomial k-covering number of Γ.*

Note that a hyperplane k-cover always gives rise to a polynomial k-cover and hence the polynomial k-covering number of Γ is at most its hyperplane k-covering number. For grids in the plane, it turns out that there can be a difference in the optimal degree of both covers once we consider covers with multiplicity $k \geq 3$.

The main goal of this thesis is to provide a starting point and a framework to attack different flavours of grid covering problems and highlight some of those problems that I found most interesting. While the first chapter is more oriented towards literature, the second one combines the existing literature with some of my own, new results together with remarks, thoughts and conjectures. To make it more clear which results were already known and which are my own, new results are marked with stars ❇ around them.

Chapter 2 first fully revolves around the polynomial method. We start by rigorously introducing the Alon-Füredi Bound. Instead of proving the bound directly, we shall derive it from a different bound: the Footprint Bound. This bound makes use of *Gröbner Bases*: a particular set of multivariate polynomials that generates an ideal in a polynomial ring, and that provides a simplified way to solve systems of equations [20]. So not only does the Footprint Bound elegantly prove the Alon-Füredi Bound, it also shows the link between grid coverings and different notions regarding polynomials that are less used in combinatorics. On the subject of lesser-known theorems on polynomial methods in combinatorics, the Alon-Füredi Bound is actually a direct result from an older theorem: one of the Cayley-Bacharach Theorems. This set of theorems has its roots in the 19th century, but it can even be argued that its most simple Cayley-Bacharach Theorem is a geometrical result that dates back to the 4th century [13]. In Section 2.3, we travel through time to explore all the different results and investigate a modern Cayley-Bacharach Theorem that immediately implies the Alon-Füredi Bound. We conclude the chapter by looking at the polynomial method for covers with higher multiplicity. As said, for such covers there exists a different bound, namely the Ball-Serra Bound. The proof of this theorem uses yet again different polynomial arguments, like Alon's Combinatorial Nullstellensatz. Hence, the first chapter gathers and highlights a variety of polynomial methods, all with applications in grid covering problems.

Chapter 3 takes a closer look at the Ball-Serra Bound. For many grids and multiplicities, it is actually not known whether the bound is tight. Moreover, there might be a difference in tightness for polynomial covers and hyperplane covers. These covers are investigated for three types of grids. First of all, there is the hypercube. Clifton and Huang [11] have shown that in this setting the Ball-Serra bound is only tight for hyperplane covers with multiplicity 1 and 2 by showing a better lower bound for covers with higher multiplicities. Sauermann and Wigderson [19] later improved this lower bound for multiplicity greater than or equal to three and have shown that their bound is tight for polynomial covers. For hyperplane covers however, there is a gap between this best possible bound obtained by the polynomial method and the conjectured value of the hyperplane covering number. Thus, if the conjectured value is true, there is a clear distinction between the lowest degree of a polynomial cover and the smallest size of a hyperplane cover. After the study of the hypercube we consider covers of the binary field. This setting requires some more set-up to investigate the multiplicity of a root of a polynomial. In that regard, we introduce *Hasse derivatives*. Using these derivatives, we determine the polynomial 4-covering number of the binary field.

❊ **Theorem** ❊ *The polynomial 4-covering number of* $\mathbb{F}_2^n$ *is equal to* $n + 4$.

This result implies that the Ball-Serra Bound is not tight for polynomial 4-covers. Generalising this result unfortunately turns out to be difficult and determining the hyperplane covering number forms a substantial challenge too. But when allowing the origin to be covered strictly less than k times in a k-cover, some results can be obtained. Bishoi et al. [9] have shown that in this setting there is indeed a difference between polynomial covers and hyperplane covers. In the final part of the second chapter, covers of grids in the Cartesian plane are investigated, just like the grid we have studied above. For line covers, the threshold when the Ball-Serra Bound is tight for k-covers of an $(n \times m)$-grid with $n \geq m$ is known [8], namely when $n \geq (k-1)(m-1)+1$. Note that for $k = 2$, this is always true, showing that it is no coincidence that the cover we found above is tight with the Ball-Serra Bound. After having discussed how to come up with this threshold, we shift our focus to polynomial k-covers of these grids. These covers have not been studied before, so we have to come up with our own techniques to investigate how they behave. At first, we come up with an algorithm to generate polynomial 3-covers of small grids. This algorithm is based on partial derivatives that have to vanish at the grid points. We try to generalise these polynomial 3-covers by looking at the *slices* of a grid. These do not yet provide a proof on when the Ball-Serra Bound is tight for these covers, but provide a lot of support for the conjectured threshold. After the review on 3-covers we also look at polynomial covers with higher multiplicity. Again, there is quite some evidence that points towards a certain threshold, namely when $n \geq m + (k-2)$.

❊ **Conjecture** ❊ *Let* n, m, k *be integers such that* $m \leq n - (k-2)$ *and* $k \leq n$. *Consider two arbitrary sets* $S_1, S_2 \subseteq \mathbb{R}_{\geq 0}$ *such that* $S_1 = \{0, s_1, \dots, s_{n-1}\}$ *and* $S_2 = \{0, t_1, \dots, t_{m-1}\}$. *Let* $\Gamma = S_1 \times S_2$. *Then there exists a polynomial* f *of degree* $k(n-1) + (m-1)$ *that covers every non-zero point of* Γ k *times while avoiding the origin.*

In addition to analysing the conjectured threshold, we also consider the behaviour of the polynomial k-covering number for grids that do not satisfy the threshold. For these polynomials we conjecture an upper bound on the covering number. In all cases, there seems to be a very clear difference in behaviour between line k-covers and polynomial k-covers. The conjectured threshold for tightness for the polynomial k-cover grows considerably slower than the threshold for the curve k-cover. But further research on this threshold is needed to rigorously prove its behaviour.

2

The Polynomial Method for Grid Covers

In many combinatorial and optimisation methods where we minimise an objective function it is easier to find an upper bound on the optimal value rather than a lower bound. That is often also the case for grid coverings. To find an upper bound on the minimal cover size, we can simply provide a cover and its size is immediately an upper bound. Proving that there does not exist a smaller cover requires some more work. One of the methods to show that we cannot do better is the so-called *polynomial method*. As the name suggests, this method translates the problem into a problem involving polynomials. Depending on whether we want to cover all non-zero points in the grid once or multiple times, two lower bounds can be derived with the polynomial method. These are respectively the *Alon-Füredi Bound* and the *Ball-Serra Bound*. This section analyses both bounds. First, we look at the Alon-Füredi bound. For the hypercube, this bound can be easily derived, but other settings are more complex. We prove the bound using a different tool that involves roots of polynomials: the *Footprint Bound*. Secondly, we show the link between the Alon-Füredi bound and the *Cayley-Bacharach Theorem*. This is a theorem with a rich history, of which the first version was already formulated in the 19th century, but notions of its statement were already made in the 4th century. Lastly, we look into the Ball-Serra Bound, for covers with multiplicities. Its proof is directly linked to Alon's Combinatorial Nullstellensatz and Hilbert's Weak Nullstellensatz.

2.1. The Alon-Füredi Bound

To introduce the Alon-Füredi Bound, it makes sense to look at hyperplane coverings of the hypercube. Specifically, we would like to know how many hyperplanes are minimally required to cover $Q^n = \{0,1\}^n$ while leaving the origin uncovered, for $n \in \mathbb{N}$. As a possible cover, we could take the following n hyperplanes: $\{\overline{x} : x_i = 1\}$ for $i \in [n]$. This clearly provides a cover that is of the right form, but how do we ensure that there is no construction that requires less hyperplanes? The argument was given by Alon and Füredi in [2]. They make use of a polynomial constructed from a cover.

Theorem 1 (Alon-Füredi for the Hypercube)**.** *Suppose that we have m hyperplanes $H_1, \ldots, H_m$ in $\mathbb{R}^n$ that avoid the origin and cover all other $2^n - 1$ vertices of the hypercube Q^n. Then $m \geq n$.*

Proof. We write $H_i = \{\overline{x} : \overline{a_i} \cdot \overline{x} = b_i\}$ for $\overline{a_i}, \overline{x} \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ for every $i \in [m]$. We construct the multivariate polynomial

$$p(\overline{x}) = \prod_{i \in [m]} (\overline{a_i} \cdot \overline{x} - b_i).$$

Out of $p(\overline{x})$ we obtain a new, multilinear polynomial $f(\overline{x})$ by replacing every occurrence of x_i^d with $d \geq 2$ by x_i. Since $p(\overline{x}) = 0$ for all $\overline{x} \in \{0,1\}^n \setminus \{\overline{0}\}$, and $x_i^d = x_i$ for all $x_i \in \{0,1\}$, we find that $f(\overline{x}) = 0$ for all $\overline{x} \in \{0,1\}^n \setminus \{\overline{0}\}$ too. Similarly $f(\overline{0}) \neq 0$. Now we claim that

$$f(\overline{x}) = c(x_1 - 1)(x_2 - 1) \cdots (x_n - 1),$$

where $c = f(\overline{0})$ for n is even and $c = -f(\overline{0})$ for n is odd.[1] To see that f can be written in this form, let

$$f(\overline{x}) = \sum_{I \subseteq [n]} c_I x_I,$$

where $x_I = \prod_{i \in I} x_i$. Then we use induction on the size of I to prove that

$$c_I = (-1)^{n-|I|} c.$$

For the base case, if $I = \emptyset$, then $C_\emptyset = f(\overline{0}) = (-1)^n c$. Now suppose we know that for all $J \subsetneq I$, $c_J = (-1)^{n-|J|}$. Then we consider the indicator vector 1_I and plug it into f:

$$\begin{aligned}
0 &= f(1_I) \\
&= \sum_{J \subseteq I} c_J && \text{(by def of } c_I\text{)} \\
&= \sum_{J \subsetneq I} c_J + c_I \\
&= \sum_{J \subsetneq I} (-1)^{n-|J|} c + c_I && \text{(induction hypothesis)} \\
&= c \left(\sum_{j=0}^{|I|-1} \binom{|I|}{j} (-1)^{n-j} \right) + c_I \\
&= c \cdot (-1)^n \left(\sum_{j=0}^{|I|-1} \binom{|I|}{j} (-1)^{j} \right) + c_I \\
&= c \cdot (-1)^{n+|I|-1} + c_I && \left(\sum_{j=0}^{|I|-1} \binom{|I|}{j} (-1)^j + \binom{|I|}{|I|} (-1)^{|I|} = 0 \right) \\
&= c \cdot (-1)^{n-|I|-1} + c_I,
\end{aligned}$$

and hence, indeed, $c_I = c \cdot (-1)^{n-|I|}$. To see that $f(\overline{x}) = c(x_1 - 1)(x_2 - 1) \dots (x_n - 1)$, it suffices to consider the coefficient of an arbitrary monomial on the right-hand side. Any x_I has coefficient $c \cdot (-1)^{n-|I|}$, which is exactly what we wanted. Thus, indeed $f(\overline{x}) = c(x_1 - 1)(x_2 - 1) \dots (x_n - 1)$ and in particular $\deg(f) = n$. Therefore, $n = \deg(f) \le \deg(p) = m$, which is what we wanted to prove. □

This shows that it can be useful to use polynomials instead of staying in the original, geometrical setting with hyperplanes. In the same paper [2], Alon and Füredi also prove a stronger theorem that implies the one above, but its proof is more complicated and does not rely anymore on the setting of the hypercube. In fact, they provide a bound on the number of points in a grid that a polynomial misses when it does not cover the entire grid.

Theorem 2 (Alon-Füredi Theorem). *Let $S_1, \dots, S_n$ be subsets of an arbitrary field $\mathbb{F}$ and let f be a polynomial of degree d that does not vanish on the entire grid $S_1 \times \dots \times S_n$. Then f does not vanish on at least $\min\{\prod_i c_i : 1 \le c_i \le |S_i| \;\; \forall i, \quad \sum_i c_i \ge \left(\sum_i |S_i|\right) - d\}$ points of the grid.*

In our case, we consider a grid where we want to avoid exactly one point, for example the origin. That is, $\min\{\prod_i c_i : 1 \le c_i \le |S_i| \;\; \forall i, \quad \sum_i c_i \ge \left(\sum_i |S_i|\right) - d\} = 1$. Hence, this theorem implies that if we want to avoid vanishing at at least one point, each c_i can be at most 1, and thus $d \ge \sum_i (|S_i| - 1)$.

Corollary 3 (The Alon-Füredi Bound). *Let $S_1, \dots, S_n$ be subsets of an arbitrary field $\mathbb{F}$. Consider a polynomial p such that p vanishes on $S_1 \times S_2 \times \dots \times S_n$ except at one point. Then $\deg(p) \ge \sum_i (|S_i| - 1)$.*

Again, this bound is tight: suppose $(z_1, \dots, z_n) \in \mathbb{F}$ is the point that we want to avoid. Then the collection of hyperplanes $\{\overline{x} : x_i = s \;\; \forall i \in [n], \forall s \in S_i \setminus \{z_i\}\}$ is a cover of size $\sum_i (|S_i| - 1)$. Moreover, if we take $S_i = \{0, 1\}$ for all $i \in [n]$, we recover Theorem 1. Rather than presenting the original proof of Theorem 2, we will use the Footprint Bound. This is a tool that is able to estimate the number of common zeroes of a set of polynomials.

[1] Note that this definition of c fixes a small typo in the original proof.

2.2. The Footprint Bound

To understand the Footprint Bound, we first dive into the world of *Gröbner Bases*. A Gröbner Basis is a set of multivariate polynomials that generalises Gaussian Elimination and the Division Algorithm for polynomials. But we are mainly interested in these bases because they enable us to find a bound on the number of common zeroes of polynomials. This will let us prove the Alon-Füredi Theorem. The following section bundles the ideas from a couple of sources and unifies their notation to create an overview of the strengths of Gröbner Bases. For an introduction to Gröbner Bases, Strumfels [20] provides a great starting point. Some of the more involved ideas and the intuition behind the Footprint Bound can also be found in [12] and the link with Alon-Füredi is well explained in [7]. First, we need to set up some notation.

Definition 4. *Let K be any field. Then we denote the ring of polynomials in n variables $x_1, \dots, x_n$ with coefficients in K by $K[x_1, \dots, x_n]$. Moreover, if $\mathcal{F}$ is any set of polynomials, then the ideal generated by $\mathcal{F}$ is $\langle \mathcal{F} \rangle$, i.e.*

$$\langle \mathcal{F} \rangle = \{a_1 f_1 + \cdots + a_\ell f_\ell : f_1, \dots, f_\ell \in \mathcal{F} \text{ and } a_1, \dots, a_\ell \in K[x_1, \dots, x_n]\}.$$

An arbitrary ideal will be denoted by I.

To be able to define a Gröbner Basis, we need to define an order on the monomials. To this end, let x^a denote $x^{a_1} \cdots x^{a_n}$.

Definition 5. *A monomial order on $K[x_1, \dots, x_n]$ is a total order $\prec$ on the set of monomials $\mathcal{M}$ with the following properties:*

1. *$x^a \prec x^b \Rightarrow x^{a+c} \prec x^{b+c}$ for all $a, b, c \in \mathbb{N}^n$.*
2. *Any nonempty subset $A \subseteq \mathcal{M}$ has a smallest element.*

Example 6. *There are multiple ways of ordering monomials. One classical example is the lexicographic order. For this order, we say $x^a \prec x^b$ if the first nonzero entry of $b - a$ is positive. Hence, $x_1^1 x_2^5 x_3^7 \prec x_1^3 x_2^1 x_3^1$ as the difference of the powers is $(2, -4, -6)$. For a bivariate polynomial of degree two, we have*

$$1 \prec x_2 \prec x_2^2 \prec x_1 \prec x_1 x_2 \prec x_1^2.$$

Example 7. *A second example is the graded lexicographic order. In this order we order say $x^\alpha \prec x^\beta$ if $\sum_i \alpha_i \leq \sum_i \beta_i$. In case of equality we use the lexicographic order described above. So for this order, we in fact have $x_1^3 x_2^1 x_3^1 \prec x_1^1 x_2^5 x_3^7$. For a bivariate polynomial of degree two, we have*

$$1 \prec x_2 \prec x_1 \prec x_2^2 \prec x_1 x_2 \prec x_1^2.$$

Observation 8. *For every monomial order, 1 is the smallest element.*

Proof. Consider an arbitrary monomial order on $K[x_1, \dots, x_n]$ and suppose there is an $a \in \mathbb{N}^n$ such that $x^a \prec 1 = x^0$, where 0 denotes the all-zeroes vector. Then, by property one in Definition 5, $x^{a+c} \prec x^c$ for all $c \in \mathbb{N}^n$. Then we consider the subset of monomials $A = \{x^{a+i} : i \in \mathbb{N}^n\}$. By the second property in the definition, A has a smallest element, say x^{a+j}, for some $j \in \mathbb{N}^n$. But then $x^{a+a+j} \prec x^{a+j}$. This forms a contradiction. □

Once we fix a certain monomial order, then every polynomial f in $K[x_1, \dots, x_n]$ has a unique leading monomial $LM(f)$. This leading monomial is the $\prec$-largest monomial that occurs with non-zero coefficient in the expansion of f. For a set of polynomials $\mathcal{F}$, we say that $LM(\mathcal{F}) = \{LM(f) : f \in \mathcal{F}\}$ is the set of leading monomials. Using this set for polynomials in an ideal, we can construct a new ideal.

Definition 9. *Consider an arbitrary ideal I of $K[x_1, \dots, x_n]$. Then the leading ideal $\langle LM(I) \rangle$ is the ideal generated by the leading monomials of all polynomials in I:*

$$\langle LM(I) \rangle = \langle LM(f) : f \in I \rangle.$$

Now we are finally able to define a Gröbner Basis. In short, this is a basis of polynomials generated by the leading monomial ideal for a given order.

Definition 10. *A finite subset of polynomials $\mathcal{G}$ of an ideal I is a Gröbner Basis with respect to a monomial order on the monomials of I if the leading monomials of the polynomials $g \in \mathcal{G}$ generate the leading monomial ideal:*

$$\langle LM(I)\rangle = \langle LM(g) : g \in \mathcal{G}\rangle.$$

For every ideal, we know that there exists a finite Gröbner basis because of Hilbert's Basis Theorem (see e.g. Chapter 2 in [12] for a detailed explanation).

Theorem 11 (Hilbert's Basis Theorem)**.** *Every ideal I of $K[x_1, \dots, x_n]$ is generated by a finite set. In other words, $I = \langle \mathcal{F}\rangle$ for some finite set of polynomials $\mathcal{F}$.*

Moreover, note that the definition of a Gröbner Basis does not impose any restrictions on the size of such a basis. If $\mathcal{G}$ is a Gröbner Basis for I, then any finite subset of I that contains $\mathcal{G}$ is also a Gröbner Basis for I. Hence, it makes sense to define a *reduced Gröbner Basis* that has some additional useful properties.

Definition 12. *Consider a Gröbner Basis $\mathcal{G}$. We say that $\mathcal{G}$ is a reduced Gröbner Basis if*

1. *For every $g \in \mathcal{G}$, the coefficient of the leading monomial of g is 1.*
2. *For every $g \in \mathcal{G}$, no monomial of g lies in $\langle LM(\mathcal{G} \setminus \{g\})\rangle$.*

So now that we have established what a (reduced) Gröbner Basis is, we are able to look into how we can use such a Basis to estimate the number of common zeroes of a set of polynomials. We will call such a set of common zeroes of polynomials a *variety*.

Definition 13. *The variety V of a subset $\mathcal{F}$ of $K[x_1, \dots, x_n]$ is the set of all common zeroes of polynomials in $\mathcal{F}$:*

$$V(\mathcal{F}) = \{(z_1, \dots, z_n) \in K^n : f(z_1, \dots, z_n) = 0 \quad \forall f \in \mathcal{F}\}.$$

A first link between the variety of an ideal and its reduced Gröbner Basis is given by *Hilbert's Weak Nullstellensatz*. Although Hilbert proved his Nullstellensatz already in 1893 [15], we will consider a slightly more accessible and modern version [12].

Theorem 14 (Hilbert's Weak Nullstellensatz)**.** *Let K be an algebraically closed field and I an ideal of $K[x_1, \dots, x_n]$ satisfying $V(I) = \emptyset$. Then $I = K[x_1, \dots, x_n]$.*

We know that the variety does not change if we change the polynomials that generate an ideal $\langle \mathcal{F}\rangle$ of $K[x_1, \dots, x_n]$. That is because if $\mathcal{F}$ and $\mathcal{G}$ both generate some ideal I,

$$V(\mathcal{F}) = V(\langle \mathcal{F}\rangle) = V(\langle \mathcal{G}\rangle) = V(\mathcal{G}).$$

The reason why we might prefer the reduced Gröbner Basis over another generating set of polynomials of an ideal is because of its nice link with the Weak Nullstellensatz. Suppose that we have a set of polynomials $\mathcal{F}$ and that we would like to know whether these polynomials have a common zero. That is, whether $V(\mathcal{F}) \neq \emptyset$. Since $V(\mathcal{F}) = V(\langle \mathcal{F}\rangle)$, we know by the Weak Nullstellensatz that $V(F)$ is empty if and only if $1 \in \langle \mathcal{F}\rangle$. Since we can choose any set of polynomials that generate the same ideal when considering the ideal, this actually implies that $V(\mathcal{F}) = \emptyset$ if and only if the reduced Gröbner basis $\mathcal{G}$ for $\langle \mathcal{F}\rangle$ equals $\{1\}$. To see why this holds, we show that $\{1\}$ is the only reduced Gröbner Basis of the ideal $\langle 1\rangle = K[x_1, \dots, x_n]$. Suppose $\{g_1, \dots, g_m\}$ is a reduced Gröbner Basis of $I = \langle 1\rangle$. Then $1 \in \langle LM(I)\rangle = \langle LM(g_1), \dots, LM(g_m)\rangle$, which means that one of the leading terms divides 1, say $LM(g_i)$. Since 1 in the smallest element in the monomial order, this specific $LM(g_i)$ should actually be equal to 1. Since g_i is included in the reduced Gröbner Basis, all other leading terms should be equal to 1 too, which means that the Gröbner Basis is indeed $\{1\}$.
So, now we have a clear way of verifying whether the number of zeroes in a variety is equal to 0 or not. Could we also say something about the number of such zeroes when the variety is non-empty? It turns out that we can say something about this number using the notion of *standard monomials*. These are the monomials that are not in the leading monomial ideal.

Definition 15. *For an arbitrary ideal I of $K[x_1, \dots, x_n]$ we say that a monomial is a standard monomial if it is not contained in $\langle LM(I)\rangle$. The set of standard monomials is denoted by $\Delta(I)$.*

This set is also sometimes called the *deltaset* of the ideal I, or its *footprint*, a term coined by Blahut in 1991 [16]. Consider an arbitrary ideal I, generated by polynomials $g_1, \dots, g_n$. We let $\Delta(g_1, \dots, g_n) = \{x^u : x^u \notin \langle LM(g_i) \rangle \ \forall i\}$. Then $\Delta(I) \subseteq \Delta(g_1, \dots, g_n)$. It follows from the definition that we have equality if $g_1, \dots, g_n$ forms a Gröbner Basis. We can also note that the number of elements in the footprint is sometimes infinite. Suppose, for example, that I is in an ideal in $K[x_1, x_2, x_3]$ and that $\langle LM(I) \rangle = \langle x_1^2, x_2^4, x_1 x_3^2 \rangle$. Then $|\Delta(I)|$ is infinite, as every monomial of the form x_3^α for $\alpha \geq 0$ is a standard monomial. Now suppose that $\langle LM(I) \rangle = \langle x_1^3, x_2^4, x_3^3 \rangle$, then the standard monomials are of the form $x_1^i x_2^j x_3^k$ with $0 \leq i \leq 2$, $0 \leq j \leq 3$ and $0 \leq k \leq 2$, so $|\Delta(I)| = 16$. This argument can be extended to prove a characterisation of the cases where the size of the footprint is finite.

Lemma 16. *Let I be an ideal of $K[x_1, \dots, x_n]$. Then $|\Delta(I)|$ is finite if and only if every x_i appears to some power in $\langle LM(I) \rangle$ for $i \in [n]$.*

In the case where the footprint of an ideal is finite, its size can actually say something about the size of its variety. Indeed, the size of the footprint upper bounds the size of the variety. This is known as the Footprint Bound.

Theorem 17 (The Footprint Bound)**.** *If $|\Delta(I)|$ is finite, then $|V(I)| \leq |\Delta(I)|$.*

To prove this inequality, we need another lemma.

Lemma 18. *Let I be an ideal in $K[x_1, \dots, x_n]$. Then $\Delta(I)$ is a basis for the vector space $K[x_1, \dots, x_n]/I$.*

Proof. Let $\mathcal{G}$ be a Gröbner Basis for I with respect to the same monomial order used to find the footprint $\Delta(I)$ and let $f \in K[x_1, \dots, x_n]$. Dividing f by $\mathcal{G}$ yields a remainder r_f of the form

$$r_f = \sum_{i=1}^{t} a_i M_i,$$

where $a_i \in K[x_1, \dots, x_n]$ and $M_i \in \Delta(I)$ for all i. Since the remainder r_f of any polynomial f is of this form, we find that $\Delta(I)$ generates $K[x_1, \dots, x_n]/I$. We still have to prove that all $M_i \in \Delta(I)$ are linearly independent modulo I. So assume that $\sum b_i M_i \in I$ equals 0 for some non-zero $b_i \in K[x_1, \dots, x_n]$. But this means that there is an element in I whose leading term is in $\Delta(I)$. This is a contradiction. Thus, $b_i = 0$ for all i and $\Delta(I)$ is indeed a basis for the quotient space. □

Now we can prove the actual Footprint Bound.

Proof of Theorem 17. Let $z_1, \dots, z_m$ be the distinct elements of $V(I)$. By the previous lemma, it suffices to find a linearly independent set in $K[x_1, \dots, x_n]/I$ of size m. To find these points, we will use the fact that given points $p_1, \dots, p_t$, there exists a polynomial f_1 such that

$$f_1(p_1) = 1 \text{ and } f_1(p_2) = \cdots = f_1(p_t) = 0.$$

To see why this is true, note that if $p_2 \neq p_1 \in K[x_1, \dots, x_n]$, these points have to differ in at least one coordinate, say the ℓ-th one. Then

$$g_2(x_1, \dots, x_n) = \frac{x_\ell - p_{2_\ell}}{p_{1_\ell} - p_{2_\ell}}$$

satisfies $g_2(p_1) = 1$ and $g_2(p_2) = 0$. This can of course be repeated for every $p_j \neq p_1$ to obtain $g_3, \dots, g_t$. Hence,

$$f_1 = g_2 g_3 \cdots g_t$$

is the polynomial that we are looking for. So, for $z_1, \dots, z_m$, we can find $f_1, \dots, f_m$ such that $f_i(z_j) = \delta_{ij}$ for all $i, j \in [m]$. Now suppose that for these f_i, we have $\sum_{i=1}^m a_i f_i \in I$ equals 0 for some $a_1, \dots, a_m \in K[x_1, \dots, x_n]$. Then we also have $\sum_{i=1}^m a_i p_i(z_j) = 0$, so $a_j = 0$ for all $j \in [m]$. So indeed, $\{f_1, \dots, f_m\}$ is a linearly independent set in $K[x_1, \dots, x_n]/I$ of size m. □

Example 19. *Let us work out an example to see how we can use the Footprint Bound to get an estimate on the size of a variety. Consider the ideal* $I_1 = \langle x^2y^3 - y, x^3y - x, x^4 - y^3, y^4 - xy^2\rangle$ *of* $\mathbb{C}[x,y]$*. Then using the graded lexicographic order yields* $\langle LM(I_1)\rangle = \langle x^2y^3, x^3y, x^4, y^4\rangle$*. To find the size of its footprint, we draw these leading monomials in Figure 2.1. One can see that there are 12 standard monomials in this case and thus, the above polynomials have at most 12 common zeroes. In fact, there are only 2 roots. There are also cases where the bound is better. Consider, for example, the ideal* $I_2 = \langle x - y^7, y^{12} - y^2\rangle$ *of* $\mathbb{C}[x,y]$*, where we use the lexicographic order. Then* $\langle LM(I)\rangle = \langle x, y^{12}\rangle$*, which shows that there are 12 standard monomials of the form* y^a *for* $0 \le a \le 11$*. So again,* $|\Delta(I)| = 12$*, while* $|V(I_2)| = |\{(y^7, y) : y^{10} - 1 = 0\} \cup (0,0)| = 11$.

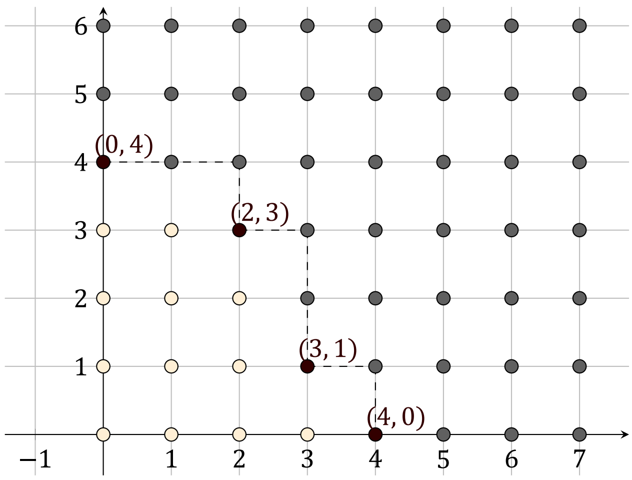


Figure 2.1: Depiction of the leading monomials of I_1. All grey monomials can be obtained using linear combinations of the leading monomials, the beige monomials are standard monomials.

As already mentioned, the footprint bound can actually be used to prove the Alon-Füredi Theorem. Let us first restate the theorem:

Theorem 2. *Let* f *be a polynomial of degree* d *that does not vanish on the entire grid* $S_1 \times \cdots \times S_n$*. Then* f *does not vanish on at least* $\min\{\prod_i c_i : 1 \le c_i \le |S_i|\ \forall i,\quad \sum_i c_i \ge \left(\sum_i |S_i|\right) - d\}$ *points of the grid.*

Proof of Theorem 2. Let $f \in K[x_1, \dots, x_n]$ be a polynomial that does not vanish on all points of a finite grid $S_1 \times \cdots \times S_n$. We let $g_i = \prod_{s \in S_i}(x_i - s)$ and say $|S_i| = \sigma_i$. Then we are interested in $V(\{f, g_1, \dots, g_n\}) = V(\langle f, g_1, \dots, g_n\rangle)$.
We order the monomials of f using the graded lexicographic ordering. We first reduce f to some polynomial f^* such that $LM(f^*)$ is of the form $\prod_i x_i^{u_i}$, where $u_i \le \sigma_i - 1$ for all i. Suppose that there is a term τ in f of the form

$$h(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \cdot x_i^{u_i},$$

for some $(n-1)$-variate polynomial h and where $u_i \ge \sigma_i$ for some i. Consider

$$f' = f - h(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \cdot x^{u_i - \sigma_i} \cdot \prod_{s \in S_i}(x_i - s).$$

Since for any vanishing point $(a_1, \dots, a_n)$ of f on the grid we have $g_i(a_i) = 0$, f' vanishes at all the vanishing points of f on the grid. Moreover, f' no longer has the term τ, instead it has a term where the degree of x_i equals $u_i - 1$. So, if we keep repeating this process, we obtain a polynomial f^* where any term is indeed of the proposed form, which still vanishes at all the grid points where f vanishes. Now say $\deg(f^*) = d^*$, then we have $LM(f^*) = x^u = \prod_i x_i^{u_i}$ such that $\sum_i u_i = d^*$. Furthermore, for all i, we have $LM(g_i) = x_i^{\sigma_i}$. All standard monomials in $\Delta(x^u, x_1^{\sigma_1}, \dots, x_n^{\sigma_n})$ are of the form $\prod_i x_i^{v_i}$, where for at least one i we have $v_i < u_i$. There are $\prod_i \sigma_i$ reduced monomials in f^* (possibly with zero coefficient) and $\prod_i(\sigma_i - u_i)$ of them are multiples of x^u. Hence, there are $\prod_i \sigma_i - \prod_i(\sigma_i - u_i)$ standard monomials. We let $c_i = \sigma_i - u_i$, so that $1 \le c_i \le \sigma_i$ for all i and

$$d^* = \sum_i u_i = \sum_i \sigma_i - \sum_i c_i \Rightarrow \sum_i c_i = \sum_i \sigma_i - d^*.$$

By the Footprint Bound, $|V(\langle f^*, g_1, \dots, g_n \rangle)| \leq \prod_i \sigma_i - \prod_i c_i$. Hence, f^* does not vanish on at least $\prod_i c_i$ points of the grid, which implies that f^* does not vanish on at least $\min\{\prod_i c_i : 1 \leq c_i \leq \sigma_i \ \ \forall i, \quad \sum_i c_i = \sum_i \sigma_i - d^*\}$ points of the grid, so f does not either. If we let $\deg(f) = d$, we know $d \geq d^*$, thus f does not vanish on at least $\min\{\prod_i c_i : 1 \leq c_i \leq |S_i| \ \ \forall i, \quad \sum_i c_i \geq \left(\sum_i |S_i|\right) - d\}$. □

2.3. Historical Intermezzo: The Cayley-Bacharach Theorems

So far, the polynomial method has given us the number of hyperplanes required to cover all points of a grid except one, given by the Alon-Füredi bound. But actually, the bound is also a direct consequence of an older theorem: the Cayley-Bacharach Theorem. This theorem has been reformulated in many forms, such that Cayley-Bacharach has become an umbrella term for multiple theorems involving the size of intersections of curves. For a more extensive analysis of the different Cayley-Bacharach Theorems that exist, [13] is a great starting point. We will specifically trace the versions through history that led up to the Cayley-Bacharach Theorem that implies Alon-Füredi, starting in the 18th century with small excursions to even earlier times.
We first consider another theorem involving the variety of a set of polynomials that we need to introduce the Cayley-Bacharach Theorems. Bézout's Theorem gives an upper bound on the size of a variety of a set of polynomials based on their degrees [5].

Theorem 20 (Bézout's Theorem, 1779)**.** *Let K be an algebraically closed field. Let p_1 and p_2 be polynomials of respective degrees d_1 and d_2 in $K[x_1, x_2]$ such that they do not share a common component. Then $|V(\{p_1, p_2\})| \leq d_1 d_2$. We have equality if the zeroes are counted with multiplicity.*

Using Bézout's Theorem, we can derive the theorem that commonly goes under the name Cayley-Bacharach, but which is actually due to Chasles [10].

Theorem 21 (Chasles' Theorem, 1865)**.** *Let C_1 and C_2 be two cubic curves that intersect over some algebraically closed field K in precisely 9 distinct points $P_1, \dots, P_9 \in K^2$. Then any other cubic curve that passes through any 8 of the 9 points must pass through the ninth point too.*

In his blog post [21], Terry Tao has given an elegant proof of the above theorem, based on a text of Husemöller [17].

Proof. Consider two arbitrary cubic curves C_1 and C_2 that intersect in precisely nine distinct points. Let $C_1 \cap C_2 = \{P_1, \dots, P_9\}$ and let $f_1 = 0$ and $f_2 = 0$ be the equations of the curves C_1 and C_2. We want to show that if $f_3 = 0$ is the equation of a third cubic curve and $f_3(P_1) = \dots = f_3(P_8) = 0$, then $f_3(P_9) = 0$. To do so, it is sufficient to show that f_3 is a linear combination of f_1 and f_2, i.e. there are constants a_1 and a_2 such that $f_3 = a_1 f_1 + a_2 f_2$. In that case $f_3(P_9) = a_1 f_1(P_9) + a_2 f_2(P_9) = 0$.
So, let us assume that f_3 is linearly independent of f_1 and f_2. We will use Bézout's Theorem to show that the points $\{P_1, \dots, P_9\}$ have a nice structure that we will exploit. First of all, no 4 points of $\{P_1, \dots, P_9\}$ can lie on some line λ, because otherwise $|V(\{C_1, \lambda\})| = 4 > 3 \cdot 1$. Since K is algebraically closed, this would be a contradiction on Bézout's Theorem unless λ is a component of C_1. But with the exact same argument, λ then also has to be a component of C_2. Since C_1 and C_2 intersect in only 9 points, this is impossible.
Secondly, we have that any 5 points of $\{P_1, \dots, P_9\}$ define a unique conic. Suppose Q_1 and Q_2 are two conics that pass through the same 5 points of $\{P_1, \dots, P_9\}$. Again, by Bézout's Theorem Q_1 and Q_2 have to share a common component. That is, $Q_1 = Q_2$, or they share a common line. Suppose the latter is true. Since this line cannot pass through 4 points P_i, there has to be a line shared by Q_1 and Q_2 that passes through exactly three of these points. The other two points define the second line in the conic and therefore $Q_1 = Q_2$.
Thirdly, we can even prove that when only considering the first eight points $\{P_1, \dots, P_8\}$, no 3 points are collinear. Suppose without loss of generality that the first three points P_1, P_2, P_3 lie on a common line μ. The remaining five points lie on a unique conic Q'. Let P' be another point that lies on μ and R' be a point that does not lie on μ nor on Q'. We can pick constants b'_1, b'_2, b'_3 such that $f' := b'_1 f_1 + b'_2 f_2 + b'_3 f_3$ vanishes on P' and R'. This can be done by just solving the above equation, having evaluated the equation in P' and R'. Then $F'(P_i) = 0$ for $i = 1, \dots, 8$. Therefore, f' vanishes on 4 collinear points P_1, P_2, P_3, P'. Hence, Bézout's Theorem shows that this line μ has to be a component of the cubic F'. The other component of f' is a conic that passes through the five points $P_4, \dots, P_8$. So, this conic has to

be equal to Q'. But then R' does not lie on μ nor Q', even though it is a vanishing point of f'. This is a contradiction. This contradiction already proves the theorem for certain choices of C_1 and C_2, like, for instance, the grid structure. In such a structure we clearly have 3 collinear points amongst $\{P_1, \ldots, P_8\}$ and thus f_3 cannot be linearly independent from f_1 and f_2.
Fourthly, in case we have not encountered a contradiction yet, no conic can go through 6 points of $\{P_1, \ldots, P_8\}$. Note that Bézout's Theorem shows that no conic can go through 7 of the points $\{P_1, \ldots, P_8\}$, otherwise C_1 and C_2 share a component. Hence, suppose that a conic Q^* goes through exactly six points, say $P_1, \ldots, P_6$. Then there is a line ν going through the remaining two points P_7 and P_8. Let P^* be another point on Q^* and R^* be another point that does not vanish on ν nor on Q^*. Again, we can find a non-trivial cubic $f^* = b_1^* f_1 + b_2^* f_2 + b_3^* f_3$ that vanishes on P^* and R^*. As f^* vanishes on seven points of Q^*, it has to consist of Q^* and a line that passes through P_7 and P_8, namely ν. But then f^* does not pass through R^*. Contradiction.
Lastly, let κ be the line through P_1 and P_2 and Q° be the conic through $P_3, \ldots, P_7$. By the above results, $P_8 \notin \kappa \cup Q^\circ$. Choose P_1° and P_2°, both on κ but neither on Q°. Pick constants $b_1^\circ, b_2^\circ, b_3^\circ$ such that $f^\circ = b_1^\circ f_1 + b_2^\circ f_2 + b_3^\circ f_3$ vanishes on P_1° and P_2°. Since f° meets κ in four points, Bézout's Theorem implies that it contains κ as a component, together with a conic. This conic passes through $P_3, \ldots, P_7$ and thus, is equal to Q°. But then f° does not pass through P_8. A final contradiction that shows that f° cannot exist and that we could not have chosen f_1, f_2, f_3 to be linearly independent. □

Even though the above theorem may seem like a pretty specific setting for which we require two cubics that intersect in exactly nine distinct points, it can actually be used to prove a lot of incidence relations between lines, conics and curves. For instance, there is the classical Pappus' Theorem that dates back from the 4th century A.D. and is accredited to Pappus of Alexandria [13].

Theorem 22 (Pappus' Theorem, 4th century)**.** *Let λ and μ be two distinct lines and let $A_1, A_2, A_3, B_1, B_2, B_3$ be distinct points such that $A_1, A_2, A_3 \in \lambda$, $B_1, B_2, B_3 \in \mu$ and no point lies on both lines. Suppose that for $ij = 12, 23, 31$, the lines A_iB_j and A_jB_i intersect in the point C_{ij}. Then C_{12}, C_{23}, C_{31} are collinear.*

Proof. A sketch of the above situation is given in Figure 2.2. Let Q_1 be the union of lines A_1B_2, A_2B_3 and A_3B_1 (the dotted lines in the figure). Similarly, let Q_2 be the union of lines A_1B_3, A_2B_1 and A_3B_2 (the dashed lines). We let Q denote the union of lines A_1A_3, B_1B_3 and $C_{12}C_{23}$. Note that Q_1 and Q_2 are two cubic curves that meet in exactly nine distinct points $\{A_1, A_2, A_3, B_1, B_2, B_3, C_{12}, C_{23}, C_{31}\}$. But Q is also a cubic curve that goes through the first eight of these points. Hence, C should also go through C_{23}. Since C_{23} cannot lie on λ nor μ, the result follows. □

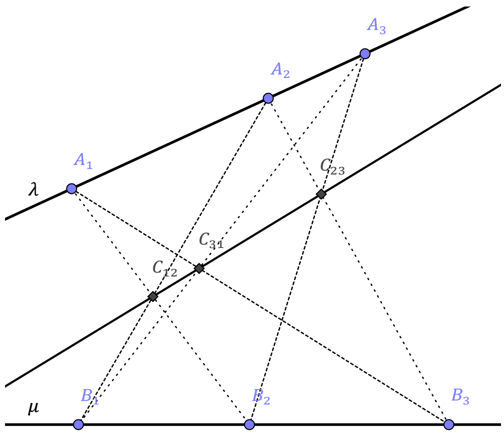


Figure 2.2: Representation of Pappus' Theorem is depicted.

Fast forward in time to the 17th century when Pascal gave a generalisation of Pappus' Theorem in his essay *Essai pour les coniques*. The essay has been lost to time, but the theorem luckily still remains [14].

Theorem 23 (Pascal's Theorem, 1640)**.** *Let $A_1, A_2, A_3, B_2, B_2, B_3$ be distinct points on a conic C. Suppose that for $ij = 12, 23, 31$, the lines A_iB_j and A_jB_i meet at point C_{ij}. Then the points C_{12}, C_{23}, C_{31} are collinear.*

Proof. The proof of Pascal's Theorem is actually the exact same proof as for Pappus' Theorem, where we replace the conic $\lambda \cup \mu$ by the conic C. Since C meets every line in at most two points, none of the C_{ij} can lie on C. See also Figure 2.3 for a drawing. □

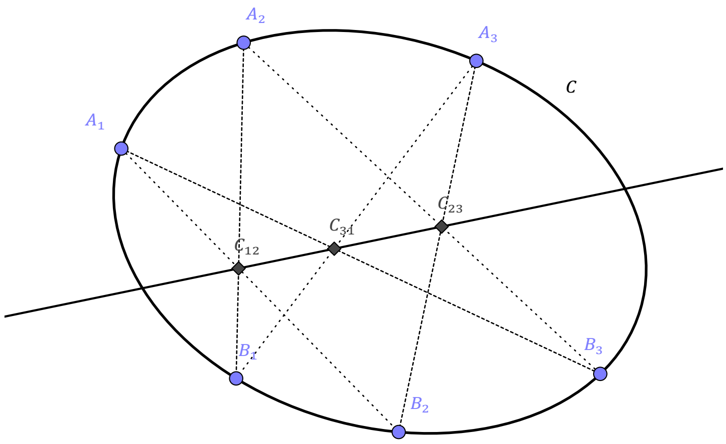


Figure 2.3: Representation of Pascal's Theorem is depicted.

So indeed, when specifically considering two conics that vanish on nine distinct points, we are able to prove some pretty and historical results from geometry. But the next Cayley-Bacharach Theorem is even more powerful than Chasles' Theorem. For this statement, we need some more notions. When we require that a curve in $K[x_1, x_2]$ of degree d vanishes at a point $p \in K$, we impose a linear constraint on the coefficients defining the curve. For example, suppose we consider a curve f in $K[x_1, x_2]$ of degree 1. Then this curve has equation $f(x, y) = ax + by + c$, where a, b, c are the coefficients. If we require that $f(2, -1) = 0$, then we impose the linear condition $2a - b + c = 0$. This idea of points imposing linear constraints allows us to define when a set of points *fail to impose* linear constraints.

Definition 24. *If a set Γ of γ points imposes only $\alpha \leq \gamma$ independent linear conditions on the coefficients of a curve of degree d, then we say that Γ fails to impose $\gamma - \alpha$ independent linear conditions on curves of degree $\leq d$.*

For instance, 9 collinear points fail to impose 5 independent conditions on curves of degree 3: by Bézout's Theorem, any cubic that passes through 4 of the collinear points must pass through them all. In general it holds that any set of k collinear points fails to impose $k - (d + 1)$ conditions on curves of degree $d \leq k - 1$.
Secondly, when returning to Pappus' Theorem, there is actually a specific case that we have not considered yet, but that gives a nice introduction to projective geometry. We might have chosen A_1, A_2, B_1 and B_2 such that A_1B_2 and A_2B_1 are actually parallel. In the Cartesian plane that means that A_1B_2 and A_2B_1 do not have an intersection point and therefore we cannot use Pappus' Theorem. In the projective plane $\mathbb{P}^2$ however, we say that C_{12} lies on the line at infinity. In the case where A_2B_1 and A_3B_2 also intersect on the line at infinity, Pappus' Theorem implies that also C_{31} lies on this line at infinity. Translated back, this means that A_1B_3 and A_3B_1 are parallel too. This example shows it can make sense to look at the projective plane $\mathbb{P}^2$ instead of the Cartesian one. This is what we shall do in the next few results to then revert back to the Cartesian plane.
Lastly, it is important to note that if we equip the set of all curves in $\mathbb{P}^2$ of degree k that vanish on some set of points Γ with the usual polynomial addition and the usual scalar multiplication, this set forms a vector space. And if $\Gamma' \subseteq \Gamma$, the vector space V of degree k curves vanishing on Γ forms a subspace of the vector space V' of degree k curves vanishing on Γ'. Therefore, we can also consider the quotient

space $V'\backslash V$, also called V' modulo V. Now we are able to consider the first Cayley-Bacharach Theorem [13].

Theorem 25 (Cayley-Bacharach 1, 19th century)**.** *Suppose that two plane curves $C_1, C_2 \subseteq \mathbb{P}^2$ of respective degrees d_1 and d_2 intersect in $d_1 d_2$ points $\Gamma = C_1 \cap C_2 = \{p_1, \dots, p_{d_1 d_2}\}$. Partition $\Gamma = \Gamma' \cup \Gamma''$ and set $s = d_1 + d_2 - 3$. If $k \leq s$ is a non-negative integer, then the dimension of the vector space of curves of degree k vanishing on Γ', modulo those vanishing on all of Γ, is equal to the failure of Γ'' to impose linearly independent conditions on curves of degree $s - k$.*

As a corollary, we formulate a generalisation of Chasles' Theorem.

Corollary 26. *Let $C_1, C_2 \subseteq \mathbb{P}^2$ be plane curves of respective degrees d_1 and d_2, meeting in $d_1 d_2$ distinct points Γ. If C is a plane curve of degree $\leq d_1 + d_2 - 3$ containing all but one point of Γ, then C contains all of Γ.*

Proof. We apply the first Cayley-Bacharach Theorem. Hence, let Γ' be the subset of Γ containing all but one point where C vanishes and let Γ'' be the singleton with the remaining point. Moreover let k denote $\deg C \leq s = d_1 + d_2 - 3$. Then the failure of Γ'' to impose linearly independent conditions on curves of degree $s - k \geq 0$ is equal to 0 as $|\Gamma''| = 1$. Hence, the dimension of the vector space of curves of degree k vanishing on Γ' modulo those vanishing on all of Γ is equal to 0. This means that any such curve vanishes on all of Γ. □

There exists a generalisation of the first Cayley-Bacharach Theorem that replaces the two curves by n hypersurfaces of $\mathbb{P}^n$, the projective space of dimension n. A hypersurface is a manifold of dimension $n - 1$ in an ambient space of dimension n. When $n = 2$, then a hypersurface is a plane curve, which brings us back to the setting of the theorem above. In a higher dimension n, we can regard a hypersurface as a polynomial in n variables [13].

Theorem 27 (Cayley-Bacharach 2, 19th century)**.** *Let $X_1, \dots, X_n$ be hypersurfaces in $\mathbb{P}^n$ of respective degrees $d_1, \dots, d_n$, meeting transversely, and suppose that its finite intersection $\Gamma = X_1 \cap \dots \cap X_n$ is partitioned in $\Gamma = \Gamma' \cup \Gamma''$. Set $s = \sum_i d_i - n - 1$. If $k \leq s$ is a non-negative integer, then the dimension of the family of curves of degree k containing Γ', modulo those containing all of Γ, is equal to the failure of Γ'' to impose independent conditions of curves of degree $s - k$.*

In the above theorem, we require the hypersurfaces to meet transversely. Meeting transversely can be seen as the opposite as meeting tangentially. So, at an intersection point, we require the tangents of the curves to be different. For a final statement of the Cayley-Bacharach Theorem, we can weaken that assumption to the hyperplanes intersecting in *singular points*, i.e. $X_1 \cap \dots \cap X_n$ is zero-dimensional. Moreover, we no longer restrict ourselves to the projective plane $\mathbb{P}^2$ and we replace the notion of hypersurfaces that intersect with a system of equations for which we want to find solutions. This yields a theorem that has the same flavour as Chasles' Theorem and Corollary 26. This theorem holds over any arbitrary field if all the intersection points Γ are defined over this field [18].

Theorem 28 (Cayley-Bacharach 3, 20th century)**.** *Consider the system of equations*

$$\begin{aligned} g_1(\overline{x}) &= 0 \\ &\vdots \\ g_n(\overline{x}) &= 0 \end{aligned}$$

of respective degrees $d_1, \dots, d_n$ with $d_1 d_2 \dots d_n$ isolated solutions Γ. If C is a curve of degree $\leq \sum_i d_i - n - 1$ containing all but one point of Γ, then C contains all of Γ.

Note that with this theorem we have entered the 20th century. Even tough Karasev [18] dates this theorem to the 19th century, it seems highly unlikely that this result was already known by then. That is because the proof requires some involved algebraic ideas that were only introduced in the 20th century. For example, it requires the notion of *Gorenstein Rings* and the intersections of the hypersurfaces are treated as *schemes*. We will not go into the details of this proof; the main takeaway from this Cayley-Bacharach Theorem is that it actually implies the Alon-Füredi Bound that can be found in Corollary 3.

Proof of Corollary 3 using Theorem 28. For every $S_i \subseteq \mathbb{F}$, we define $g_i = \prod_{s \in S_i}(x_i - s)$, such that every g_i has degree $d_i = |S_i|$. Then any polynomial p with $\deg p \leq \sum_i d_i - n - 1 = \sum_i (d_i - 1) - 1$ that vanishes on all common zeroes of $g_1, \ldots, g_n$ except one, vanishes on all of them. Hence, if p does vanish on $S_1 \times \cdots \times S_n$ except one point, $\deg p \geq \sum_i (d_i - 1)$. □

So with two different polynomial methods, we are able to find a lower bound on a grid cover where all points are covered once, except one point which is left uncovered. A sensible next step is to consider what happens if we do not want to cover every point once, but multiple times while still avoiding the origin. This brings us to the setting of the Ball-Serra bound.

2.4. The Ball-Serra Bound

In this section we consider the case where we do not cover every non-zero point of the grid only once, but k times, for some integer $k \geq 2$, while still avoiding the origin. We call such a cover a k-cover. Because of the increased multiplicity, the Alon-Füredi does not provide a useful bound. However, the polynomial method is able to provide another bound. Specifically, this section considers the *Punctured Combinatorial Nullstellensatz* formulated by Ball and Serra in [4]. This Nullstellensatz requires some notation. First of all, let $T(n, k)$ be the set of all non-decreasing sequences of length k on the set $[n]$. Note that non-decreasing means that a sequence can have repeated elements. Moreover, for any such sequence $\tau \in T(n, k)$, we let $\tau(i)$ denote its i-th element. When summing over elements in a sequence τ, repeated elements are counted multiple times.

Example 29. *Consider* $T(2, 3) = \{(1, 1, 1), (1, 1, 2), (1, 2, 2), (2, 2, 2)\}$. *Let* $\tau = (1, 1, 2)$, *then* $\tau(1) = \tau(2) = 1$ *and* $\tau(3) = 2$. *Summing over its elements becomes* $\sum_{i \in \tau} i = 1 + 1 + 2 = 4$.

Secondly, we need to define when a point vanishes with multiplicity k for a polynomial in an arbitary field.

Definition 30. *Consider an arbitrary field* $\mathbb{F}$. *We say that a polynomial* $f \in \mathbb{F}[x_1, \ldots, x_n]$ *has a zero of multiplicity* k *at point* $(a_1, \ldots, a_n) \in \mathbb{F}^n$ *if* k *is the maximum non-negative integer such that all monomials in the polynomial* $f(x_1 + a_1, \ldots, x_n + a_n)$ *in expanded form have degree* k.

Lastly, if we consider the grid $\Gamma = S_1 \times \cdots \times S_n$ in $\mathbb{F}^n$, we again formulate the polynomials

$$g_i(x_i) = \prod_{s \in S_i}(x_i - s),$$

such that Γ is the set of all common zeroes of $g_1, \ldots, g_n$. The set-up allows us to have some points on our grids that we do not cover. For these points, we define the subsets $D_i \subset S_i$ and polynomials

$$l_i(x_i) = \prod_{d \in D_i}(x_i - d).$$

Theorem 31 (Punctured Combinatorial Nullstellensatz). *If* f *vanishes at least* k *times at all elements of* $S_1 \times \cdots \times S_n$, *except at at least one point of* $D_1 \times \cdots \times D_n$ *where it has a zero of multiplicity less than* k, *then there are polynomials* h_τ *in* $\mathbb{F}[x_1, \ldots, x_n]$ *satisfying* $\deg(h_\tau) \leq \deg(f) - \sum_{i \in \tau} \deg(g_i)$ *and a nonzero polynomial satisfying* $\deg(u) \leq \deg(f) - \sum_{i=1}^{n} (\deg(g_i) - \deg(l_i))$, *such that*

$$f = \sum_{\tau \in T(n,k)} g_{\tau(1)} \cdots g_{\tau(k)} h_\tau + u \prod_{i=1}^{n} \frac{g_i}{l_i}.$$

And if there is a point of $D_1 \times \cdots \times D_n$ *where* f *does not vanish, then*

$$\deg(f) \geq (k-1) \max_{j \in [n]} \left(|S_j| - |D_j|\right) + \sum_{i=1}^{n} \left(|S_i| - |D_i|\right).$$

Example 32. *Suppose there exists a degree 12 polynomial that covers* $\{0,1,2,3\} \times \{0,1,2,3\} \setminus \{(0,0)\}$ *with multiplicity 3 while avoiding the origin. Note that such existence would give tightness in the lower bound on the degree of* f *as* $(3-1)\max_j\left(|S_j| - |D_j|\right) + \sum_{i=1}^{n}(|S_i| - |D_i|) = 2 \cdot 3 + 3 + 3 = 12$. *Then Theorem 31 implies that* f *can be written in the following form:*

$$f = h_{111} \cdot g_1^3 + h_{112} \cdot g_1^2 g_2 + h_{122} \cdot g_1 g_2^2 + h_{222} \cdot g_2^3 + u \cdot \frac{g_1}{l_1} \cdot \frac{g_2}{l_2},$$

where h_τ *are constants for all* $\tau \in T(2,3)$ *and* $\deg(u) \le 12 - 3 - 3 = 6$, $g_1(x) = x(x-1)(x-2)(x-3)$, $g_2(y) = y(y-1)(y-2)(y-3)$, $l_1(x) = x$ *and* $l_2(y) = y$. *Hence,*

$$\begin{aligned} f =& h_{111} \cdot (x(x-1)(x-2)(x-3))^3 + h_{112} \cdot (x(x-1)(x-2)(x-3))^2 \cdot y(y-1)(y-2)(y-3) + \\ & h_{122} \cdot x(x-1)(x-2)(x-3) \cdot (y(y-1)(y-2)(y-3))^2 + h_{222} \cdot (y(y-1)(y-2)(y-3))^3 + \\ & u \cdot (x-1)(x-2)(x-3)(y-1)(y-2)(y-3). \end{aligned}$$

The Ball-Serra bound will play a vital role in this thesis, so let us take a closer look at its proof. In the first place, we require Alon's Combinatorial Nullstellensatz [1].

Theorem 33 (Alon's Combinatorial Nullstellensatz)**.** *Let* $\mathbb{F}$ *be an arbitrary field. Suppose a polynomial* f *vanishes over all elements of* $S_1 \times \cdots \times S_n$, *that is,* $f(s_1, s_2, \ldots, s_n) = 0$ *for all* $s_i \in S_i$. *Then there exist polynomials* $h_1, h_2, \ldots, h_n \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ *with* $\deg(h_i) \le \deg(f) - \deg(g_i)$ *such that*

$$f = \sum_{i=1}^{n} h_i g_i.$$

To prove Alon's Combinatorial Nullstellensatz, we need another lemma regarding polynomials that vanish on grids [1].

Lemma 34. *Let* f *be a polynomial in* n *variables over an arbitrary field* $\mathbb{F}$. *Suppose that* $\deg_{x_i}(f) \le t_i$ *for* $i \in [n]$, *where* $\deg_{x_i}(f)$ *denotes the degree of* f *as a polynomial in* x_i. *For every* i, *let* $S_i \subseteq \mathbb{F}$ *such that* $|S_i| \ge t_i + 1$. *If* f *vanishes on all of* $S_1 \times \cdots \times S_n$, *then* $f \equiv 0$.

Proof. We will prove this lemma using induction on n. For the base case, when $n = 1$, the lemma just corresponds to the fact that a univariate polynomial of degree t_1 can have at most t_1 zeroes.
Now assume that the lemma holds for polynomials in $n-1$ variables and consider an n-variate polynomial and sets S_i satisfying the hypotheses. We can write

$$f = \sum_{i=0}^{t_n} f_i(x_1, \ldots, x_n) \cdot x_n^i,$$

where each f_i is a polynomial such that $\deg_{x_j}(f_i) \le t_j$ for all j. Moreover, for each fixed point $(x_1, \ldots, x_{n-1}) \in S_1 \times \cdots \times S_{n-1}$, the polynomial in x_n obtained by substituting $(x_1, \ldots, x_{n-1})$ vanishes for all $x_n \in S_n$, and is thus identically zero by the base case. Hence, $f_i(x_1, \ldots, x_{n-1}) = 0$ for all $(x_1, \ldots, x_{n-1}) \in S_1 \times \cdots \times S_{n-1}$. So, by the induction hypothesis $f_i \equiv 0$ for all i and so $f \equiv 0$. □

Proof of Alon's Combinatorial Nullstellensatz. Let $t_i = |S_i| - 1$ for all i. Moreover, we rewrite each g_i:

$$g_i(x_i) = \prod_{s \in S_i}(x_i - s) = x_i^{t_i+1} - \sum_{j=0}^{t_i} g_{ij} x_i^j,$$

for some coefficients g_{ij}. This enables us to say that if $x_i \in S_i$, then $g_i(x_i) = 0$, yielding the relation

$$x_i^{t_i+1} = \sum_{j=0}^{t_i} g_{ij} x_i^j.$$

Let f^* be the polynomial obtained by replacing each occurrence of $x_i^{u_i}$ with $u_i > t_i$ by a linear combination of smaller powers of x_i, using the above relation. For the resulting polynomial it holds that $\deg_{x_i}(f^*) \leq t_i$ for all i and f^* is obtained by subtracting products of the form $h_i g_i$ from f, where $h_i \in \mathbb{F}[x_1, \dots, x_n]$ and $\deg(h_i) \leq \deg(f) - \deg(g_i)$. Furthermore, $f^*(x_1, \dots, x_n) = f(x_1, \dots, x_n) = 0$ for all $(x_1, \dots, x_n) \in S_1 \times \cdots \times S_n$, because the relations used to replace the x_i with high degree hold exactly for these points. By the above lemma, $f^* \equiv 0$ and hence,

$$f = \sum_{i=1}^{n} h_i g_i. \qquad \square$$

We generalise this statement to the case where f vanishes with multiplicity at all common elements of $S_1 \times \cdots \times S_n$.

Theorem 35. *Suppose a polynomial f vanishes with multiplicity k over all elements of $S_1 \times \cdots \times S_n$. Then there exist polynomials $h_\tau \in \mathbb{F}[X_1, X_2, \dots, X_n]$ for every $\tau \in T(n,k)$ with $\deg(h_\tau) \leq \deg(f) - \sum_{i \in \tau} \deg(g_i)$ such that*

$$f = \sum_{\tau \in T(n,k)} g_{\tau(1)} \cdots g_{\tau(k)} h_\tau.$$

Proof. This theorem is proven using double induction on n and k. If $k = 1$, then this just corresponds to Alon's Combinatorial Nullstellensatz. Moreover, if $n = 1$, then f is a univariate polynomial that vanishes with multiplicity k on all $s_i \in S_1$. Hence, $f(x) = \prod_{s_i \in S_i} (x - s_i)^k \cdot h(x) = g^k(x) \cdot h(x)$. Since $T(1,k)$ only contains one sequence, namely the all-1 sequence, this indeed corresponds to the formula in the theorem.

For the induction hypothesis, assume that the statement holds for $m < n$ and $\ell \leq k$ and for $m \leq n$ and $\ell < k$. Now suppose f is an n-variate polynomial that vanishes on all of $S_1 \times \cdots \times S_n$ with multiplicity k. Let $\alpha \in S_n$. Then we write $f = (x_n - \alpha)A_\alpha + B_\alpha$, with $A_\alpha \in \mathbb{F}[x_1, \dots, x_n]$ and $B_\alpha \in \mathbb{F}[x_1, \dots, x_{n-1}]$. We know that $(s_1, \dots, s_{n-1}, \alpha)$ is a root of f with multiplicity k for all $s_1 \in S_1, \dots, s_n \in S_n$. Hence, B_α vanishes with multiplicity k at all points of $S_1 \times \cdots \times S_{n-1}$ and by the induction hypothesis,

$$B_\alpha = \sum_{\kappa \in T(n-1,k)} g_{\kappa(1)} \cdots g_{\kappa(k)} h_\kappa,$$

where $\deg(h_\kappa) \leq \deg(B_\alpha) - \sum_{i \in \kappa} \deg(g_i) \leq \deg(f) - \sum_{i \in \kappa} \deg(g_i)$.

Now we split A_α into two polynomials by considering $\beta \in S_n, \beta \neq \alpha$. Then we write $A_\alpha = (x_n - \beta)A_\beta + B_\beta$. We again apply the induction hypothesis, this time on B_β:

$$B_\beta = \sum_{\kappa \in T(n-1,k)} g_{\kappa(1)} \cdots g_{\kappa(k)} l_\kappa,$$

with $\deg(l_\kappa) \leq \deg(B_\beta) - \sum_{i \in \kappa} \deg(g_i) \leq \deg(f) - 1 - \sum_{i \in \kappa} \deg(g_i)$. We plug this into the original expression for f to obtain

$$f = (x_n - \alpha)(x_n - \beta)A_\beta + U_{\alpha\beta},$$
$$U_{\alpha\beta} = \sum_{\kappa \in T(n-1,k)} g_{\kappa(1)} \cdots g_{\kappa(k)} m_\kappa$$

and $\deg(m_\kappa) \leq \deg(f) - \sum_{i \in \kappa} \deg(g_i)$. We keep repeating this to write f in the form

$$f = \prod_{s_i \in S_n} (x_n - s_i) \cdot A + B = g_n(x_n) \cdot A + B,$$

where $\deg(A) \leq \deg(f) - \deg(g_n)$.

Moreover, B can be written as

$$B = \sum_{\kappa \in T(n-1,k)} g_{\kappa(1)} \cdots g_{\kappa(k)} w_\kappa,$$

such that $\deg(w_\kappa) \leq \deg(f) - \sum_{i\in\kappa}\deg(g_i)$. Since $g_n(x_n)\cdot A$ has a zero of multiplicity k at all points of $S_1\times\cdots\times S_n$ and g_n has a zero of multiplicity 1 at all these zeroes, A vanishes with multiplicity $k-1$ on $S_1\times\cdots\times S_n$. Therefore we can use the induction hypothesis on A too:

$$A = \sum_{\eta\in T(n,k-1)} g_{\eta(1)}\cdots g_{\eta(k-1)}p_\eta,$$

with $\deg(p_\eta) \leq \deg(A) - \sum_{i\in\eta}\deg(g_i)$. Putting everything together yields

$$\begin{aligned} f &= g_n\cdot\sum_{\eta\in T(n,k-1)} g_{\eta(1)}\cdots g_{\eta(k-1)}p_\eta + \sum_{\kappa\in T(n-1,k)} g_{\kappa(1)}\cdots g_{\kappa(k)}w_\kappa \\ &= \underbrace{\sum_{\eta\in T(n,k-1)} g_{\eta(1)}\cdots g_{\eta(k-1)}g_n\,p_\eta}_{\text{includes all }\tau\in T(n,k)\text{ with }\tau(k)=n} + \underbrace{\sum_{\kappa\in T(n-1,k)} g_{\kappa(1)}\cdots g_{\kappa(k)}}_{\text{includes all }\tau\in T(n,k)\text{ with }\tau(k)\neq n}\; w_\kappa \\ &= \sum_{\tau\in T(n,k)} g_{\tau(1)}\cdots g_{\tau(k)}h_\tau, \end{aligned}$$

with $\deg(h_\tau) \leq \deg(f) - \sum_{i\in\tau}\deg(g_i)$, which is exactly what we wanted to prove. □

Now we finally proceed to prove the Ball-Serra Bound.

Proof of the Punctured Combinatorial Nullstellensatz. Suppose that f vanishes with multiplicity k at all grid points of $S_1\times\cdots\times S_n$, except at the points of $D_1\times\cdots\times D_n$. At those points, f has a zero of multiplicity less than k. Then we can write f in the form

$$f = \sum_{\tau\in T(n,k)} g_{\tau(1)}\cdots g_{\tau(k)}h_\tau + w,$$

where we note that w does not contain a monomial $x_1^{\ell_1}\ldots x_n^{\ell_n}$ such that there is a $\tau\in T(n,k)$ with $\ell_j \geq \mathrm{mult}(j,\tau)\cdot|S_j|$ for all $j\in[n]$, where $\mathrm{mult}(j,\tau)$ denotes the number of occurrences of j in τ. We can assume that f has this form because if there is such a monomial in w for a certain κ, we can obtain $w' = w - g_{\kappa(1)}\ldots g_{\kappa(k)}$ and therefore $f = \sum_{\tau\in T(n,k)} g_{\tau(1)}\cdots g_{\tau(k)}h'_\tau + w'$.
But maybe more importantly, we know that $f\cdot l_i^k$ vanishes at all common zeroes of $g_1,\ldots,g_n$. As these common zeroes are clearly also zeroes of $\sum_{\tau\in T(n,k)} g_{\tau(1)}\cdots g_{\tau(k)}h_\tau$, we know that $w\cdot l_i^k$ should also vanish at these points $(s_1,\ldots,s_n)\in S_1\times\cdots\times S_n$. If we regard $w\cdot l_i^k$ as a univariate polynomial in x_i, we find that g_i divides this polynomial. Since l_i divides g_i, that means that $\frac{g_i}{l_i}$ divides $w\cdot l_i^{k-1}$. But $\frac{g_i}{l_i}$ cannot divide l_i^{k-1}, since $\frac{g_i}{l_i} = \prod_{s_i\in S_i\setminus D_i}(x_i - s_i)$ and $l_i = \prod_{s_i\in D_i}(x_i - s_i)$. Therefore we know that $\frac{g_i}{l_i}$ divides w. This holds for every i, so $w = u\prod_{i\in[n]}\frac{g_i}{l_i}$ for some polynomial u. Hence,

$$f = \sum_{\tau\in T(n,k)} g_{\tau(1)}\cdots g_{\tau(k)}h_\tau + u\prod_{i\in[n]}\frac{g_i}{l_i}.$$

Moreover, u cannot be the zero polynomial, because otherwise f would vanish on all common zeroes of $g_1,\ldots,g_n$, which contradicts our assumptions. This concludes the first part of the statement. Now we still have to prove the bound on the degree. To do so, let $d_2\in D_2,\ldots,d_n\in D_n$ and consider $f(x_1,d_2,\ldots,d_n)$. Since $f(x_1,d_2,\ldots,d_n)$ vanishes with multiplicity k on all $s\in S_1\setminus D_1$, we know that $\left(\frac{g_1}{l_1}\right)^k$ divides $f(x_1,d_2,\ldots,d_n)$. Furthermore, evaluating $(x_1,d_2,\ldots,d_n)$ in

$$\sum_{\tau\in T(n,k)} g_{\tau(1)}\cdots g_{\tau(k)}h_\tau + u\cdot\prod_{s_{i_j}\in S_j\setminus D_j}(x_j - s_{i_j})$$

yields

$$g_1(x_1)^k h_{1\ldots1} + c\cdot u(x_1,d_2,\ldots,d_n)\cdot\prod_{s\in S_1\setminus D_1}(x_1 - s),$$

for some constant c. Since $\left(\frac{g_1}{l_1}\right)^k$ divides this sum, it has to divide both terms. For the first term this is obviously true, for the second one this implies that $\left(\frac{g_1}{l_1}\right)^{k-1}$ divides $u(x_1, d_2, \ldots, d_n)$. Therefore, $\deg(u) \geq (k-1) \cdot (|S_1| - |D_1|)$, and this argument can be repeated for any $j \in [n]$. Hence,

$$\deg(u) \geq (k-1) \cdot (|S_j| - |D_j|), \quad \forall j \in [n].$$

Since $\deg\left(\prod_{i\in[n]} \frac{g_i}{l_i}\right) = \sum_i (|S_i| - |D_i|)$, we find for all $j \in [n]$,

$$\deg(f) \geq (k-1) \cdot (|S_j| - |D_j|) + \sum_{i\in[n]} (|S_i| - |D_i|). \qquad \square$$

Thus, if we would like to cover a certain grid $S_1 \times \cdots \times S_n$ k times, where all S_i are subsets of some field $\mathbb{F}$, while completely avoiding the origin, then the Ball-Serra Bound implies that we need at least

$$(k-1) \max_j \left(|S_j| - 1\right) + \sum_{i=1}^{n} (|S_i| - 1)$$

hyperplanes.

2.5. Conclusion

This section provided lower bounds on the sizes of grid covers, depending on their multiplicity. First, we considered a grid cover where all points of $\Gamma = S_1 \times \cdots \times S_n$ are covered with multiplicity one, except for one point that must remain uncovered. The Alon-Füredi Bound implies that such a cover has size at least $\sum_i (|S_i| - 1)$. This bound could easily be proven in the case when $\Gamma = Q^n$, by making use of the fact that the polynomial that we define from the cover only needs to vanish on binary vectors. In the general setting, proving this lower bound required some more complicated polynomial methods. Specifically, we used the Footprint Bound, which states that the number of common zeroes of polynomials in an ideal is upper bounded by the number of standard monomials of that ideal. By considering a grid as a set of common zeroes of univariate polynomials, we could translate this upper bound to the upper bound given by Alon-Füredi. Interpreting a grid as such a set of common zeroes also enabled us to prove the Alon-Füredi Bound using one of the Cayley-Bacharach Theorems. This is an older theorem that links the dimension of the vector space of curves of given degree that vanish on a set to the failure of the ”complementary” set to curves of ”complementary” degree.
When we increase the multiplicity of the grid cover, the Alon-Füredi Bound no longer holds. In that case, Ball and Serra provided a different bound with their Punctured Combinatorial Nullstellensatz. They showed that the minimum size of the cover is then equal to $(k-1)\max_j \left(|S_j| - 1\right) + \sum_{i=1}^{n} (|S_i| - 1)$. This bound was proven using polynomial methods too, this time using Alon's Combinatorial Nullstellensatz. This Nullstellensatz originally gave the form of a polynomial that vanishes on the entire grid $\Gamma = S_1 \times \cdots \times S_n$ with multiplicity 1. Adapting this form to the case where f vanishes with multiplicity k on Γ except at one point, where it does not vanish, yields the Ball-Serra Bound. While in the case of the Alon-Füredi we have seen that this lower bound is actually tight, it's not as clear whether the Ball-Serra bound is tight too. That is what is investigated in the next chapter.

3

Tightness of the Ball-Serra Bound

The Ball-Serra Bound states that a polynomial that vanishes at all points of $S_1 \times \cdots \times S_n$ with multiplicity k, except at one uncovered point, has degree at least $(k-1)\max_j \left(|S_j| - 1\right) + \sum_{i=1}^{n} (|S_i| - 1)$. In particular, this minimum degree immediately gives a lower bound on the number of hyperplanes required to form a hyperplane k-cover of $S_1 \times \cdots \times S_n$. However, the bound does not assume anything about the polynomial except its roots. In this light, we investigate two things in this chapter. For starters, we examine whether the Ball-Serra Bound is tight for hyperplane covers. Next, in the cases where we do not know if we have tightness, we explore if the Ball-Serra Bound is tight when we omit the constraint that the polynomial needs to be a product of hyperplanes, i.e. a product of degree 1 polynomials. Such a polynomial is called a polynomial k-cover.

We start by introducing the basic ideas in an analysis of tightness of the Ball-Serra Bound for the hypercube in Section 3.1. We give the best lower bounds possible using the polynomial method and compare them with the conjectured hyperplane covering number. The polynomial method for the hypercube interestingly also raises some questions about the polynomial covering number of the vector space $\mathbb{F}_2^n$ over the binary field. Both the polynomial and the hyperplane covering number of the binary field are addressed in Section 3.2. The largest part of this chapter is spent on how to cover grids in the Cartesian plane. While the behaviour of the hyperplane covering number has been studied fairly well, the polynomial covering number has not been studied yet and hence is unknown. In the remaining sections of this chapter we look into multiple approaches taken to figure out when the Ball-Serra Bound is tight for these grids. At first, we generate different polynomial 3-covers using an algorithm. We investigate the different properties that these polynomial covers in the plane can have. Based on the example covers, a threshold is proposed when the Ball-Serra Bound is tight for polynomial covers of grids in the Cartesian plane. To further investigate the 3-covers we also look at a different method to construct them, based on slices of the grid. From this method, the exact same threshold seems to arise. As there are still open questions regarding proof methods, this threshold is presented as a conjecture. Generalising the analysis of the 3-covers, we are also able to conjecture a threshold for any polynomial k-cover and to conjecture the behaviour of the polynomial k-covering number for grids that do not satisfy the threshold. In this chapter, we always assume that the considered grid $S_1 \times S_2 \times \cdots \times S_n$ has the property $|S_1| \geq |S_2| \geq \cdots \geq |S_n|$, which we are allowed to do because of symmetry.

3.1. Covers of the Hypercube

Just as with the Alon-Füredi bound, we start our analysis of the bound by looking at the hypercube, making use of the fact that all points that we want to cover are binary. Note that for the hypercube $Q^n = \{0, 1\}^n$, the Ball-Serra Bound implies that a k-cover has size at least $(k-1) + n$. For $k = 1$, this coincides with the – tight – Alon-Füredi Bound. Before looking into covers with higher multiplicities, it is important to establish a method to actually search for polynomials that vanish with the right multiplicity. Already in the analysis of the Cayley-Bacharach Theorem in Section 2.3, we have seen that requiring that a polynomial vanishes at a point imposes a linear constraint on the coefficients of that polynomial. In a similar fashion, there are linear constraints that encode that we require that the polynomial should vanish with higher multiplicity at that point. For a polynomial f to vanish at some point (u, v) with multiplicity k, we should have

$$f(u,v) = \frac{\partial^{(i+j)} f}{\partial x^i \partial y^j}(u,v) = 0 \text{ for all } 0 \le i+j \le k.$$

Hence to find out whether there exists a polynomial cover of a certain degree for a given grid, we can verify whether there exists a solution on the linear system constructed with the above constraints, where the coefficients of the polynomial are variables in the system.

Example 36. *Suppose we would like to find out whether there is a polynomial 2-cover of degree 3 for the grid $Q^2 = \{0,1\} \times \{0,1\}$. Then we know that such a polynomial needs to be of the form*

$$f(x,y) = a_{30}x^3 + a_{21}x^2y + a_{12}xy^2 + a_{03}y^3 + a_{20}x^2 + a_{11}xy + a_{02}y^2 + a_{10}x + a_{01}y + 1.$$

We are allowed to set the constant term equal to 1, because the polynomial should not vanish at the origin. Imposing that the points $(1,1)$, $(1,0)$ and $(0,1)$ are covered by f yields the linear constraints:

$$\begin{aligned} a_{30} + a_{21} + a_{12} + a_{03} + a_{20} + a_{11} + a_{02} + a_{10} + a_{01} + 1 &= 0 \\ a_{30} \qquad\qquad\qquad + a_{20} \qquad\qquad + a_{10} \qquad + 1 &= 0 \\ a_{03} \qquad\qquad + a_{02} \qquad + a_{01} + 1 &= 0. \end{aligned}$$

Since f has to form a 2-cover, we should also take into account the derivatives of first order:

$$\begin{aligned} \frac{\partial f}{\partial x} &= 3 \cdot a_{30}x^2 + 2 \cdot a_{21}xy + a_{12}y^2 + 2 \cdot a_{20}x + a_{11}y + a_{10} \\ \frac{\partial f}{\partial y} &= a_{21}x^2 + 2 \cdot a_{12}xy + 3 \cdot a_{03}y^2 + a_{11}x + 2 \cdot a_{02}y + a_{01}. \end{aligned}$$

We require that $(1,1)$, $(1,0)$ and $(0,1)$ are also roots of these polynomials and combine all the equations in the system in a matrix. Then we perform Gaussian Elimination on the matrix to find out whether there exists a polynomial with coefficients that satisfy the constraints.

$$\left[\begin{array}{ccccccccc|c} a_{30} & a_{21} & a_{12} & a_{03} & a_{20} & a_{11} & a_{02} & a_{10} & a_{01} & \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & -1 \\ 3 & 2 & 1 & 0 & 2 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 & 1 & 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 2 & 0 & 1 & 0 \end{array}\right]$$

$$\sim \left[\begin{array}{ccccccccc|c} a_{30} & a_{21} & a_{12} & a_{03} & a_{20} & a_{11} & a_{02} & a_{10} & a_{01} & \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 \end{array}\right].$$

This shows that there is a unique degree-3 polynomial that covers every non-zero point of $\{0,1\}\times\{0,1\}$ *twice while avoiding the origin. Its function is given by the row reduced echelon form and equals*

$$f(x,y) = -x^2y - xy^2 + x^2 + 3xy + y^2 - 2x - 2y + 1.$$

And hence, the Ball-Serra Bound is tight for this grid. We factorise f *to obtain*

$$f(x,y) = -(x+y-1)(x-1)(y-1),$$

showing that the unique tight polynomial 2-cover for Q^2 *is in fact a hyperplane 2-cover.*

Now that we have established a method to ensure that a polynomial vanishes with the right multiplicity at a point, we continue the analysis of k-covers of the hypercube. For $k = 2$, Example 36 has shown that there exist instances for which the Ball-Serra Bound, which is equal to $n+1$, is tight. In fact, the bound is tight for $k = 2$. A 2-cover of the hypercube of size $n+1$ can be obtained by generalising the cover from the example and taking the n hyperplanes of the form $x_i = 1$ and adding the hyperplane $\sum_i x_i = 1$. Starting from a 3-cover, things become more complicated and hence also more interesting. Exactly these covers were investigated by Clifton and Huang [11] in 2020. They came up with the following theorem.

Theorem 37 (Clifton-Huang)**.** *Let* $f(n,k)$ *denote the minimum size of a hyperplane* k*-cover of* Q^n*. For* $n \geq 2$*,*

$$f(n,3) = n+3.$$

For $k \geq 4$ *and* $n \geq 3$*,*

$$n+k+1 \leq f(n,k) \leq n + \binom{k}{2}.$$

We give an outline of the proof.

Proof. First of all, note that for $k = 3$, $n+3$ is equal to $n + \binom{k}{2}$. A k-cover of this size is given by: $x_i = i$ for $i \in [n]$ together with $k-\ell$ copies of $\sum_{i=1}^{n} x_i = \ell$ for $\ell \in [k-1]$. This set of hyperplanes covers every point of Hamming weight (number of 1's in its coordinate) equal to ℓ exactly ℓ times by $x_i = 1$ and $k-\ell$ times by $\sum_{i=1}^{n} x_i = \ell$. And the size of this cover is indeed $n + \sum_{\ell=1}^{k-1}(k-\ell) = n + \binom{k}{2}$. Hence, for $k = 3$, we find $n+2 \leq f(n,k) \leq n+3$.
Suppose there exists a 3-cover of Q^n of $n+2$ hyperplanes $H_1, \dots, H_{n+2}$. We can assume that every hyperplane H_i is defined by $\overline{a_i} \cdot \overline{x} = 1$ for $\overline{a_i}, \overline{x} \in \mathbb{R}^n$. So we define the polynomials $p_i = \overline{a_i} \cdot \overline{x} - 1$ and $f = p_1 \dots p_{n+2}$. Using the Combinatorial Nullstellensatz with $D_i = \{0\}$, $S_i = \{0,1\}$, $g_i = x_i(x_i-1)$ and $l_i = x_i$ we write f in the form

$$f = \sum_{1 \leq i \leq j \leq k \leq n} x_i(x_i-1)x_j(x_j-1)x_k(x_k-1)h_{ijk} + u \cdot \prod_{i=1}^{n}(x_i - 1),$$

with $\deg(u) \leq 2$.
It is easy to see that $f(\overline{x}) = 0$ for all $\overline{x} \in Q^n \setminus \{\overline{0}\}$. But for f to have a zero of multiplicity 3 at every point of $Q^n \setminus \{\overline{0}\}$, we need all its partial derivatives up to second order to vanish on $Q^n \setminus \{\overline{0}\}$. Since $x_i(x_i-1)x_j(x_j-1)x_k(x_k-1)h_{ijk}$ has its first and second order derivatives vanishing on $Q^n \setminus \{\overline{0}\}$, we require

$$\frac{\partial}{\partial x_i}\left(u \cdot \prod_{i=1}^{n}(x_i - 1)\right)(\overline{x}) = 0$$
$$\frac{\partial^2}{\partial x_i \partial x_j}\left(u \cdot \prod_{i=1}^{n}(x_i - 1)\right)(\overline{x}) = 0$$

for all $i, j \in [n]$ and all $\overline{x} \in Q^n \setminus \{\overline{0}\}$. Working this out explicitly will imply that $u \equiv 0$, a contradiction. Hence $f(n,3) = n+3$ for $n \geq 2$.

Exactly similar, one can prove that $f(n,4) \in \{n+5, n+6\}$ for $n \geq 3$. Since removing 1 hyperplane from a k-cover leaves us with a $(k-1)$-cover, we have

$$f(n,k) \geq f(n,k-1) + 1.$$

So for $k \geq 4$ and $n \geq 3$ we find

$$f(n,k) \geq f(n,4) + (k-4) \geq n+5+(k-4) = n+k+1. \qquad \square$$

This result shows that Ball-Serra is actually only tight for $k=1$ or $k=2$. In any other case, Clifton and Huang provide a better lower bound. Moreover, they conjecture that the upper bound $n+\binom{k}{2}$ is tight, rather than the lower bound. The most significant progress in increasing this lower bound was made by Sauermann and Wigderson in [19] in 2022, again using the polynomial method rather than looking at the geometric properties of the problem. They were able to improve the lower bound to

$$n + 2k - 3 \leq f(n,k).$$

Specifically, they showed the following theorem.

Theorem 38 (Sauermann-Wigderson)**.** *Let $k \geq 2$ and $n \geq 2k-3$. Then any polynomial $p \in \mathbb{R}[x_1, \dots, x_n]$ with $p(\overline{0}) \neq 0$ having zeroes of multiplicity at least k at all points in $\{0,1\}^n \backslash \{\overline{0}\}$ has degree $\deg p \geq n+2k-3$. Furthermore, there exists such a polynomial p with degree $\deg p = n+2k-3$.*

There are a couple of interesting things to note from this statement. First of all, this shows that we cannot expect to increase the lower bound by only using arguments related to polynomials. Since there exists a polynomial of degree $n+2k-3$ that satisfies the above properties, the lower bound cannot be increased further using the polynomial method. This implies that if the conjecture of Clifton and Huang is correct and the hyperplane k-cover number is equal to $n+\binom{k}{2}$, there is a different regime in the polynomial cover number and the hyperplane cover number. Furthermore, it also means that to find a better lower bound, other arguments are required that specifically use the fact that we are considering a hyperplane cover. Secondly, the theorem has now been formulated specifically for polynomials over the reals. The result actually holds in some more cases, but for those we need the notion of the characteristic of a field.

Definition 39. *Let $\mathbb{F}$ be an arbitrary field. The characteristic of $\mathbb{F}$, denoted by $\mathrm{char}(K)$ is the smallest non-negative number m such that $m \cdot 1_{\mathbb{F}}$ equals $0_{\mathbb{F}}$, where $1_{\mathbb{F}}$ is the multiplicative identity of $\mathbb{F}$ and $0_{\mathbb{F}}$ the additive identity. If no such number exists, the field is said to have characteristic 0.*

The Sauermann-Wigderson holds for all fields with characteristic zero. What happens for fields with positive characteristic will be investigated in Section 3.2. Sauermann and Wigderson also looked at *almost* k-covers of the hypercube where the origin is covered exactly ℓ times, where $\ell \in \{0, \dots, k-2\}$.

Theorem 40. *Let $k \geq 2$ and $n \geq 2k-3$. Let $p \in \mathbb{R}[x_1, \dots, x_n]$ be a polynomial having zeroes of multiplicity at least k at all points in $\{0,1\}^n \backslash \{\overline{0}\}$, and such that p does not have a zero of multiplicity at least $k-1$ at the origin. Then p must have $\deg p \geq n+2k-3$. Furthermore, for every $\ell \in [k-2]$, there exists a polynomial p with $\deg p = n+2k-3$ having zeroes of multiplicity at least k at all points in $\{0,1\}^n \backslash \{\overline{0}\}$, and such that p has a zero of multiplicity exactly ℓ at $\overline{0}$.*

The case when $\ell = k-1$ is slightly different.

Theorem 41. *Let $k \geq 2$ and $n \geq 1$. Let $p \in \mathbb{R}[x_1, \dots, x_n]$ be a polynomial having zeroes of multiplicity at least k at all points in $\{0,1\}^n \backslash \{\overline{0}\}$, and a zero of multiplicity exactly $k-1$ at the origin. Then p must have $\deg p \geq n+2k-2$. Furthermore, there exists such a polynomial p with $\deg p = n+2k-2$.*

This last theorem holds for any field, independent of its characteristic. So we are able to conclude that the Ball-Serra bound is only tight for the hypercube in very few cases, namely for covers with multiplicity 1, which is in fact the Alon-Füredi bound, and for covers with multiplicity 2. For higher multiplicity, the best known lower bound is given by Sauermann and Wigderson and states that $f(n,k) \geq n+2k-3$. The conjectured hyperplane k-covering number is $n+\binom{k}{2}$ and to prove this to be true, other arguments than the polynomial method are required. Furthermore, Ball-Serra is not even tight if we allow the origin to be covered with lower multiplicity than the other grid points. Suppose we allow the origin to be covered ℓ times with $\ell < k$ and let $f(n,k;\ell)$ denote the minimum size of such an almost k-cover, then $f(n,k;\ell) \geq n+2k-3$ for $\ell \in [k-2]$ and $f(n,k;k-1) \geq n+2k-2$.

3.2. Covers of the Binary Field

As we have noted previously, Theorem 38 only holds for fields with characteristic 0. If we consider a field K with odd characteristic $\mathrm{char}(K) > 0$ or with characteristic 2, there exists an integer $k > 1$ such that we can actually find a polynomial $p \in K[x_1, \dots, x_n]$ of degree $\leq n + 2k - 4$ that vanishes on $K \setminus \{\overline{0}\}$ with multiplicity k while p has a non-zero value at the origin [19]. In this section, we provide a short analysis of the case where K is the binary field $\mathbb{F}_2$. As one can expect, by considering a finite field instead of the hypercube, both the hyperplane and polynomial covers exhibit different behaviours. To verify the multiplicity of a root of a polynomial in a finite field, the regular derivative cannot be used anymore. Instead, we have to consider a generalisation of the derivative of a polynomial, called *Hasse derivatives* [6].

Definition 42. *Let $K[x]$ be a polynomial ring of positive characteristic. The Hasse derivative of order r of x^n is equal to*

$$D^{(r)} = \binom{n}{r} x^{n-r},$$

if $r \leq n$ and 0 otherwise.

In the multivariate case, we take the derivatives sequentially, e.g.:

$$\frac{D^{(2)}}{Dx_i Dx_j} p = \frac{D^{(1)}}{Dx_i} \left(\frac{D^{(1)}}{Dx_j} p \right) = \frac{D^{(1)}}{Dx_j} \left(\frac{D^{(1)}}{Dx_i} p \right).$$

From now on, for the first order partial derivative, the '(1)' in the exponent will be omitted and the derivative will simply be denoted by $\frac{D}{Dx_i}$. The product rule of the Hasse derivative is slightly different compared to the normal derivative. To derive this product rule, we need an elemental equality in combinatorics: the Vandermonde's Identity.

Theorem 43 (Vandermonde's Identity)**.** *Let m, n, r be integers such that $r \leq n + m$. Then*

$$\binom{n+m}{r} = \sum_{i=0}^{r} \binom{n}{i} \binom{m}{r-i}.$$

Proof. Let S_n, S_m be two sets of respective sizes n and m. The binomial coefficient $\binom{n+m}{r}$ is the number of ways to pick r elements out of $S_n \cup S_m$. This is equal to the sum of all ways of first picking i elements out of S_n and then $r - i$ elements out of S_m, for $i = 0, \dots, r$. This is equal to $\sum_{i=0}^{k} \binom{n}{i} \binom{m}{r-i}$. □

This identity almost immediately provides the product rule for Hasse derivatives.

Lemma 44 (Product Rule)**.** *Let $f, g \in K[x]$ and let $r \in \mathbb{N}$. Then the r-th order Hasse derivative of $f \cdot g$ is given by*

$$D^{(r)}(f \cdot g) = \sum_{i=0}^{r} D^{(i)} f \cdot D^{(r-i)} g.$$

Proof. To prove this equality, it suffices to look at $D^{(r)}(x^n x^m) = D^{(r)}(x^{n+m})$. On one hand side, this is equal to $\binom{n+m}{r} x^{n+m-r}$. On the other hand we obtain $\sum_{i=0}^{r} D^{(i)} x^n \cdot D^{(r-i)} x^m$. This is equal to $\sum_{i=0}^{r} \binom{n}{i} \cdot x^{n-i} \binom{m}{r-i} \cdot x^{m-r+i} = \sum_{i=0}^{r} \binom{n}{i} \binom{m}{r-i} \cdot x^{n+m-r}$, which is indeed equal to the first expression by the Vandermonde Identity. □

A similar product rule can be derived in the multivariate case.

Lemma 45 (Multivariate Product Rule)**.** *Let $f, g \in K[x_1, \dots, x_\ell]$ and let $r, r_1, r_2, \dots, r_\ell \in \mathbb{N}$ such that $r_1 + r_2 + \dots + r_\ell = r$. Then any r-th order partial derivative on ℓ variables is given by*

$$\frac{D^{(r)}}{Dx_1^{r_1} Dx_2^{r_2} \cdots Dx_\ell^{r_\ell}} (f \cdot g) = \sum_{n_1=0}^{r_1} \sum_{n_2=0}^{r_2} \cdots \sum_{n_\ell=0}^{r_\ell} \frac{D^{(n_1)}}{Dx_1^{n_1}} \frac{D^{(n_2)}}{Dx_2^{n_2}} \cdots \frac{D^{(n_\ell)}}{Dx_\ell^{n_\ell}} f \cdot \frac{D^{(r_1-n_1)}}{Dx_1^{r_1-n_1}} \frac{D^{(r_2-n_2)}}{Dx_2^{r_2-n_2}} \cdots \frac{D^{(r_\ell-n_\ell)}}{Dx_\ell^{r_\ell-n_\ell}} g.$$

So in other words, the r_i derivatives that have to be taken of the product get split into all possible ways between f and g. The proof is obtained by induction on ℓ.

Proof. Suppose $\ell = 1$, then this is just the univariate product rule that has been proved above. Hence, suppose the formula to hold for $\ell < k$ and consider

$$\begin{aligned}\frac{D^{(r)}}{Dx_1^{r_1}Dx_2^{r_2}\cdots Dx_k^{r_k}}f\cdot g &= \frac{D^{(r_1)}}{Dx_1^{r_1}}\left(\frac{D^{(r-r_1)}}{Dx_2^{r_2}\cdots Dx_k^{r_k}}f\cdot g\right)\\ &= \frac{D^{(r_1)}}{Dx_1^{r_1}}\left(\sum_{n_2=0}^{r_2}\cdots\sum_{n_k=0}^{r_k}\frac{D^{(n_2)}}{Dx_2^{n_2}}\cdots\frac{D^{(n_k)}}{Dx_k^{n_k}}f\cdot\frac{D^{(r_2-n_2)}}{Dx_2^{r_2-n_2}}\cdots\frac{D^{(r_k-n_k)}}{Dx_k^{r_k-n_k}}g\right)\\ &= \sum_{n_1=0}^{r_1}\sum_{n_2=0}^{r_2}\cdots\sum_{n_k=0}^{r_k}\frac{D^{(n_1)}}{Dx_1^{n_1}}\frac{D^{(n_2)}}{Dx_2^{n_2}}\cdots\frac{D^{(n_k)}}{Dx_k^{n_k}}f\cdot\frac{D^{(r_1-n_1)}}{Dx_1^{r_1-n_1}}\frac{D^{(r_2-n_2)}}{Dx_2^{r_2-n_2}}\cdots\frac{D^{(r_k-n_k)}}{Dx_k^{r_k-n_k}}g.\end{aligned}$$ □

The Hasse derivative allows us to say that a zero of a polynomial in a finite field has multiplicity k if all derivatives up to order $k-1$ vanish at that point. Having established this framework, we are able investigate Theorem 38 in the binary field. For $k=4$, the polynomial

$$f_4(\overline{x}) = \left(\prod_{\ell=1}^{n}(x_\ell+1)\right)\cdot\left(1+\sum_{i=1}^{n}(x_i^3+x_i^2+x_i)+\sum_{1\le i\ne j\le n}(x_i^3+x_i^2)x_j+\sum_{1\le i<j\le n}x_ix_j+\sum_{1\le i<j<k\le n}x_ix_jx_k\right)$$

vanishes with multiplicity k on $\mathbb{F}_2^n\setminus\{\overline{0}\}$ while $f_4(\overline{0})\ne 0$. The degree of p is equal to $n+4=n+2k-4$. That means that indeed Theorem 38 does not hold in this field. From its formula, it is clear that $f_4(\overline{x})=0$ with multiplicity at least 4 for all $\overline{x}$ with Hamming weight greater than or equal to 4. It can also be easily seen that indeed $f_4(\overline{0})\ne 0$. It remains to be verified that for the polynomial

$$p_4(\overline{x}) = \left(1+\sum_{i=1}^{n}(x_i^3+x_i^2+x_i)+\sum_{1\le i\ne j\le n}(x_i^3+x_i^2)x_j+\sum_{1\le i<j\le n}x_ix_j+\sum_{1\le i<j<k\le n}x_ix_jx_k\right),$$

any vector with Hamming weight 1 vanishes with multiplicity 3, any vector of Hamming weight 2 vanishes with multiplicity 2 and any vector of Hamming weight 3 vanishes with multiplicity 1. It can be easily seen that any such vector indeed is a root of f_4. To verify the multiplicities of the vectors of Hamming weight 1 and 2, we compute the Hasse derivatives.

$$\begin{aligned}\frac{Dp_4}{Dx_i} &= 3x_i^2+2x_i+1+(3x_i^2+2x_i)\sum_{j\ne i}x_j+\sum_{j\ne i}(x_j^3+x_j^2)+\sum_{j\ne i}x_j+\sum_{\substack{j,k\ne i\\ j<k}}x_jx_k\\ &\equiv x_i+1+x_i\sum_{j\ne i}x_j+\sum_{j\ne i}x_j+\sum_{\substack{j,k\ne i\\ j<k}}x_jx_k,\end{aligned}$$

where $\equiv$ denotes that both expressions evaluate to the same over $\mathbb{F}_2^n$. Hence, it follows that

$$\frac{Dp_4}{Dx_i}(e_i)=\frac{Dp_4}{Dx_i}(e_j)=\frac{Dp_4}{Dx_i}(e_i+e_j)=\frac{Dp_4}{Dx_i}(e_j+e_k)=0.$$

We conclude that the vectors of Hamming weight 2 indeed vanish with multiplicity 2. The two second order Hasse derivatives are

$$\begin{aligned}
\frac{D^{(2)}p_4}{Dx_i^2} &= \binom{3}{2}x_i + \binom{2}{2} + \left(\binom{3}{2}x_i + \binom{2}{2}\right)\sum_{j\neq i} x_j \\
&\equiv x_i + 1 + (x_i + 1)\sum_{j\neq i} x_j \\
\frac{D^{(2)}p_4}{Dx_iDx_j} &= (3x_i^2 + 2x_i) + (3x_j^2 + 2x_j) + 1 + \sum_{k\neq i,j} x_k \\
&\equiv x_i + x_j + 1 + \sum_{k\neq i,j} x_k.
\end{aligned}$$

Since

$$\frac{D^{(2)}p_4}{Dx_i^2}(e_i) = \frac{D^{(2)}p_4}{Dx_i^2}(e_j) = \frac{D^{(2)}p_4}{Dx_iDx_j}(e_i) = \frac{D^{(2)}p_4}{Dx_iDx_j} = \frac{D^{(2)}p_4}{Dx_iDx_j}(e_k) = 0,$$

all vectors of Hamming weight 1 are indeed roots of p_4 with multiplicity 3. Therefore f_4 is a polynomial k-cover of $\mathbb{F}_2^n$. If we apply the Ball-Serra Bound for 4-cover of $\mathbb{F}_2^n$, the lower bound on the degree of f_4 is

$$\deg(f_4) \geq (4-1) + n = n + 3.$$

So in theory, there could exist a polynomial of even lower degree than the above one that vanishes everywhere on $\mathbb{F}_2^n\backslash\{(0,\dots,0)\}$ with multiplicity 4 and that is non-zero at the origin. We will show however that the Ball-Serra Bound is not tight for 4-covers of the binary field.

❇ **Theorem 46.** ❇ *The polynomial 4-covering number of $\mathbb{F}_2^n$ is equal to $n+4$.*

Proof. Sauermann and Wigderson already provided a polynomial of degree $n+4$ that covers each point of $\mathbb{F}_2^n \setminus \{\bar{0}\}$ 4 times while avoiding the origin. Thus, proving the above theorem corresponds to proving that there is no degree $n+3$ polynomial that covers all points of $\mathbb{F}_2^n\setminus\{\bar{0}\}$ with multiplicity four and that leaves the origin uncovered. To do so, we will use the same proof method as Clifton and Huang did for Theorem 37, but now using Hasse derivatives.

In search of a contradiction, suppose that there does exist such a polynomial f, with $\deg(f) = n+3$. Then by the Punctured Combinatorial Nullstellensatz we can write f in the form

$$f = \sum_{1\leq i\leq j\leq k\leq \ell\leq n} x_i(x_i-1)x_j(x_j-1)x_k(x_k-1)x_\ell(x_\ell-1)g_{ijk\ell} + u\cdot\prod_{i=1}^{n}(x_i-1),$$

with $\deg(u) \leq \deg(f) - n = 3$. Since for $t = 0,1,2,3$, the t-th partial Hasse derivative of $x_i(x_i-1)x_j(x_j-1)x_k(x_k-1)x_\ell(x_\ell-1)$ is zero on $\mathbb{F}_2^n$, the polynomial $h := u\cdot\prod_{i=1}^n(x_i-1)$ has t-th order partial Hasse derivatives vanishing on $\mathbb{F}_2^n\setminus\{\bar{0}\}$. Using the product rules that we have just derived, we find the following expressions for the different derivatives.

$$\frac{Dh}{Dx_i} = \frac{Du}{Dx_i}\prod_{j=1}^{n}(x_j-1) + u\prod_{j\neq i}(x_j-1) \tag{3.1}$$

$$\frac{D^{(2)}h}{Dx_i^2} = \frac{D^{(2)}u}{Dx_i^2}\prod_{j=1}^{n}(x_i-1) + \frac{Du}{Dx_i}\prod_{j\neq i}(x_j-1) \tag{3.2}$$

$$\begin{aligned}\frac{D^{(2)}h}{Dx_iDx_j} &= \frac{D^{(2)}u}{Dx_iDx_j}\prod_{k=1}^{n}(x_k-1) + \frac{Du}{Dx_i}\prod_{k\neq j}(x_k-1) \\ &\quad + \frac{Du}{Dx_j}\prod_{k\neq i}(x_k-1) + u\prod_{k\neq i,j}(x_k-1)\end{aligned} \tag{3.3}$$

$$\frac{D^{(3)}h}{Dx_i^3} = \frac{D^{(3)}u}{Dx_i^3}\prod_{j=1}^{n}(x_j-1) + \frac{D^{(2)}u}{Dx_i^2}\prod_{j\neq i}(x_j-1) \tag{3.4}$$

$$\begin{aligned}\frac{D^{(3)}h}{Dx_i^2Dx_j} &= \frac{D^{(3)}u}{Dx_i^2Dx_j}\prod_{k=1}^{n}(x_k-1) + \frac{D^{(2)}u}{Dx_i^2}\prod_{k\neq j}(x_k-1) \\ &\quad + \frac{D^{(2)}u}{Dx_iDx_j}\prod_{k\neq i}(x_k-1) + \frac{Du}{Dx_i}\prod_{k\neq i,j}(x_k-1)\end{aligned} \tag{3.5}$$

$$\begin{aligned}\frac{D^{(3)}h}{Dx_iDx_jDx_k} &= \frac{D^{(3)}u}{Dx_iDx_jDx_k}\prod_{\ell=1}^{n}(x_\ell-1) + \frac{D^{(2)}}{Dx_iDx_j}\prod_{\ell\neq k}(x_\ell-1) \\ &\quad + \frac{D^{(2)}u}{Dx_iDx_k}\prod_{\ell\neq j}(x_\ell-1) + \frac{D^{(2)}u}{Dx_jDx_k}\prod_{\ell\neq i}(x_\ell-1) \\ &\quad + \frac{Du}{Dx_i}\prod_{\ell\neq j,k}(x_\ell-1) + \frac{Du}{Dx_j}\prod_{\ell\neq i,k}(x_\ell-1) \\ &\quad + \frac{Du}{Dx_k}\prod_{\ell\neq i,j}(x_\ell-1) + u\prod_{\ell\neq i,j,k}(x_\ell-1).\end{aligned} \tag{3.6}$$

Since all above polynomials have to vanish on $\mathbb{F}_2^n \setminus \{\overline{0}\}$, we evaluate its points to get additional restrictions on u.

$$\frac{Dh}{Dx_i}(e_i) = u(e_i) = 0 \tag{3.7}$$

$$\frac{D^{(2)}h}{Dx_i^2}(e_i) = \frac{Du}{Dx_i}(e_i) = 0 \tag{3.8}$$

$$\frac{D^{(2)}h}{Dx_iDx_j}(e_i) = \frac{Du}{Dx_j}(e_i) + u(e_i) = 0 \tag{3.9}$$

$$\Rightarrow \frac{Du}{Dx_j}(e_i) = 0$$

$$\frac{D^{(2)}h}{Dx_iDx_j}(e_i + e_j) = u(e_i + e_j) = 0 \tag{3.10}$$

$$\frac{D^{(3)}h}{Dx_i^3}(e_i) = \frac{D^{(2)}u}{Dx_i^2}(e_i) = 0 \tag{3.11}$$

$$\frac{D^{(3)}h}{Dx_i^2Dx_j}(e_i) = \frac{D^{(2)}u}{Dx_iDx_j}(e_i) + \frac{Du}{Dx_i}(e_i) = 0 \tag{3.12}$$

$$\Rightarrow \frac{D^{(2)}u}{Dx_iDx_j}(e_i) = 0$$

$$\frac{D^{(3)}h}{Dx_i^2Dx_j}(e_j) = \frac{D^{(2)}u}{Dx_i^2}(e_j) + \frac{Du}{Dx_i}(e_j) = 0 \tag{3.13}$$

$$\Rightarrow \frac{D^{(2)}u}{Dx_i^2}(e_j) = 0$$

$$\frac{D^{(3)}h}{Dx_i^2Dx_j}(e_i + e_j) = \frac{Du}{Dx_i}(e_i + e_j) = 0 \tag{3.14}$$

In fact, more conditions arise from the other third-order partial derivatives, but we will not need those to derive a contradiction. Since u has maximum degree 3 and does not vanish at the origin, we can assume that it takes the following form:

$$u(x) = \sum_i a_i x_i^3 + \sum_i b_i x_i^2 + \sum_{i \neq j} c_{ij} x_i^2 x_j + \sum_{i<j} d_{ij} x_i x_j + \sum_{i<j<k} e_{ijk} x_i x_j x_k + \sum_i f_i x_i + 1.$$

Taking its first order Hasse derivative:

$$\begin{aligned}
\frac{Du}{Dx_i} &= 3a_i x_i^2 + 2b_i x_i + \sum_{j \neq i} c_{ij} \cdot 2x_i x_j + \sum_{j \neq i} c_{ji} x_j^2 + \left(\sum_{j<i} d_{ji} + \sum_{j>i} d_{ij} \right) x_j \\
&\quad + \left(\sum_{j<k<i} e_{jki} + \sum_{j<i<k} e_{jik} + \sum_{i<j<k} e_{ijk} \right) x_j x_k + f_i \\
&\equiv a_i x_i + \sum_{j \neq i} c_{ji} x_j + \left(\sum_{j<i} d_{ji} + \sum_{j>i} d_{ij} \right) x_j + \left(\sum_{j<k<i} e_{jki} + \sum_{j<i<k} e_{jik} + \sum_{i<j<k} e_{ijk} \right) x_j x_k + f_i.
\end{aligned}$$

The next derivative we need is

$$\begin{aligned}
\frac{D^{(2)}u}{Dx_i^2} &= \binom{3}{2} a_i x_i + \binom{2}{2} b_i + \binom{2}{2} \sum_{j \neq i} c_{ij} x_j \\
&\equiv a_i x_i + b_i + \sum_{j \neq i} c_{ij} x_j.
\end{aligned}$$

And lastly, assuming that $i < j$, we have

$$\frac{D^{(2)}u}{Dx_iDx_j} = 2c_{ij}x_i + 2c_{ji}x_j + d_{ij} + \left(\sum_{j<k} e_{ijk} + \sum_{i<k<j} e_{ikj} + \sum_{k<i} e_{kij}\right)x_k$$
$$\equiv d_{ij} + \left(\sum_{j<k} e_{ijk} + \sum_{i<k<j} e_{ikj} + \sum_{k<i} e_{kij}\right)x_k.$$

Now we use the conditions we have derived to try to solve for the coefficients in u. Let us consider the above conditions from top to bottom. Note that in certain choices for i and j, d_{ij} should be replaced by d_{ji}. To ease notation, however, we only write d_{ij} below, as it does not change the conclusion.

$$u(e_i) = a_i + b_i + f_i + 1 = 0 \tag{3.15}$$
$$\frac{Du}{Dx_i}(e_i) = a_i + f_i = 0 \tag{3.16}$$
$$\frac{Du}{Dx_i}(e_j) = c_{ji} + d_{ij} + f_i = 0 \tag{3.17}$$
$$u(e_i + e_j) = a_i + a_j + b_i + b_j + c_{ij} + c_{ji} + d_{ij} + f_i + f_j + 1 = 0 \tag{3.18}$$
$$\frac{D^{(2)}u}{Dx_i^2}(e_i) = a_i + b_i = 0 \tag{3.19}$$
$$\frac{D^{(2)}u}{Dx_iDx_j}(e_i) = d_{ij} = 0 \tag{3.20}$$
$$\frac{D^{(2)}u}{Dx_i^2}(e_j) = b_i + c_{ij} = 0 \tag{3.21}$$
$$\frac{Du}{Dx_i}(e_i + e_j) = a_i + a_j + c_{ji} + d_{ij} + f_i = 0. \tag{3.22}$$

Combining (3.15), (3.16) and (3.19), we obtain $a_i = b_i = f_i = 1$. Moreover, (3.17) and (3.21) imply $c_{ij} = c_{ji} = 1$. Plugging all these values in the left hand side of (3.18) gives 1, while it should be zero. Hence u can indeed not exist, which concludes the proof. □

So even though the improved lower bound of Sauermann and Wigderson does not hold in the binary field, we have shown that in this specific case the Ball-Serra Bound is not tight either. Knowing the exact cover number for a 4-cover of $\mathbb{F}_2^n$, an interesting further research would be to investigate whether this result can be extended to any k-cover for $k \geq 5$. Perhaps there is a recurring relation similar to the one in the proof of Theorem 37 that allows us to lower bound the cover number of higher multiplicities.

❊ **Question 1.** ❊ *Can we find bounds on the polynomial k-covering number of $\mathbb{F}_2^n$ for $k \geq 5$, knowing that the optimal polynomial 4-cover has degree $n + 4$?*

Moreover, by allowing fields of other characteristic, it would be interesting to research whether there are finite fields for which the Ball-Serra bound is tight for a certain polynomial k-cover. When restricting the cover to hyperplanes, the question becomes even more difficult. Bishnoi et al. [9] have proven that if the hyperplane k-cover of $\mathbb{F}_2^n$ is not allowed to cover the origin at all, the problem is equivalent to finding linear binary codes of large minimum distance. This problem is well-studied and known to be difficult, showing that finding the minimum size of a hyperplane k-cover is a tough nut to crack. In the same paper, it is studied what happens if we want to cover every non-zero point $\mathbb{F}_2^n$ with multiplicity k by $(n-d)$-dimensional subspaces while the origin is covered at most $k-1$ times. We denote the minimum number of such affine subspaces required for an almost k-cover by $g(n, k, d)$. When d is equal to one, this is the hyperplane case, similar to what Sauermann and Wigderson investigated in Theorem 40 and Theorem 41. The latter did hold over any field, but specifically considers the case where the origin is covered exactly $k - 1$ times. The behaviour of $g(n, k, d)$ is given in the following theorem.

Theorem 47 (Bishnoi-Boyadzhiyska-Das-Mészáros). *Let $k \geq 1$ and $n \geq d \geq 1$. Then:*

(a) If $k \geq 2^{n-d-1}$, then $g(n,k,d) = 2^d k - \lfloor \frac{k}{2^{n-d}} \rfloor$.

(b) If $n > 2^{2^d k-d-k+1}$, then $g(n,k,d) = n + 2^d k - d - 2$.

(c) If $k \geq 2$ and $n \geq \lfloor \log_2 k \rfloor + d + 1$, then

$$n + 2^d k - d - \log_2(2k) \leq g(n,k,d) \leq n + 2^d k - d - 2.$$

We see that there is a different behaviour of $g(n,k,d)$ when k is fixed and n is large compared to when n is fixed and k is large. And, more importantly in the investigation of the Ball-Serra Bound, it shows a separation between polynomial covers and hyperplane covers of the binary field. For any $k \geq 4$, we set

$$f_k(\overline{x}) = x_1^{k-4}(x_1 - 1)^{k-4} f_4(\overline{x}),$$

where $f_4(\overline{x})$ is as defined above. Then f_k vanishes with multiplicity k on $\mathbb{F}_2^n \backslash \{\overline{x}\}$, while having multiplicity $k-4$ at the origin. Hence, the minimum degree of a polynomial cover is at most $\deg f_k = n + 2k - 4$, while Theorem 47 shows that for any $k \geq 4$ and n sufficiently large, $g(n,k,1) = n + 2k - 3$.

3.3. Covers in the Cartesian Plane

In the proof of Alon-Füredi, we saw that the hypercube is a particular setting with nice properties, because all points that have to be covered have binary entries. While the binary field still has binary entries, things became more complicated because of the different derivative that we had to consider. In this section, we leave the finite fields behind and return back to the Cartesian plane. More specifically, we focus on k-covers of planar grids $\Gamma = S_1 \times S_2 \subseteq \mathbb{R}^2$, where we assume that $(0,0) \in \Gamma$. Hyperplane covers of such grids have been studied in literature and the threshold is known for which grids the Ball-Serra Bound is tight. The first part of this section looks at these results and highlights the gaps in what is known so far. Secondly, we consider polynomial k-covers of grids in $\mathbb{R}^2$. This has not been studied in literature so far. An algorithm is presented to compute the minimum degrees of such polynomials and it is investigated whether the coefficients in optimal polynomials show a nice pattern that can be generalised. The generated curves seem to show a threshold when the Ball-Serra Bound is tight. This threshold is further investigated by considering slices of the grid and evaluating polynomials on these slices. This method suggests the same threshold, which is included as a conjecture at the end of the chapter.

3.3.1. Hyperplane Covers

In the Cartesian plane, a hyperplane simply corresponds to a line. The topic of hyperplane k-covers in the plane was investigated by Yvonne den Bakker in her thesis [3]. An intuitive approach to finding the cover number for arbitrary planar grids is to first find the optimal values for small grids with small covering number and then generalise the cover to other cases. An integer programme can be constructed that finds the hyperplane k-cover number for a specific grid. For every possible origin-avoiding line $\lambda \in \Gamma$, we introduce a variable $z(\lambda)$ that indicates how many times λ is used in our cover. Of course there are infinitely many lines that pass through the considered grid. However, we can restrict ourselves to origin-avoiding lines that pass through at least two points of Γ. If a k-cover of minimum size contains a line that passes through exactly one point of Γ, we can always replace this line by another that passes through that point and a second one, while avoiding the origin. This does not decrease the size of the k-cover. Hence, the set of to be considered lines Λ is finite. The linear programme becomes:

$$\begin{array}{llll} \text{min} & \sum\limits_{\lambda \in \Lambda} & z(\lambda) & \\ \text{subject to} & \sum\limits_{\substack{\lambda \in \Lambda \\ (x,y) \in \lambda}} & z(\lambda) \geq k & \text{for all } (x,y) \in \Gamma \setminus \{(0,0)\} \\ & & z(\lambda) \in \mathbb{Z}_{\geq 0} & \text{for all } \lambda \in \Lambda. \end{array}$$

When solving this integer programme for grids in the plane we are mainly interested in whether the returned value corresponds with the Ball-Serra Bound or whether there is a gap between the lower bound and the actual value. As mentioned at the beginning of the chapter, for $|S_1| = n$ and $|S_2| = m$, we can – and will – always assume that $n \geq m$ because of symmetry. For $\Gamma = S_1 \times S_2$, we denote the minimum size of a hyperplane k-cover of Γ by $\mathrm{cov}_k(\Gamma)$. The Ball-Serra Bound for a k-cover of $S_1 \times S_n$ then becomes

$$\mathrm{cov}_k(\Gamma) \geq (k-1)(n-1) + (n-1) + (m-1) = k(n-1) + (m-1).$$

When the integer programme is run for $k = 3$ on the grid $\{0, 1, \ldots, n-1\} \times \{0, 1, \ldots, m-1\}$, the left table in Table 3.1 emerges [3]. In this table, we see that the Ball-Serra is indeed tight in some cases, namely when n is large compared to m. This corresponds to wide, rectangular grids. In other cases the Ball-Serra Bound is not tight. Moreover, the gap between the lower bound and the optimal value increases as the grids become larger. A similar analysis can be made when increasing the cover number from 3 to 4. The minimal cover sizes are given in the right table of Table 3.1. We again see that for wide rectangular grids, the Ball-Serra Bound is tight. And again, the gap between the lower bound and the optimal value increases as the grid becomes larger. This time, the gap seems to increase even faster. One can verify that the region in both tables where Ball-Serra is tight corresponds to $n \geq (k-1)(m-1) + 1$. Indeed, in the paper following the thesis, Bishnoi et al. proved this threshold for tightness of Ball-Serra for k-covers in the plane [8].

m \ n	2	3	4	5	6	7	8
2	5	7	10	13	16	19	22
3		9	12	14	17	20	23
4			14	16	19	21	24
5				18	21	23	26
6					23	25	28
7						27	30
8							32

m \ n	2	3	4	5	6	7	8
2	6	10	13	17	21	25	29
3		12	15	19	23	26	30
4			18	21	25	28	32
5				24	27	30	34
6					30	33	36
7						36	39
8							42

0
1
2
3
4
5
6
7

Table 3.1: Tables with the hyperplane covering numbers for $\{0, 1, \ldots, n-1\} \times \{0, 1, \ldots, m-1\}$ [3]. The left table contain the covering numbers for 3-covers, the right table shows the values for 4-covers. The colours indicate the gap with Ball-Serra.

Theorem 48 (Tightness of Ball-Serra in the plane)**.** *Let $S_1, S_2 \subseteq \mathbb{R}$ have respective sizes $|S_1| = n$ and $|S_2| = m$. Assume $0 \in S_1 \cap S_2$ and let $\Gamma = S_1 \times S_2$. If for a positive integer k, we have $n \geq (k-1)(m-1)+1$,*

$$\mathrm{cov}_k(\Gamma) = k(n-1) + (m-1).$$

To prove this theorem, we must make a distinction between *interior points* and *boundary points*.

Definition 49. *Let $\Gamma = S_1 \times S_2 \subseteq \mathbb{R}^2$ be a grid such that $(0,0) \in \Gamma$. We say that a point in Γ is a boundary point if one of its coordinates is zero and that it is an interior point otherwise.*

Proof of Theorem 48. Say $S_1 = \{0, s_1, \ldots, s_{n-1}\}$ and $S_2 = \{0, t_1, \ldots, t_{m-1}\}$. We arbitrarily partition $S_1 \setminus \{0\}$ in $P_1 \cup \cdots \cup P_{m-1}$ such that $|P_j| \geq k-1$ for all j. Note that is always possible since $n-1 \geq (k-1)(m-1)$. The cover that we will be using consists of three types of lines.

1. $k-1$ copies of the line $x = s_i$ for all $i \in [n-1]$;
2. the line $y = t_j$ for all $j \in [m-1]$;
3. the line connecting $(s, 0)$ and $(0, t_j)$ for every $j \in [m-1]$ and $s \in P_j$.

There are $(k-1)(n-1)$ lines of the first type, $m-1$ lines of the second one and $n-1$ lines of the third one, as there is exactly one such line through every s_i for $i \in [n-1]$. Hence, this collection contains $k(n-1)+(m-1)$ lines. Moreover, every interior point is covered $k-1$ times by a line of type 1 and once by a line of type 2. Therefore the only points for which we should still verify whether they are covered k times are the boundary points. Let us first consider points $(s_i, 0)$ on the x-axis. Such a point is covered

$k-1$ times by a line of type 1 and once by a line of type 3. And secondly, a point on the y-axis $(0, t_j)$ is covered once by type 2 and $|P_j| \geq k-1$ times by a line of type 3. Therefore this collection of lines is a k-cover for $S_1 \times S_2$ of size $k(n-1)+(m-1)$, matching the lower bound given by Ball-Serra. □

The question arises what happens if Ball-Serra is not tight, i.e. what is the minimum size of a k-cover of an $(n \times m)$-grid when $n \leq (k-1)(m-1)$? First of all, we can find an upper bound on the cover number of an arbitrary $(n \times m)$-grid if k is divisible by $\frac{n+m-2}{\gcd(n-1,m-1)}$ [9].

Theorem 50. *Let Γ be an arbitrary $(n \times m)$-grid and suppose k is divisible by $\frac{n+m-2}{\gcd(n-1,m-1)}$. Then,*

$$\operatorname{cov}_k(\Gamma) \leq k(n-1) + \frac{k}{n+m-2}(m-1)^2.$$

To find a k-cover that gives this upper bound, we have the opportunity to use some graph theory, namely matchings of bipartite graphs. Before we look at the construction, we quickly refresh our memories on Hall's Matching Theorem and a corollary thereof [22].

Theorem 51 (Hall's Matching Theorem)**.** *A bipartite graph $G = (A, B, E)$ has a matching that covers all vertices of A if and only if*

$$|N(S)| \geq |S|, \quad \forall S \subseteq A.$$

Corollary 52. *Every non-empty regular bipartite graph contains a perfect matching.*

These two fundamental statements in graph theory will remarkably allow us to come up with a tight hyperplane cover for the grids that are specified above.

Proof of Theorem 50. Let Γ be an arbitrary $(n \times m)$-grid and k be a positive integer such that $\frac{n+m-2}{\gcd(n-1,m-1)}$ divides k. We let

$$a = \frac{n-1}{\gcd(n-1,m-1)} \text{ and } b = \frac{m-1}{\gcd(n-1,m-1)},$$

such that $a+b$ divides k. We also define the integers

$$d_1 = \frac{bk}{a+b} \text{ and } d_2 = \frac{ak}{a+b},$$

such that $d_1 + d_2 = k$. We let $G = (S_1 \backslash \{0\}, S_2 \backslash \{0\}, E)$ be a biregular bipartite graph of respective degrees d_1 and d_2. A way of constructing such a graph G is the following. First, we construct a bipartite graph $G_1 = (S_1 \backslash \{0\}, A, E_{G_1})$, where every $s \in S_1 \backslash \{0\}$ has d_1 unique neighbours in some set A of size $d_1(n-1)$. Similarly, we construct a bipartite graph $G_2 = (S_2 \backslash \{0\}, B, E_{G_2})$, such that every $t \in S_2 \backslash \{0\}$ has D_2 unique neighbours in some set B of size $d_2(m-1)$. Thus, $|A| = |B| = d$ for some integer d. We connect every $a \in A$ with every $b \in B$ to obtain one large graph H. Note that the induced subgraph of the vertices in A and B forms a complete bipartite graph $K_{d,d}$. By the corollary to Hall's Matching Theorem, this subgraph graph has a perfect matching. Consider such a perfect matching M. Now we say we have an edge $\{s, t\}$ in $E(G)$ if and only if there is a path $\{s, a, b, t\}$ in H with $\{a, b\} \in M$. Now we define our k-cover.

1. d_2 copies of the line $x = s$ for each $s \in S_1 \setminus \{0\}$;
2. d_1 copies of the line $y = t$ for each $t \in S_2 \setminus \{0\}$;
3. the line connecting $(s, 0)$ and $(0, t)$ for each $\{s, t\} \in E(G)$.

With this collection of lines, every interior point is covered d_2 times by a line of type 1 and d_1 times by a line of type 2. Since $d_1 + d_2 = k$, every interior point is indeed covered k times. A point on the x-axis $(s, 0)$ is covered d_2 times by a line of type 1 and $\deg_G(s) = d_1$ times by a line of type 3. Similarly, a point on the y-axis $(0, t)$ is covered d_1 times by a line of type 2 and $\deg_G(t) = d_2$ times by a line of type

3. So every boundary point is covered k times too. Moreover, the origin clearly remains uncovered. Hence, this collection of lines is a k-cover of size $d_2(n-1)+d_1(m-1)+d_1(n-1)$. Therefore,

$$\begin{aligned}\mathrm{cov}_k(\Gamma) &\leq d_2(n-1)+d_1(m-1)+d_1(n-1)\\ &= (d_1+d_2)(n-1)+d_1(m-1)\\ &= k(n-1)+\frac{bk}{a+b}(m-1)\\ &= k(n-1)+\frac{k}{n+m-2}(m-1)^2.\end{aligned}$$ □

Ideally, we would like to show that we can never find a cover of smaller size than the above one. Unfortunately, it's not that easy. A matching lower bound can be shown for a specific type of grid.

Definition 53. *Let $S_1, S_2 \subseteq \mathbb{R}$ and $\Gamma = S_1 \times S_2$. If any line that goes through two boundary points does not pass through any of the interior points, then we say that Γ is generic.*

Note that the vast majority of grids is generic. When sampling the points for S_1 independently and uniformly at random from a fixed interval $[s_1, s_2]$ and the points for S_2 from $[t_1, t_2]$, then the grid $S_1 \times S_2$ is generic with probability 1. But it is easier to cover a non-generic grid with few lines than a generic grid, because a cover of a generic $(n \times m)$-grid is also a cover for any non-generic grid while the other way around is not necessarily true. Therefore, a lower bound on a generic grid is not necessarily a lower bound on a non-generic grid [9].

Theorem 54. *Let $\Gamma = S_1 \times S_2 \subseteq \mathbb{R}^2$ be a generic $(n \times m)$-grid such that $(0,0) \in \Gamma$. Then,*

$$\mathrm{cov}_k(\Gamma) \geq k(n-1)+\frac{k}{n+m-2}(m-1)^2.$$

So in particular, if k is divisible by $\frac{n+m-2}{\gcd(n-1,m-1)}$ we have equality.

The proof of this theorem can be obtained by defining a solution to the dual of the linear programme given above and using weak duality. In the case where a grid is not generic the lower bound can still be increased compared to the Ball-Serra Bound [9].

Theorem 55. *Let $S_1 \subseteq \mathbb{R}$ such that $0 \in S_1$ and $|S_1| = n$. Then for $\Gamma = S_1 \times S_1$, we have for $n \to \infty$*

$$\left(10-4\sqrt{5}+o(1)\right)k(n-1) \leq \mathrm{cov}_k(\Gamma) \leq \lceil \tfrac{3}{2}k \rceil (n-1).$$

Compared to Ball-Serra, which gives the lower bound $(k+1)(n-1)$, the above theorem provides a constant factor improvement for $k \geq 18$, which shows that Ball-Serra can never be tight for hyperplane covers with high multiplicity of large square grids. However, this analysis of the Ball-Serra bound is assuming that the cover is given by lines, hence the corresponding polynomial is a product of degree one polynomials. The question arises for which grids the bound is tight when we look at polynomial k-covers.

3.3.2. Generating polynomial 3-Covers with Derivatives

Unlike the hyperplane covers in the Cartesian plane, the Ball-Serra Bound has not yet been investigated for polynomial covers. Therefore, we have to start at the very beginning and investigate polynomial covers of small grids and small multiplicities. From there on, we work our way towards a conjecture about the threshold for tightness for any k-cover. Everything in the remainder of this chapter is own work and cannot be found in literature. First, as a starting point for investigating whether there is a clear difference between polynomial covers and line covers, we are going to recompute the values in Table 3.1 for this new setting. Yet now we cannot make use anymore of a linear programme, as for a given degree there are infinitely many polynomials to be considered over the grid. Hence, a different approach is needed. At the beginning of this chapter, we have seen that we can come up with polynomial covers by using the vanishing conditions on the partial derivatives. Let us first analyse an algorithm that implements this method. Since we know that the Ball-Serra Bound is tight for $k = 2$ for a line cover, we obviously cannot beat this when we allow ourselves to use any type of polynomial.

Therefore, we start by looking at 3-covers. The most straightforward way of computing polynomials is by precisely performing the steps of Example 36. The first approach taken is to compute the polynomials with a symbolic solver in Python, namely SymPy. This package allows to set up a polynomial of certain degree, where the coefficient in front of $x^i y^j$ is a variable a_{ij}. Moreover, the package includes functions to compute partial derivatives of polynomials. Hence, we can easily construct the systems of equations that needs to be solved. The entire code can be found in Appendix B.1. This most basic approach is implemented by the function `curvemaker`. This function generates all monomials of the right degree and sets up the coefficients a_{ij}. Using the SymPy command `sp.diff()`, the partial derivatives are computed and all grid points are evaluated. SymPy also allows to solve a system of equations using `sp.solve()`. The only remaining step is to choose a value for possible free variables in the system. These are always set to 0, because this choice leads to the most compact formulas of polynomials.

Example 56. *Using the constraints on the partial derivatives, the algorithm finds a 3-cover of degree 11 for the grid* $\{0,1,2,3\} \times \{0,1,2\}$*:*

$$\begin{aligned} f(x,y) = (y-2)\cdot(y-1)\cdot(&314x^9 - 5652x^8 + 1727x^7y^2 - 5181x^7y + 44274x^7 - 21666x^6y^2 \\ &+ 64998x^6y - 197820x^6 + 111470x^5y^2 - 334410x^5y + 554838x^5 - 303324x^4y^2 \\ &+ 909972x^4y - 1011708x^4 + 4239x^3y^5 - 22608x^3y^4 + 38151x^3y^3 + 449177x^3y^2 \\ &- 1406877x^3y + 1197910x^3 - 25434x^2y^5 + 135648x^2y^4 - 228906x^2y^3 - 291078x^2y^2 \\ &+ 1229310x^2y - 887364x^2 + 46629xy^5 - 248688xy^4 + 419661xy^3 - 31086xy^2 - 559548xy \\ &+ 373032x + 216y^9 - 243y^8 - 1161y^7 - 2997y^6 - 6669y^5 + 111672y^4 - 219510y^3 \\ &+ 84780y^2 + 101736y - 67824). \end{aligned}$$

The plot of this polynomial can be found in Figure 3.1. The example shows that the Ball-Serra Bound is tight for this grid. However, when looking at its formula and plot, it is not clear how this cover can be generalised to a cover of higher degree for a larger grid. The system obtained for this cover did contain free variables, which were set to zero. Therefore, there are actually infinitely many 3-covers of this grid of degree 11, depending on the choice for the free variables.

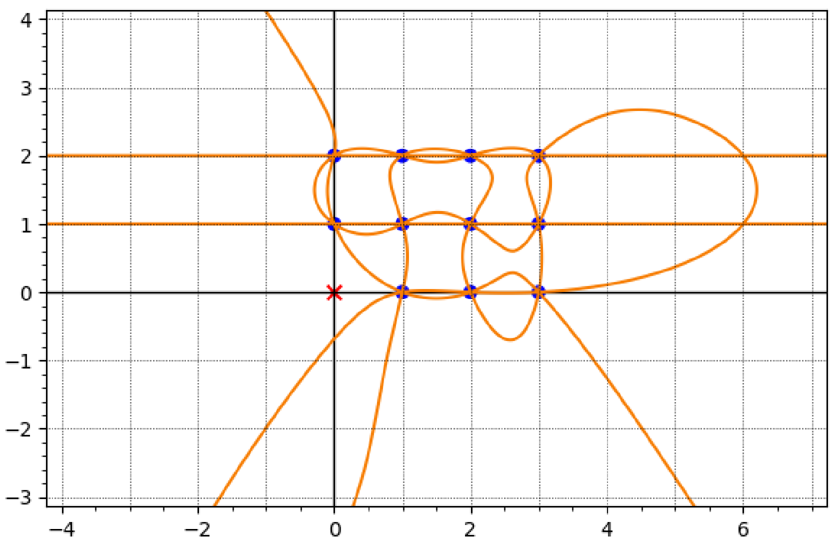


Figure 3.1: Plot of a 3-cover of $\{0,1,2,3\} \times \{0,1,2\}$ of degree 11.

As explained, we would like to reproduce Table 3.1 and thus we want to rerun the code for larger grids too. The above cover has already shown that there exist grids for which polynomial covers perform better than line covers. However, in its current form, the program is very slow and it takes a while to run it for larger grids. Luckily, there are a couple of observations that make it run faster.

Observation 57. *Consider a symmetric grid* Γ*, i.e. a grid such that if* $(a,b) \in \Gamma$*, then* $(b,a) \in \Gamma$*. For such a grid and for any* k*, there exists a* k*-cover that is symmetric in* x *and* y*.*

This observation was made by Alessandro Neri during his research visit in Delft. To see why it holds, consider an arbitrary k-cover $f(x,y)$ of Γ. Since (y,x) is also a k-cover of the same grid, we find that $f(x,y)+f(y,x)$ is a k-cover too. Clearly, this polynomial is symmetric in x and y. This enables us to decrease the number of variables in our system, since knowing a_{ij} implies knowing a_{ji}. Interestingly, keeping the constraint that the polynomial should be symmetric on the non-symmetric grids of the form $\{0,1,\dots,n-1\}\times\{0,1,\dots,m-1\}$ often also yields covers of the tight degree compared to the Ball-Serra Bound. Hence, we can first try to match the bound with this constraint added and if such a polynomial cannot be found, the slower code can be run to investigate whether a different tight cover exists. In the code in the appendix, this approach is implemented in the function `curvemaker_symmetric`. This is almost the same function as `curvemaker`, to which the constraints $a_{ij}-a_{ji}=0$ have been added.

Example 58. *Returning to the example grid* $\{0,1,2,3\}\times\{0,1,2\}$, *we can rerun the algorithm after having added the constraint that the polynomial needs to be symmetric in* x *and* y. *There indeed exists a symmetric 3-cover of degree 11 and this cover is found more quickly than the previous one. Its formula is*

$$f(x,y)=(x-2)\cdot(x-1)\cdot(y-2)\cdot(y-1)\cdot(x+y-3)\cdot(x^2-3x+y^2-3y+2)\cdot(2x^4+4x^3y - 18x^3+5x^2y^2-33x^2y+58x^2+4xy^3-33xy^2+89xy-78x+2y^4-18y^3+58y^2 - 78y+36).$$

Plotting the polynomial in Figure 3.2 clearly shows the symmetry in x *and* y.

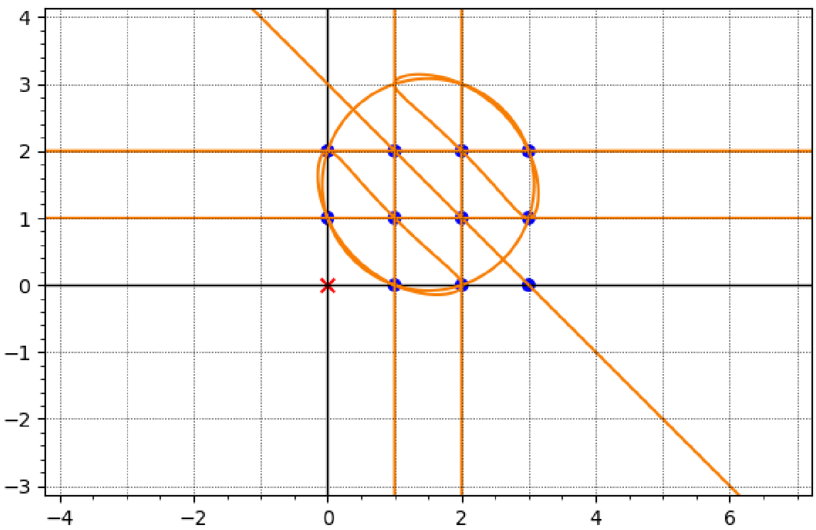


Figure 3.2: Plot of a symmetric 3-cover of $\{0,1,2,3\}\times\{0,1,2\}$ of degree 11.

The above example has an interesting property. It contains almost every horizontal and vertical line through a boundary point of the grid, only the line $x=3$ is missing. This leads to the question whether we can find tight 3-covers of $\{0,1,\dots,n-1\}\times\{0,1,\dots,m-1\}$ of the form

$$f(x,y)=\prod_{i=1}^{n-1}(x-i)\cdot\prod_{j=1}^{m-1}(y-j)\cdot p(x,y). \tag{3.23}$$

The greatest gain in terms of computation with this approach is that the interior points are already covered twice with the lines and the boundary points once. Hence, the partial derivatives of second order do not need to be computed anymore and the first order partial derivatives do not need to vanish on the interior points anymore. Furthermore, we can even investigate whether the remaining polynomial $p(x,y)$ is symmetric in x and y. That would mean that the cover is not symmetric anymore, but further reduces the computation time to find p. The respective functions in the code for these approaches are `curvemaker_lines` and `curvemaker_lines_symmetric`.

Example 59. *Returning to the grid $\{0, 1, 2, 3\} \times \{0, 1, 2\}$ we used the algorithm to investigate whether there is a 3-cover of degree 11 of the above form with p symmetric. This turned out to be the case, with the cover given by:*

$$f(x, y) = (x-3)\cdot(x-2)\cdot(x-1)\cdot(y-2)\cdot(y-1)\cdot(2x^6 - 24x^5 + 11x^4y^2 - 33x^4y + 116x^4 + 13x^3y^3 - 111x^3y^2 + 242x^3y - 288x^3 + 11x^2y^4 - 111x^2y^3 + 413x^2y^2 - 627x^2y + 386x^2 - 33xy^4 + 242xy^3 - 627xy^2 + 682xy - 264x + 2y^6 - 24y^5 + 116y^4 - 288y^3 + 386y^2 - 264y + 72).$$

The plot of this polynomial in Figure 3.3 has again a completely different form than the previous 3-covers of the same degree.

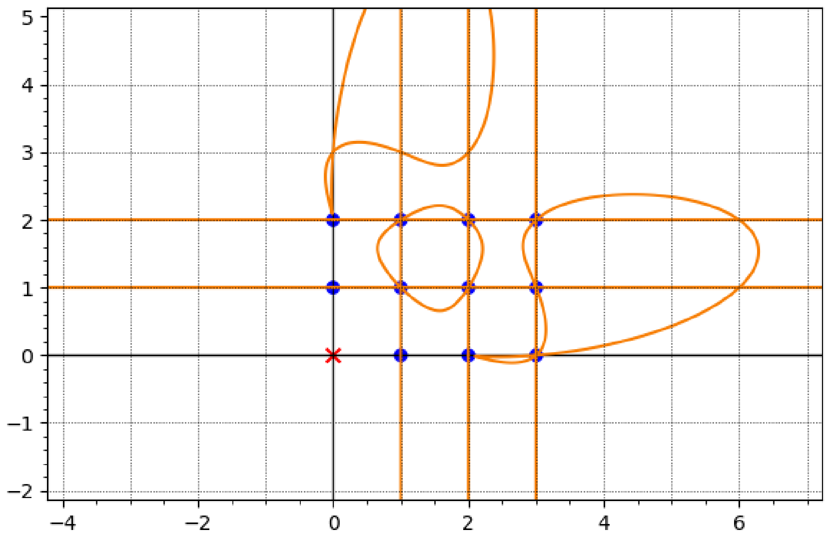


Figure 3.3: Plot of a 3-cover of $\{0, 1, 2, 3\} \times \{0, 1, 2\}$ of degree 11 that contains all horizontal and vertical lines in the grid and has a symmetric remaining factor.

While the SymPy code has proven to be useful and was able to generate a lot of covers, computing the derivatives symbolically still takes a long time. Since all covers are polynomials and the derivatives of polynomials follow rules that are easy to implement, the matrix with the linear constraints as in Example 36 can also be computed without symbolic computations, but using Numpy. The disadvantage of this is that Numpy performs Gaussian Elimination numerically. Hence, the coefficients in the polynomials obtained with Numpy are not exact, meaning that the obtained polynomials do not actually vanish on the grid points. This can be resolved using SageMath. A matrix in Sage can be defined over a certain domain. Since the polynomials vanish on integer points, the coefficients of the polynomial should be rationals. So, setting the domain of the matrix to the rational field resolves this issue. The Sage code can be found in Appendix B.2. This code in Sage is much faster than its Python counterpart. The code also returns the exact same covers as the SymPy code. A second advantage of this approach using matrices instead of symbolic computations is that when no cover of a tight degree can be found, the linear dependencies between the rows of the matrix can be more easily investigated. With this improved code, Table 3.1 can be easily remade for polynomial 3-covers. The minimum degrees of polynomial 3-covers for $\{0, 1, \dots, n-1\} \times \{0, 1, \dots, m-1\}$ can be found in Table 3.2. Moreover, all found curves are plotted in Appendix A.1.

The table shows clear differences between the minimum degree of line and polynomial covers of multiplicity 3. In the former case, we proved tightness with Ball-Serra whenever $n \geq 2m-1$. This table seems to suggest that Ball-Serra is tight for polynomial 3-cover whenever $n \geq m+1$. Furthermore, for all considered grids, there is always a construction that contains all horizontal and vertical lines and that has a symmetric remaining component. This is clearly visible in the plots in the appendix. The presence of these lines is intriguing because it implies that if such a tight construction exist for the

m \ n	2	3	4	5	6	7	8
2	5	7	10	13	16	19	22
3		9	11	14	17	20	23
4			13	15	18	21	24
5				17	19	22	25
6					21	23	26
7						25	27
8							29

0
1

Table 3.2: Minimum degree of a polynomial 3-cover for the grids $\{0, 1, \dots, n-1\} \times \{0, 1, \dots, m-1\}$.

$(n \times (n-1))$ grid, then Ball-Serra is tight for any other, more rectangular $(n \times m)$-grid where $m \leq n-1$. A tight construction can be obtained by removing the redundant horizontal lines from this first construction. We make this insight more formal in the next section. Additionally, for the square grid, Ball-Serra does not seem tight. In the case of line covers however, the gap between the optimal degree of the cover and the lower bound increased when n and m increase in such a way that $n-m$ remains equal. For polynomial covers the gap seems to only depend on the value of $n-m$. This seems to imply a completely different regime for line and polynomial covers. Interestingly, for the degree one higher than Ball-Serra and the square grids in Table 3.2, there always exist a polynomial 3-cover that contains each non-zero horizontal and vertical line of the grid once and that has a symmetric remaining factor. This suggests that for any grid Γ, if there exists a 3-cover of degree d, there exists a 3-cover of the same degree that contains all non-zero vertical and lines in Γ.

3.3.3. Symmetric Slices of 3-covers

The clear advantage of the above observation is that for 3-covers, it allows us to focus on $(n \times (n-1))$-grids and square grids. To prove tightness of the Ball-Serra Bound, it would suffice to come up with a method to predict the terms of covers of the $(n \times (n-1))$-grids. For the less square grids, we can then remove the redundant horizontal lines to still obtain a tight polynomial cover.

❇ **Lemma 60.** ❇ *Let $S_1, S_2 \subseteq \mathbb{R}$ be sets of respective sizes n and m that both contain 0. Set $\Gamma = S_1 \times S_2$ and consider a sub-grid $\Gamma' = S_1 \times S_2'$ with $S_2' \subseteq S_2$ such that $0 \in S_2'$. If there is a polynomial k-cover of Γ for some natural number k that is tight with respect to the Ball-Serra bound and the contains every horizontal line of the form $y = s$ for $s \in S_2 \setminus \{0\}$ at least once, then there is also a tight cover of Γ'.*

Proof. Suppose there is a polynomial f of degree $k(n-1) + (m-1)$ that gives a k-cover of Γ that contains every horizontal line of the form $y = s$ for $s \in S_2 \setminus \{0\}$ at least once. Then we write

$$f(x,y) = p(x,y) \cdot \prod_{s \in S_2 \setminus \{0\}} (y - s),$$

where $\deg p = k(n-1)$ and p covers the points on the slice $y = 0$ at least k times and the points at the slices $y = s$ at least $k-1$ times for all $s \in S_2 \setminus \{0\}$. Hence,

$$g(x,y) = p(x,y) \cdot \prod_{t \in S_2' \setminus \{0\}} (y - t)$$

has degree $k(n-1) + (|S_2'| - 1)$. Since $S_2' \subseteq S_2$, p still covers all points of Γ' on the slice $y = 0$ at least k times and the points at the slices $y = t$ at least $k-1$ times for all $t \in S_2' \setminus \{0\}$. Thus, g is a tight polynomial k-cover of Γ'. □

Unfortunately, coming up with a polynomial 3-cover of an arbitrary $(n \times (n-1))$-grid is no easy feat. For instance, let us consider the tight polynomial 3-covers with symmetric remaining factors of the (4×3)-grid and the (5×4)-grid:

$$f_{4\times3}(x,y) = (x-3)\cdot(x-2)\cdot(x-1)\cdot(y-2)\cdot(y-1)\cdot(2x^6 - 24x^5 + 11x^4y^2 - 33x^4y + 116x^4 + 13x^3y^3 - 111x^3y^2 + 242x^3y - 288x^3 + 11x^2y^4 - 111x^2y^3 + 413x^2y^2 - 627x^2y + 386x^2 - 33xy^4 + 242xy^3 - 627xy^2 + 682xy - 264x + 2y^6 24y^5 + 116y^4 - 288y^3 + 386y^2 - 264y + 72)$$

$$f_{5\times4}(x,y) = (x-4)\cdot(x-3)\cdot(x-2)\cdot(x-1)\cdot(y-3)\cdot(y-2)\cdot(y-1)\cdot(3x^8 - 60x^7 + 510x^6 + 25x^5y^3 - 150x^5y^2 + 275x^5y - 2400x^5 + 28x^4y^4 - 430x^4y^3 + 1880x^4y^2 - 3050x^4y + 6819x^4 + 25x^3y^5 - 430x^3y^4 + 2950x^3y^3 - 9200x^3y^2 + 12625x^3y - 11940x^3 - 150x^2y^5 + 1880x^2y^4 - 9200x^2y^3 + 21700x^2y^2 - 24250x^2y + 12540x^2 + 275xy^5 - 3050xy^4 + 12625xy^3 - 24250xy^2 + 21600xy - 7200x + 3y^8 - 60y^7 + 510y^6 - 2400y^5 + 6819y^4 - 11940y^3 + 12540y^2 - 7200y + 1728).$$

Comparing the two formulas, there is no clear way of linking the coefficients of both polynomials. Since the second polynomial has more terms than the first, it is in the first place unclear which terms we should compare. Is there a sequence to be found in the terms of highest degree? Or should we try to relate the coefficients of the same monomials in the polynomials? Looking further into the formulas of larger grids does not seem to show a clear pattern in any coefficients. Of course, we could investigate other 3-covers of the grids to see whether a pattern emerges there. But after having examined the four different types of covers as explained in Section 3.3.2, no clear way of easily predicting the coefficients in a tight 3-cover of an arbitrary $(n\times(n-1))$-grid could be found. When investigating the non-tight degree 13 cover of $\{0,1,2,3\}\times\{0,1,2,3\}$, there is an interesting pattern that can be found in a maybe less obvious place. The formula of the 3-cover of the (4×4)-grid with lines and symmetric other component is

$$f_{4\times4} = (x-3)\cdot(x-2)\cdot(x-1)\cdot(y-3)\cdot(y-2)\cdot(y-1)\cdot(6x^6 - 72x^5 - 11x^4y^3 + 66x^4y^2 - 121x^4y + 348x^4 - 11x^3y^4 + 138x^3y^3 - 553x^3y^2 + 858x^3y - 864x^3 + 66x^2y^4 - 553x^2y^3 + 1668x^2y^2 - 2123x^2y + 1158x^2 - 121xy^4 + 858xy^3 - 2123xy^2 + 2178xy - 792x + 6y^6 - 72y^5 + 348y^4 - 864y^3 + 1158y^2 - 792y + 216).$$

Although at first there might not seem to be a pattern between the coefficients of the symmetric parts of $f_{4\times3}$ and $f_{4\times4}$, the plots of these polynomials reveal a surprising property. The first polynomial can be found in Figure 3.3 and the second one in Figure 3.4.
Just as the coefficients that do not show a clear connection between the two polynomials, the shapes of the polynomials are not similar either. Yet in both cases, the horizontal *slices* of the symmetric factor of the polynomial yield a pattern. To investigate this, we keep the notation of Equation (3.23) and let $p_{4\times3}$ and $p_{4\times4}$ denote the respective symmetric and irreducible parts of our covers. The horizontal slices of these polynomials are univariate polynomials that can be obtained by evaluating the y-values in the grid. In particular, the slices of $p_{4\times3}$ are

$$p_{4\times3}(x,0) = (x-1)^2\cdot(x-2)^2\cdot(x-3)^2$$
$$p_{4\times3}(x,1) = p_{4\times3}(x,2) = x^2\cdot(x-1)\cdot(x-2)\cdot(x-3)\cdot(x-6).$$

Note that the slice that coincides with the x-axis has the maximal degree possible for $f_{4\times3}$ to be tight with respect to the Ball-Serra Bound. Also for the cover of the (4×4)-grid we can compute the slices:

$$p_{4\times4}(x,0) = (x-1)^2\cdot(x-2)^2\cdot(x-3)^2$$
$$p_{4\times4}(x,1) = p_{4\times3}(x,2) = p_{4\times3}(x,3) = x^2\cdot(x-1)\cdot(x-2)\cdot(x-3)\cdot(x-6).$$

These slices are equal for both covers. Furthermore, we see that in the slices at $y=1$, $y=2$ and $y=3$, there is an additional factor $(x-6)$. The question arises whether this factor is unique and if so, what it is determined by. It will turn out that it is not completely unique, but that there are only specific choices for this factor. Predicting the form of polynomial covers on the different slices is an interesting approach because of the following observation.

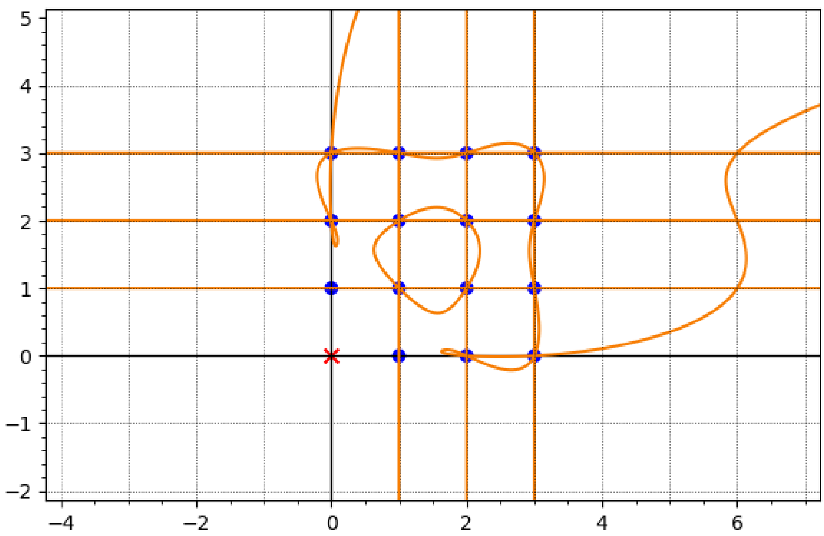


Figure 3.4: Plot of a 3-cover of $\{0, 1, 2, 3\} \times \{0, 1, 2, 3\}$ of degree 13 that contains all horizontal and vertical lines in the grid and has a symmetric remaining factor. This polynomial is not tight with respect to the Ball-Serra Bound.

❊ **Observation 61.** ❊ *Consider an arbitrary polynomial $p(x, y) \in \mathbb{R}[x, y]$ and an arbitrary point $(s, t) \in \mathbb{R}^2$. Then,*

$$\left.\frac{\partial p(x,y)}{\partial x}\right|_{(x,y)=(s,t)} = \left.\frac{dp(x,t)}{dx}\right|_{x=s}.$$

Proof. It is important to note that p is a polynomial. Hence, it is very well-behaved and both derivatives always exist. Using the definitions of partial derivatives and univariate derivatives, we find

$$\left.\frac{\partial p(x,y)}{\partial x}\right|_{(x,y)=(s,t)} = \lim_{h\to 0} \frac{p(s+h,t)-p(s,t)}{h}$$
$$\left.\frac{dp(x,t)}{dx}\right|_{x=s} = \lim_{h\to 0} \frac{p(s+h,t)-p(s,t)}{h}.$$

Therefore, they are indeed equal. □

Of course, a similar result holds for the partial derivative with respect to y. Moreover, by taking the corresponding limits, the result can be generalised to partial derivatives of higher order that are all with respect to x or all with respect to y. For $k \geq 3$,

$$\left.\frac{\partial^k p(x,y)}{\partial x^k}\right|_{(x,y)=(s,t)} = \left.\frac{d^k p(x,t)}{dx^k}\right|_{x=s}.$$

Hence, (s, t) is a root of multiplicity two of $p(x, y)$ if and only if s is a root of multiplicity two of $p(x, t)$ and t is a root of multiplicity two of $p(s, y)$. For roots with higher multiplicity $k \geq 3$, only one direction of the statement follows, since we also have to take into account partial derivatives with respect to x and y combined. If (s, t) is a root of multiplicity k of $p(x, y)$, then s is a root of multiplicity k of $p(x, t)$ and t is a root of multiplicity k of $p(s, y)$.
Let us see how we can use these slices to come up with a 3-cover with a symmetric component, without relying on partial derivatives but by using slices.

Example 62. *Consider the grid $\{0, 1, 2, 3\} \times \{0, 1, 2\}$ and suppose we want to find a 3-cover of degree 11. This time we do not use partial derivatives to compute the cover, but slices. To start, based on the observations we previously made, we assume that the cover contains all non-zero horizontal and vertical lines in the grid. Hence, we have simplified to problem from finding a 3-cover of degree 11* *to a polynomial p of degree 6 that covers every boundary point twice and every interior point once. Moreover, we want p to be symmetric in x and y to simplify the calculations. We write p in the following form:*

$$p(x,y) = a_6(y)x^6 + a_5(y)x^5 + a_4(y)x^4 + a_3(y)x^3 + a_2(y)x^2 + a_1(y)x + a_0(y),$$

where $a_i(y)$ is a univariate polynomial of maximum degree $6 - i$ for all $0 \leq i \leq 6$. Because of the vanishing conditions of p, we want that

$$\begin{aligned} p(x,0) &= (x-1)^2 \cdot (x-2)^2 \cdot (x-3)^2 \\ &= x^6 - 12x^5 + 58x^4 - 144x^3 + 193x^2 - 132x + 36. \end{aligned}$$

Since this univariate polynomial already has degree 6, there cannot be another factor in this slice. For the other slices at $y = 1$ and $y = 2$, however, we add a factor of degree 1:

$$\begin{aligned} p(x,1) &= x^2 \cdot (x-1) \cdot (x-2) \cdot (x-3) \cdot (a_1x + a_0) \\ &= a_1x^6 + (-6a_1 + a_0)x^5 + (11a_1 - 6a_0)x^4 + (-6a_1 + 11a_0)x^3 - 6a_0x^2 \\ p(x,2) &= x^2 \cdot (x-1) \cdot (x-2) \cdot (x-3) \cdot (b_1x + b_0) \\ &= b_1x^6 + (-6b_1 + b_0)x^5 + (11b_1 - 6b_0)x^4 + (-6b_1 + 11b_0)x^3 - 6b_0x^2 \end{aligned}$$

The formula of $a_6(y)$ can easily be found, since it is a degree 0 polynomial. To satisfy that the coefficient of x^6 in $p(x,0)$ equals 1, we obtain $a_6(y) = 1$. Note that this also implies $a_1 = b_1 = 1$, because otherwise we would have a contradiction.
For $a_5(y)$, we know that it should be of the form

$$a_5(y) = v_0^{(5)} + v_1^{(5)}y,$$

for some $v_0^{(5)}, v_1^{(5)} \in \mathbb{R}$. We want $a_5(y)$ to be equal to the coefficient of the corresponding monomials in the slices for $y = 0, 1, 2$. Hence, this yields the system of equations

$$\begin{aligned} v_0^{(5)} &= -12 \\ v_0^{(5)} + v_1^{(5)} &= -6a_1 + a_0 = a_0 - 6 \\ v_0^{(5)} + 2v_1^{(5)} &= -6b_1 + b_0 = b_0 - 6. \end{aligned}$$

This system only has a solution if $b_0 = 6 + 2a_0$. One such choice is $a_0 = b_0 = -6$, which shows why the factor $(x-6)$ appeared in the slices of the example above. For this choice of a_0 and b_0 we find that $a_5(y) = -12$. The slices at $y = 1$ and $y = 2$ become

$$p(x,1) = p(x,2) = x^6 - 12x^5 + 47x^4 - 72x^3 + 36x^2.$$

Knowing the coefficients at the three slices, we proceed to solve for

$$a_4(y) = v_0^{(4)} + v_1^{(4)}y + v_2^{(4)}y^2.$$

The system of equations becomes

$$\begin{aligned} v_0^{(4)} &= 58 \\ v_0^{(4)} + v_1^{(4)} + v_2^{(4)} &= 47 \\ v_0^{(4)} + 2v_1^{(4)} + 4v_2^{(4)} &= 47. \end{aligned}$$

This has the unique solution $v_0^{(4)} = 58$, $v_1^{(4)} = -33/2$ and $v_2^{(4)} = 11/2$. Hence,

$$a_4(y) = 58 - \frac{33}{2}y + \frac{11}{2}y^2.$$

For $a_3(y)$, we again get one more variable:

$$a_3(y) = v_0^{(3)} + v_1^{(3)} y + v_2^{(3)} y^2 + v_3^{(3)} y^3.$$

The system of equations after plugging in $y = 0$, $y = 1$ and $y = 2$, written in matrix form is

$$\left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & -144 \\ 1 & 1 & 1 & 1 & -72 \\ 1 & 2 & 4 & 8 & -72 \end{array}\right].$$

Solving the system yields one free variable, such that

$$a_3(y) = -144 + \left(108 + 2v_3^{(3)}\right) y + \left(-36 - 3v_3^{(3)}\right) y^2 + v_3^{(3)} y^3.$$

We proceed in the exact same way to find expressions for $a_2(y)$ and $a_1(y)$. Let

$$a_2(y) = v_0^{(2)} + v_1^{(2)} y + v_2^{(2)} y^2 + v_3^{(2)} y^3 + v_4^{(2)} y^4,$$

to obtain the system of equations

$$\left[\begin{array}{ccccc|c} 1 & 0 & 0 & 0 & 0 & 193 \\ 1 & 1 & 1 & 1 & 1 & 36 \\ 1 & 2 & 4 & 8 & 16 & 36 \end{array}\right].$$

Solving yields

$$a_2(y) = 193 + \left(-\frac{471}{2} + 2v_3^{(2)} + 6v_4^{(2)}\right) y + \left(\frac{157}{2} - 3v_3^{(2)} - 7v_4^{(2)}\right) y^2 + v_3^{(2)} y^3 + v_4^{(2)} y^4.$$

The last system of equations we will solve corresponds to

$$a_1(y) = v_0^{(1)} + v_1^{(1)} y + v_2^{(1)} y^2 + v_3^{(1)} y^3 + v_4^{(1)} y^4 + v_5^{(1)} y^5,$$

and is equal to

$$\left[\begin{array}{cccccc|c} 1 & 0 & 0 & 0 & 0 & 0 & -132 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 4 & 8 & 16 & 32 & 0 \end{array}\right].$$

Its solution gives the expression

$$\begin{aligned} a_1(y) = &- 132 + \left(198 + 2v_3^{(1)} + 6v_4^{(1)} + 14v_5^{(1)}\right) y + \left(-66 - 3v_3^{(1)} - 7v_4^{(1)} - 15v_5^{(1)}\right) y^2 \\ &+ v_3^{(1)} y^3 + v_4^{(1)} y^4 + v_5^{(1)} y^5. \end{aligned}$$

We should still find an expression for $a_0(y) = p(0, y)$. Since we assumed p to be symmetric, we use $p(0, y) = p(y, 0)$ to find

$$a_0(y) = y^6 - 6y^5 + 4y^4 + 42y^3 - 113y^2 + 108y - 36.$$

Note that because of this choice we indeed have $a_0(0) = 36$, $a_0(1) = 0$ and $a_0(2) = 0$, as required. Now we put everything together to find an expression for p:

$$\begin{aligned} p(x, y) =& x^6 - 12x^5 + x^4 \left(58 - \frac{33}{2} y + \frac{11}{2} y^2\right) \\ &+ x^3 \left(-144 + \left(108 + 2v_3^{(3)}\right) y + \left(-36 - 3v_3^{(3)}\right) y^2 + v_3^{(3)} y^3\right) \\ &+ x^2 \left(193 + \left(-\frac{471}{2} + 2v_3^{(2)} + 6v_4^{(2)}\right) y + \left(\frac{157}{2} - 3v_3^{(2)} - 7v_4^{(2)}\right) y^2 + v_3^{(2)} y^3 + v_4^{(2)} y^4\right) \\ &+ x \left(-132 + \left(198 + 2v_3^{(1)} + 6v_4^{(1)} + 14v_5^{(1)}\right) y + \left(-66 - 3v_3^{(1)} - 7v_4^{(1)} - 15v_5^{(1)}\right) y^2 \right. \\ &\qquad \left. + v_3^{(1)} y^3 + v_4^{(1)} y^4 + v_5^{(1)} y^5\right) \\ &+ y^6 - 6y^5 + 4y^4 + 42y^3 - 113y^2 + 108y - 36. \end{aligned}$$

We impose that p is symmetric, such that the number of free variables gets reduced. For instance, the symmetry implies that

$$\begin{aligned} v_5^{(1)} &= 0 \\ v_4^{(1)} &= -\frac{33}{2} \\ v_4^{(2)} &= \frac{11}{2} \\ v_3^{(1)} &= 108 + 2v_3^{(3)} \\ v_3^{(2)} &= -36 - 3v_3^{(3)}. \end{aligned}$$

Thus, p becomes

$$\begin{aligned} p(x,y) = {} & x^6 + y^6 - 12x^5 - 12y^5 + 58x^4 + 58y^4 - 144x^3 - 144y^3 + 193x^2 + 193y^2 - 132x - 132y \\ & + 36 - \frac{33}{2}x^4y - \frac{33}{2}xy^4 + \frac{11}{2}x^4y^2 + \frac{11}{2}x^2y^4 + \left(108 + 2v_3^{(3)}\right)x^3y + \left(108 + 2v_3^{(3)}\right)xy^3 \\ & + \left(-36 - 3v_3^{(3)}\right)x^3y^2 + \left(-36 - 3v_3^{(3)}\right)x^2y^3 + \left(-\frac{471}{2} + 2v_3^{(2)} + 6v_4^{(2)}\right)x^2y \\ & + \left(-66 - 3v_3^{(1)} - 7v_4^{(1)}\right)xy^2 + v_3^{(3)}x^3y^3 + \left(\frac{157}{2} - 3v_3^{(2)} - 7v_4^{(2)}\right)x^2y^2 \\ & + \left(198 + 2v_3^{(1)} + 6v_4^{(1)}\right)xy. \end{aligned}$$

We rewrite all free variables in function of $v_3^{(3)}$:

$$\begin{aligned} -\frac{471}{2} + 2v_3^{(2)} + 6v_4^{(2)} &= -\frac{549}{2} - 6v_3^{(3)} \\ -66 - 3v_3^{(1)} - 7v_4^{(1)} &= -\frac{549}{2} - 6v_3^{(3)} \\ \frac{157}{2} - 3v_3^{(2)} - 7v_4^{(2)} &= 148 + 9v_3^{(3)} \\ 198 + 2v_3^{(1)} + 6v_4^{(1)} &= 315 + 4v_3^{(3)}. \end{aligned}$$

Substituting this yields

$$\begin{aligned} p(x,y) = {} & x^6 + y^6 - 12x^5 - 12y^5 + 58x^4 + 58y^4 - 144x^3 - 144y^3 + 193x^2 + 193y^2 - 132x - 132y \\ & + 36 - \frac{33}{2}x^4y - \frac{33}{2}xy^4 + \frac{11}{2}x^4y^2 + \frac{11}{2}x^2y^4 + \left(108 + 2v_3^{(3)}\right)x^3y + \left(108 + 2v_3^{(3)}\right)xy^3 \\ & + \left(-36 - 3v_3^{(3)}\right)x^3y^2 + \left(-36 - 3v_3^{(3)}\right)x^2y^3 + \left(-\frac{549}{2} - 6v_3^{(3)}\right)x^2y \\ & + \left(-\frac{549}{2} - 6v_3^{(3)}\right)xy^2 + v_3^{(3)}x^3y^3 + \left(148 + 9v_3^{(3)}\right)x^2y^2 \\ & + \left(315 + 4v_3^{(3)}\right)xy. \end{aligned}$$

For any choice of $v_3^{(3)}$, p is a symmetric polynomial with the slices

$$\begin{aligned} p(x,0) &= (x-1)^2 \cdot (x-2)^2 \cdot (x-3)^2 \\ p(x,1) = p(x,2) &= x^2 \cdot (x-1) \cdot (x-2) \cdot (x-3) \cdot (x-6) \\ p(0,y) &= (y-1)^2 \cdot (y-2)^2 \cdot (y-3)^2 \\ p(1,y) = p(2,y) &= y^2 \cdot (y-1) \cdot (y-2) \cdot (y-3) \cdot (y-6). \end{aligned}$$

This means that all points of $\{0,1,2,3\}\times\{0,1,2\}$ *are roots of* p. *Moreover, all boundary points except* $(3,0)$ *are zeroes of multiplicity two. Lastly, to ensure that* $(3,0)$ *also has this multiplicity, we need that* $p(3,y)$ *is of the form*

$$p(3,y) = y^2 \cdot r(y),$$

for some polynomial $r(y)$. *If we consider this slice, we obtain*

$$p(3,y) = \left(y^5 - 12y^4 + 58y^3 + \left(-144 - 21v_3^{(3)}\right)\right)y^2 + \left(175 - 18v_3^{(3)}\right)y + 12v_3^{(3)} - 78\Big)y$$

Hence, we need to set $v_3^{(3)} = \frac{78}{12} = 13$ *to get the final expression for* $p(x,y)$. *Adding each horizontal and vertical line of the grid once yields the final 3-cover with formula*

$$\begin{aligned} f(x,y) = &= (x-3)\cdot(x-2)\cdot(x-1)\cdot(y-2)\cdot(y-1)\cdot(x^6+y^6-12x^5-12y^5+58x^4+58y^4 \\ &- 144x^3 - 144y^3 + 193x^2 + 193y^2 - 132x - 132y - \frac{33}{2}x^4y - \frac{33}{2}xy^4 + \frac{11}{2}x^4y^2 \\ &+ \frac{11}{2}x^2y^4 + 121x^3y + 121xy^3 - \frac{111}{2}x^3y^2 - \frac{111}{2}x^2y^3 - \frac{627}{2}x^2y - \frac{627}{2}xy^2 + \frac{13}{2}x^3y^3 \\ &+ \frac{413}{2}x^2y^2 + 314xy + 36). \end{aligned}$$

Note that this cover is the same cover that we had already found in Example 59.

The above example shows that we do not have to use partial derivatives to find a cover, we can also focus on the slices of the grid. Even though this approach might seem to be at least as complicated, mimicking the steps in the example appears to give a promising method in the search of proving that for any grid $\Gamma = S_1 \times S_2$, of respective sizes $n \geq 3$ and m such that $S_2 \subsetneq S_1$ and $(0,0) \in \Gamma$, there exists a polynomial f of degree $3(n-1)+(m-1)$ that covers every non-zero point of Γ three times while avoiding the origin. All polynomial covers found so far suggest that f can be written in the form

$$f(x,y) = \prod_{s\in S_1\setminus\{0\}} (x-s) \cdot \prod_{t\in S_2\setminus\{0\}} (y-t)\cdot p(x,y),$$

where $p(x,y)$ is symmetric in x and y.
As said, we follow the steps from Example 62. At first, we consider the slices at $y=0$ and $y=s$ for all $s\in S_2\setminus\{0\}$. As we have seen in the example, there is a restriction on the form of the latter type of slices, since certain coefficients of the two types of slices have to be equal. Hence, we first formulate a lemma that we will use to *match the first* ℓ *coefficients* of both slices, where ℓ is some integer. That is, we want to ensure that the coefficients of the first ℓ monomials when ordered in decreasing degree are equal.

❇ **Lemma 63.** ❇ *Let* ℓ, k *be two integers such that* $\ell \leq k$. *Let* $h(x)$ *and* $q(x)$ *be arbitrary univariate polynomials of respective degrees* k *and* $k-\ell$. *Then there exists a univariate polynomial* $c(x)$ *of degree* ℓ *such that the first* $\ell+1$ *coefficients of* $h(x)$ *and* $q(x)\cdot c(x)$ *are matched.*

Proof. Let us consider the formulas of $h(x)$ and $q(x)\cdot c(x)$:

$$\begin{aligned} h(x) &= a_kx^k + a_{k-1}x^{k-1} + \cdots + a_1x + a_0 \\ q(x)\cdot c(x) &= (b_{k-\ell}x^{k-\ell} + \cdots + b_1x + b_0)\cdot(\beta_\ell x^\ell + \cdots + \beta_1 x + \beta_0), \end{aligned}$$

where all coefficients of $h(x)$ and $q(x)$ are known, while the β_i are to be chosen. To ensure that the first $\ell+1$ coefficients of $h(x)$ and $q(x)\cdot c(x)$ are equal, we should look into how the monomials of highest degrees are obtained in $q(x)\cdot c(x)$. For instance, the only term of degree k in this product is given by $b_{k-\ell}x^{k-\ell}\beta_\ell x^\ell$. Since this has to be equal to a_k, we have $\beta_\ell = \frac{a_k}{b_{k-\ell}}$. In general, for $0 \leq j \leq \ell$, we need to solve the linear equation

$$a_{k-j} = \sum_{i=0}^{j} \beta_{\ell-j+i} b_{k-\ell-i}.$$

As we have just seen, for $j = 0$ this can be immediately solved. Now suppose we have solved the equations for $j < m$ for some $m < \ell$. Then for $j = m$, we need to solve

$$a_{k-m} = \beta_{\ell-m} b_{k-\ell} + \sum_{i=1}^{j} \beta_{\ell-j+i} b_{k-\ell-i}.$$

It follows that

$$\beta_{\ell-m} = \frac{a_{k-m} - \sum_{i=1}^{j} \beta_{\ell-j+i} b_{k-\ell-i}}{b_{k-\ell}}.$$

Therefore, these equations can be solved sequentially, giving all values of the coefficients in $c(x)$. Once we have found the value of β_0, we have matched terms of degree $k - \ell$, thus the first $\ell + 1$ terms of both polynomials are equal, as required. □

Note that in the above lemma we could have replaced *the first* $\ell + 1$ terms by *any* $\ell + 1$ terms. For any set S consisting of to be matched terms we first solve for the term of highest degree in S and again go sequentially through the set.
Having proven the lemma, we proceed with our search for the symmetric polynomial p.
We write $S_1 = \{0, s_1, \dots, s_{n-1}\}$ and $S_2 = \{0, s_1, \dots, s_{n-2}\}$. The polynomial that we are looking for has degree $2(n-1)$, so we express it as

$$p(x,y) = a_{2(n-1)}(y) \cdot x^{2(n-1)} + a_{2(n-1)-1}(y) \cdot x^{2(n-1)-1} + \cdots + a_1(y) \cdot x + a_0(y),$$

where $a_i(y)$ are polynomials of degree $2(n-1) - i$ for $i \in \{0, \dots, 2(n-1)\}$. Since p needs to vanish twice on the boundary points of Γ, we want

$$p(x,0) = \prod_{s \in S_1 \setminus \{0\}} (x - s)^2. \tag{3.24}$$

Similarly, for p to vanish once on the interior points and twice on the boundary points on the y-axis, we impose

$$p(x, s_i) = x^2 \prod_{s \in S_1 \setminus \{0\}} (x - s) \cdot c_i(x), \tag{3.25}$$

with polynomials $c_i(x)$ for $i \in [n-2]$. To satisfy the degree constraint, $\deg c_i \leq n - 3$. Note that $c_i(x)$ does exist since $n \geq 3$. For every i, we mimic the above example and pick the same $c(x)$, thus

$$p(x, s_1) = p(x, s_2) = \cdots = p(x, s_{n-2}).$$

The $c(x)$ we choose is given by Lemma 63 with $\ell = n - 3$ such that the first $n - 2$ terms of $p(x, 0)$ are equal to the first $n - 2$ terms of $p(x, s_1)$. For such a univariate polynomial $p(x, s)$, we let $p(x, s)_i$ denote the coefficient of the term of degree i. Now we should determine $a_0(y), \dots, a_{2(n-1)}(y)$ that satisfy condition (3.24) and the $n - 2$ conditions (3.25). To find their expressions, we split the $a_i(y)$ into three categories.

1. *$a_i(y)$ of low degree*

   We start by looking into the first $n - 2$ terms in the representation of $p(x, y)$. Clearly, condition (3.24) imposes that

   $$a_{2(n-1)}(y) = 1.$$

   Secondly, we have

   $$a_{2(n-1)-1}(y) = v_0^{(2(n-1)-1)} + v_1^{(2(n-1)-1)} y.$$

Plugging in the different values for y for conditions (3.25):

$$\begin{aligned} v_0^{(2(n-1)-1)} + s_1 v_1^{(2(n-1)-1)} &= p(x,s_1)_{2(n-1)-1} \\ v_0^{(2(n-1)-1)} + s_2 v_1^{(2(n-1)-1)} &= p(x,s_1)_{2(n-1)-1} \\ &\vdots \\ v_0^{(2(n-1)-1)} + s_{n-2} v_1^{(2(n-1)-1)} &= p(x,s_1)_{2(n-1)-1}. \end{aligned}$$

The solution to this system is $v_0^{(2(n-1)-1)} = p(x,s_1)_{2(n-1)-1}$ and $v_1^{(2(n-1)-1)} = 0$.
Thus,

$$\begin{aligned} a_{2(n-1)-1}(y) &= p(x,s_1)_{2(n-1)-1} \\ &= p(x,0)_{2(n-1)-1}, \end{aligned}$$

where the last equality is needed to satisfy condition (3.24).
Hence we continue to $a_{2(n-1)-2}(y) = v_0^{(2(n-1)-2)} + v_1^{(2(n-1)-2)}y + v_2^{(2(n-1)-2)}y^2$. By a completely similar analysis, one easily finds

$$\begin{aligned} a_{2(n-1)-2}(y) &= p(x,s_1)_{2(n-1)-2} \\ &= p(x,0)_{2(n-1)-2}. \end{aligned}$$

In fact, this analysis can be repeated for the first $n-2$ terms, yielding that for $0 \le i \le n-3$,

$$\begin{aligned} a_{2(n-1)-i}(y) &= p(x,s_1)_{2(n-1)-i} \\ &= p(x,0)_{2(n-1)-i}. \end{aligned}$$

2. $a_{2(n-1)-(n-2)}(y)$
   For $a_{2(n-1)-(n-2)}(y)$, we have

$$a_{2(n-1)-(n-2)}(y) = v_0^{(2(n-1)-(n-2))} + v_1^{(2(n-1)-(n-2))}y + \cdots + v_{n-2}^{(2(n-1)-(n-2))}y^{n-2}.$$

Just as before, we plug in the conditions (3.25).

$$\begin{aligned} v_0^{(2(n-1)-(n-2))} + s_1 v_1^{(2(n-1)-(n-2))} + \cdots + s_1^{n-2} v_{n-2}^{(2(n-1)-(n-2))} &= p(x,s_1)_{2(n-1)-(n-2)} \\ v_0^{(2(n-1)-(n-2))} + s_2 v_1^{(2(n-1)-(n-2))} + \cdots + s_2^{n-2} v_{n-2}^{(2(n-1)-(n-2))} &= p(x,s_1)_{2(n-1)-(n-2)} \\ &\vdots \\ v_0^{(2(n-1)-(n-2))} + s_{n-2} v_1^{(2(n-1)-(n-2))} + \cdots + s_{n-2}^{n-2} v_{n-2}^{(2(n-1)-(n-2))} &= p(x,s_1)_{2(n-1)-(n-2)}. \end{aligned}$$

To satisfy condition (3.24), we need $v_0^{(2(n-1)-(n-2))} = p(x,0)_{2(n-1)-(n-2)}$. Therefore the above system of equations becomes

$$\begin{aligned} s_1 v_1^{(2(n-1)-(n-2))} + \cdots + s_1^{n-2} v_{n-2}^{(2(n-1)-(n-2))} &= p(x,s_1)_{2(n-1)-(n-2)} - p(x,0)_{2(n-1)-(n-2)} \\ s_2 v_1^{(2(n-1)-(n-2))} + \cdots + s_2^{n-2} v_{n-2}^{(2(n-1)-(n-2))} &= p(x,s_1)_{2(n-1)-(n-2)} - p(x,0)_{2(n-1)-(n-2)} \\ &\vdots \\ s_{n-2} v_1^{(2(n-1)-(n-2))} + \cdots + s_{n-2}^{n-2} v_{n-2}^{(2(n-1)-(n-2))} &= p(x,s_1)_{2(n-1)-(n-2)} - p(x,0)_{2(n-1)-(n-2)}. \end{aligned}$$

One may notice that the left-hand side of this system corresponds to a special matrix: a Vandermonde Matrix. But also without this realisation, it is easy to see that the different equations are linearly independent. The right-hand side is always the same constant, while the left-hand side are pairwise different equations. Therefore, this is a system of $n-2$ linearly independent equations and $n-2$ variables and thus there is a unique solution.

3. $a_{2(n-1)-(n-1)}(y)$
   We write

$$a_{2(n-1)-(n-1)}(y) = v_0^{(2(n-1)-(n-1))} + v_1^{(2(n-1)-(n-1))}y + \cdots + v_{n-2}^{(2(n-1)-(n-1))}y^{n-2} + v_{n-1}^{(2(n-1)-(n-1))}y^{n-1}.$$

   After having set $v_0^{(2(n-1)-(n-1))} = p(x,0)_{2(n-1)-(n-1)}$, the remaining system of equations is

$$\begin{aligned} s_1 v_1^{(2(n-1)-(n-1))} + \cdots + s_1^{n-1} v_{n-1}^{(2(n-1)-(n-1))} &= p(x,s_1)_{2(n-1)-(n-1)} - p(x,0)_{2(n-1)-(n-1)} \\ s_2 v_1^{(2(n-1)-(n-1))} + \cdots + s_2^{n-1} v_{n-1}^{(2(n-1)-(n-1))} &= p(x,s_1)_{2(n-1)-(n-1)} - p(x,0)_{2(n-1)-(n-1)} \\ &\vdots \\ s_{n-2} v_1^{(2(n-1)-(n-1))} + \cdots + s_{n-2}^{n-1} v_{n-1}^{(2(n-1)-(n-1))} &= p(x,s_1)_{2(n-1)-(n-1)} - p(x,0)_{2(n-1)-(n-1)}. \end{aligned}$$

   This system has $n-2$ linearly independent equations and $n-1$ variables. Thus there is one free variable. Hence, we can write $v_i^{(2(n-1)-(n-1))}$ as a linear combination of 1 and $v_{n-1}^{(2(n-1)-(n-1))}$, for all $i \in [n-2]$.

4. $a_i(y)$ of high degree
   At this point we have found an expression for the first half of the $a_i(y)$. For the second half we impose the assumption that p is symmetric. For $n \leq i \leq 2(n-1)-1$, we write

$$\begin{aligned} a_{2(n-1)-i}(y) =& v_0^{(2(n-1)-i)} + v_1^{(2(n-1)-i)}y + \cdots + v_{n-1}^{(2(n-1)-i)}y^{n-1} \\ &+ \cdots + v_i^{(2(n-1)-i)}y^i. \end{aligned}$$

   Condition (3.24) implies $v_0^{(2(n-1)-i)} = p(x,0)_{2(n-1)-i}$. Next, we consider $v_n^{(2(n-1)-i)}$. This is the coefficient of the monomial $x^{2(n-1)-i}y^n$. Because of symmetry, this should be equal to the coefficient of $x^n y^{2(n-1)-i}$. Since $n = 2(n-1)-(n-2)$, we have determined this coefficient in the second category above and there is a unique solution for $v_n^{(2(n-1)-i)}$. And if $i \geq n+1$, we have by a similar analysis that for $n+1 \leq j \leq i$, the coefficient of $x^{2(n-1)-i}y^j$ should be equal to the coefficient of $x^j y^{2(n-1)-i}$. We have determined the coefficients of these monomials in the first category. Since $2(n-1)-i \neq 0$, these coefficients are in fact all equal to zero, because all polynomials in the first category are just constants. Hence, the only variables we should still determine are $v_\ell^{(2(n-1)-i)}$ for $1 \leq \ell \leq n-1$. Plugging in all conditions (3.25) yields a system of $n-2$ linearly independent equations and $n-1$ variables. So there is one free variable, namely $v_{n-1}^{(2(n-1)-i)}$. Because of symmetry, this is also the coefficient of the monomial $x^{n-1}y^{2(n-1)-i} = x^{2(n-1)-(n-1)}y^{2(n-1)-i}$. The analysis of the third category has shown that $v_{n-1}^{(2(n-1)-i)}$ can be written as a linear combination of 1 and $v_{n-1}^{(2(n-1)-(n-1))}$.
   This is the point where things become more complicated. Among the terms of the different $a_i(y)$ of high degree in this category, there are also linear dependencies. To illustrate this, consider an arbitrary $i > n+2$. There is a coefficient $v_{n-2}^{(2(n-1)-i)}$ that corresponds to the monomial $x^{2(n-1)-i}y^{n-2}$. Since $n-2 = 2(n-1)-n$, symmetry implies that $v_{n-2}^{(2(n-1)-i)} = v_{2(n-1)-i}^{(2(n-1)-n)}$, another coefficient in this category. For both coefficients we had already found an expression in terms of $v_{n-1}^{(2(n-1)-(n-1))}$ to which it should be equal. This could possibly lead to contradictions. Going back to Example 62, we found that $v_1^{(2)} = v_2^{(1)} = -\frac{549}{2} - v_3^{(3)}$. However, this equality was never explicitly imposed in the calculations. If these two coefficients had not been equal to each other, finding p would have been impossible. Since for the grids that we are considering, there always exists a tight cover with symmetric p, it seems the case that these coefficients will always have the same expression in terms of the last remaining free variable, but that is still to be proven.

❇ **Question 2.** ❇ *Can we prove that making p symmetric does not yield any contradictions in the linear constraints on the coefficients?*

If we assume that there indeed are no contradictions and that we can find a proper expression for each $a_i(y)$ in the category above, the proof of tightness for the Ball-Serra Bound for these grids follows. There is still one expression that we need to find.

5. $a_0(y)$

   The last expression remaining is the one of $a_0(y)$. One may note that $a_0(y) = p(0, y)$. Using symmetry, this is equal to $p(y, 0)$, which we know.

In that case, we have an expression of infinitely many $p(x, y)$ – because of the free variable $v_{n-1}^{(2(n-1)-(n-1))}$ – such that p is symmetric and has slices equal to

$$
\begin{aligned}
p(x,0) &= \prod_{s\in S_1\setminus\{0\}} (x-s)^2 \\
p(x,s_i) &= x^2 \prod_{s\in S_1\setminus\{0\}} (x-s)\cdot c(x), \quad \forall s_i \in S_2\setminus\{0\} \\
p(0,y) &= \prod_{s\in S_1\setminus\{0\}} (y-s)^2 \\
p(s_i,y) &= y^2 \prod_{s\in S_1\setminus\{0\}} (y-s)\cdot c(y), \quad \forall s_i \in S_2\setminus\{0\}
\end{aligned}
$$

For any choice of the free variable, $p(x, y)$ has roots of multiplicity one at all interior points and roots of multiplicity two at all boundary points except $(s_{n-1}, 0)$. This point is a root of multiplicity at least one, because of the slice $p(x, 0)$. Moreover, this slice implies that $\frac{\partial p}{\partial x}(s_{n-1}, 0) = 0$. Hence, we only still require $\frac{\partial p}{\partial y}(s_{n-1}, 0) = 0$. This last linear equation determines the value of $v_{n-1}^{(2(n-1)-(n-1))}$. The obtained polynomial p would then be exactly the polynomial that we were looking for.
Hence, there is only one step remaining to find the existence of the polynomial p, namely proving that requiring the polynomial to be symmetric does not yield any contradictions. All generated polynomials support this assumption. In Appendix A.1, one can verify that a symmetric polynomial has been found for every grid that has been considered. Proving this existence will nevertheless require some additional effort. Once we have obtained p, a tight polynomial 3-cover of Γ can be obtained by adding the horizontal and vertical lines.
If we first consider the case where $|S_1| = n$ and $|S_2| = m = n-1$, and there is a polynomial p of degree $2(n-1)$ that covers every boundary point twice and every interior point once, while avoiding the origin, then we add every horizontal and every vertical line once:

$$
f = p \cdot \prod_{s\in S_1\setminus\{0\}} (x-s) \cdot \prod_{t\in S_2\setminus\{0\}} (y-t).
$$

Clearly, this covers every non-zero point of Γ three times and does not vanish at the origin. Furthermore,

$$
\begin{aligned}
\deg f &= \deg p + n - 1 + n - 2 \\
&= 2(n-1) + n - 1 + n - 2 \\
&= 3(n-1) + m - 1.
\end{aligned}
$$

For grids where S_2 has a smaller size, Lemma 60 implies the existence of a tight 3-cover if the above cover exists.

3.3.4. Asymmetric Slices of 3-covers

For $(n \times m)$-grids where the points on the y-axis are a subset of the points on the x-axis, there are quite some indications that Ball-Serra is tight whenever $n \geq m - 1$. Table 3.2 gave the first suggestion that this is the correct threshold and this was supported by the method of slices. However, there is

no definitive answer on the correctness of the threshold yet. If we drop the constraint that one set of axis points has to be a subset of the other set, Table 3.2 does not change, suggesting that the same threshold should hold for any grid. But in this case, proving tightness seems to be even more difficult as we cannot use symmetry to reduce the number of free variables. First of all, the horizontal slices can again be chosen using Lemma 63. But it is unclear how the vertical slices should be picked. Is there an incentive to pick the slices such that some of the coefficients are matched up? Or should it be chosen related to the systems of equations that are given by the horizontal slices? And how do we combine horizontal and vertical slices to one polynomial cover?
Instead of using slices to prove the existence of certain polynomial covers, we can also try to use them to disprove this existence. For example, let n be an integer and consider an arbitrary $(n\times n)$-grid $\Gamma = S_1 \times S_2$ that contains the origin for which we want to find a tight 3-cover $f(x, y)$ that contains all horizontal and all vertical lines at least once, together with some other factor $p(x, y)$. Then $\deg f = 3(n-1) + (n-1)$ and it follows that $\deg p = 2(n-1)$.
Moreover, if we let $S_2 = \{0, t_1, \dots, t_{n-1}\}$, we need that

$$p(x, 0) = \prod_{s \in S_2 \setminus \{0\}} (x - s)^2$$
$$p(x, t_i) = x^2 \prod_{s \in S_2 \setminus \{0\}} (x - s) \cdot c_i(x),$$

for all $i \in [n-1]$. We again set

$$p(x, y) = a_{2(n-1)}(y) \cdot x^{2(n-1)} + a_{2(n-1)-1}(y) \cdot x^{2(n-1)-1} + \dots + a_1(y) \cdot x + a_0(y),$$

where $a_i(y)$ have degree $2(n-1) - i$ for $i \in \{0, \dots, 2(n-1)\}$.
Then by repeating the steps from Section 3.3.3, we notice that $0 \leq i \leq n-2$, finding $a_{2(n-1)-i}(y)$ corresponds to solving a linear system of $n-1$ equations and at most $n-2$ variables. By using Lemma 63, we are only able choose the slices such that we can for sure solve $n-2$ of these $n-1$ linear systems. It might be possible that because of a very specific choice of S_1 and S_2, matching up the first $n-2$ terms led to the $(n-1)^{\text{th}}$ terms also being matched, but in general, one of the linear systems contains a contradiction. This means that the polynomial p does not exist and gives intuition why Ball-Serra seems not to be tight for square grids. Allowing p to have a degree $2(n-1)+1$ solves this problem as we are able to match up one additional term in that case, so this might explain why we see a gap of one between the optimal cover and the Ball-Serra Bound for $(n \times n)$-grids. This analysis does rely on the assumption that if, for some grid Γ, there exists a 3-cover of degree d, there exists a 3-cover of Γ of degree d that contains every non-zero horizontal and vertical line of Γ at least once. While all generated covers in Appendix A.1 indicate this to be true, this still remains to be proven. A good first step might be to retrace the steps of the proof of the Punctured Combinatorial Nullstellensatz (Theorem 31) while adding the restriction of the horizontal and vertical lines. Another approach might be to immediately look at the slices of f rather than those of p. But setting

$$f(x, 0) = \prod_{s \in S_2 \setminus \{0\}} (x - s)^3 \cdot c_0(x),$$

leaves us with a choice for a polynomial $c_0(x)$ of degree $n-1$. Proving that there is no choice for this polynomial such that f is a tight 3-cover still requires some work. In any case, the analysis of the last two sections allow us to formulate a conjecture and a question on the tightness of the Ball-Serra Bound.

❇ **Conjecture 1.** ❇ *Let $S_1, S_2 \subseteq \mathbb{R}_{\geq 0}$ such that $S_1 = \{0, s_1, \dots, s_{n-1}\}$ and $S_2 = \{0, t_1, \dots, t_{m-1}\}$ for integers $2 \leq m < n$. Let $\Gamma = S_1 \times S_2$. Then there exists a polynomial f of degree $3(n-1) + (m-1)$ that covers every non-zero point of Γ three times while avoiding the origin. Furthermore, f contains each vertical line $x = s_i$ for $i \in [n-1]$ and horizontal line $y = t_i$ for $i \in [m-1]$ at least once.*

In the case where $|S_1| = |S_2|$, it would be interesting to find out whether the bound is never tight for a 3-cover of $S_1 \times S_2$, or whether there are specific grids for which a tight 3-cover does exist.

❇ **Question 3.** ❇ *Do there exist $S_1, S_2 \subseteq \mathbb{R}_{\geq 0}$ both of size n and both containing zero such that there is a 3-cover of $S_1 \times S_2$ of degree $4(n-1)$?*

3.3.5. Polynomial Covers of Higher Multiplicity

To see whether the conjectured threshold for polynomial 3-covers can be generalised to polynomial k-covers for $k \geq 4$, we start by exploring the values of polynomial 4-covers of the $(n \times m)$-grids $\{0, 1, \ldots, n-1\} \times \{0, 1, \ldots, m-1\}$ for small values of n and m. These values can be generated by adapting the existing code. The only change that needs to be made is adding the third order derivatives as well. However, the symbolic code becomes extremely slow and Sage also starts making numeric errors when performing Gaussian Elimination, especially for square grids. Hence, certain observations again have to be made to speed up the code. First of all, there again seems to always exist a cover of best possible degree that contains every non-zero horizontal and vertical line once. For the example grids with a tight polynomial 4-cover with respect to the Ball-Serra Bound, there always exists such a cover that contains every non-zero vertical line twice and all horizontal lines once. As assuming the existence of these lines decreases the remaining degree of the cover and speeds up the algorithm, it makes sense to first search for a cover with the horizontal lines twice. If this doesn't exist, we continue with the horizontal lines once. The SymPy code for this approach is given in Appendix B.3. The algorithm allows us to solve for n up to seven. For greater n, even the symbolic code starts making numerical errors, meaning that a different approach is needed if we want to generate more values. The optimal degrees of the 4-covers are given in Table 3.3 and the corresponding covers are plotted in Appendix A.2.

m \ n	2	3	4	5	6	7
2	6	10	13	17	21	25
3		11	15	18	22	26
4			16	20	23	27
5				21	25	28
6					26	30
7						31

0
1

Table 3.3: Minimum degree of a polynomial 4-cover for the grids $\{0, 1, \ldots, n-1\} \times \{0, 1, \ldots, m-1\}$.

From the first look at this table and the plots, many interesting observations can be made. First of all, there is a clear distinction between the hyperplane 4-cover in Table 3.1 and the polynomial covers. Again, the threshold seems to behave differently. A second remark is that the considered grids $\{0, 1, \ldots, n-1\} \times \{0, 1, \ldots, m-1\}$ are a specific type of grid. We considered these to compare them to line covers in Table 3.1, but it has to be noted that the fact that the grids are evenly-spaced has an influence on their polynomial covering number. For instance, for the square grids in this table, the gap between the minimum degree and the Ball-Serra Bound is one. Consider the grid $\{0, 1, 2\} \times \{0, 1, 2\}$. A polynomial 4-cover of degree 11 is

$$
\begin{aligned}
f(x, y) = &(x-2)\cdot(x-1)\cdot(y-2)\cdot(y-1)\cdot(x+y-2)\cdot(x^2+xy-3x+y^2-3y+2)\cdot(2x^4 \\
&-4x^3y-2x^3-3x^2y^2+21x^2y-14x^2-4xy^3+21xy^2-43xy+26x+2y^4-2y^3 \\
&-14y^2+26y-12).
\end{aligned}
$$

The plot of this polynomial is given in Figure 3.5. From this plot, it is clear that the polynomial contains a conic that goes through six points of the grid. This is of course not possible for all grids. Because of this conic, this grid can be covered in a more efficient way than usual. For example, the grid $\{0, 1, 5\} \times \{0, 1, 2\}$ requires a polynomial of degree 12 to find a 4-cover. The same remark can be made for the other square grids in the table, where in general the gap with the Ball-Serra Bound is equal to two.

Thirdly, as we already mentioned, in the cases in the table where the bound is tight, there exists a cover that contains every non-zero horizontal line of the grid once and every non-zero vertical line twice. But because of the evenly-spaced grids that have lower degree 4-cover than most of the grids of the same size, there are grids where no optimal 4-cover contains every vertical line twice. For instance, the grid $\{0, 1, 2, 3\} \times \{0, 1, 2, 3\}$ has a 4-cover of degree 16. This cover contains every vertical and horizontal line once. If we require that every vertical line is contained twice in the cover, then the minimum degree is not even 17, it is equal to 18. The reason why it is interesting that there are tight 4-covers with the vertical lines twice and the horizontal lines once is that it provides a link with the slices that we

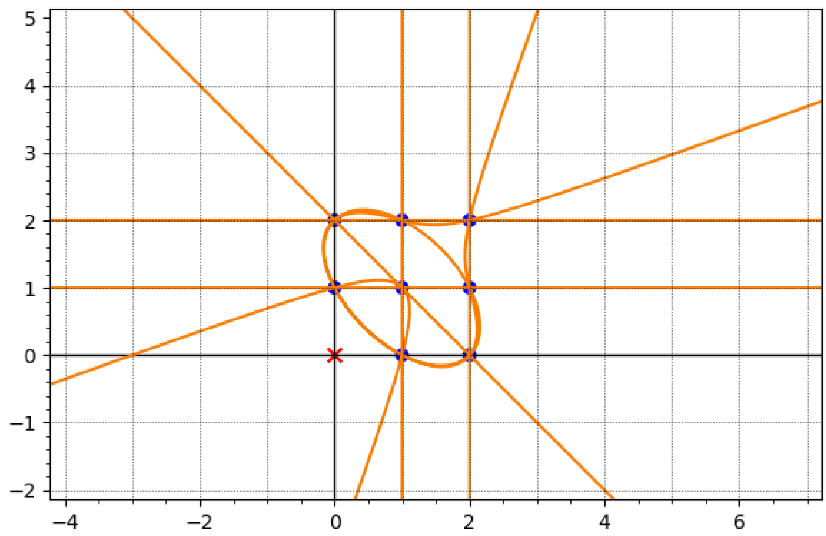


Figure 3.5: Plot of a 4-cover of $\{0,1,2\} \times \{0,1,2\}$ of degree 11. This polynomial contains a conic through six points of the grid.

investigated for 3-covers. A tight cover f of an $(n \times m)$-grid $S_1 \times S_2$ has degree $4(n-1)+(m-1)$. Assuming that f is of the form

$$f(x,y) = \prod_{s \in S_1 \setminus \{0\}} (x-s)^2 \cdot \prod_{t \in S_2 \setminus \{0\}} (y-t) \cdot p(x,y),$$

then $\deg p \leq 2(n-1)$, the same maximum degree p had in the 3-cover case. Furthermore, let k be an arbitrary integer and let $S_1, S_2 \subseteq \mathbb{R}_{\geq 0}$ be sets such that $S_1 = \{0, s_1, \ldots, s_{n-1}\}$ and $S_2 = \{0, t_1, \ldots, t_{m-1}\}$ and such that $n \geq k$. If we assume that there always exists a tight k-cover of $S_1 \times S_2$ that contains every horizontal line once and every vertical line $k-2$ times the degree of the remaining factor $p(x,y)$ is always $2(n-1)$. Again considering the horizontal slices of this polynomial p, we have

$$p(x,0) = \prod_{s \in S_1 \setminus \{0\}} (x-s)^2$$
$$p(x,t_i) = x^{k-1} \cdot \prod_{s \in S_1 \setminus \{0\}} (x-s) \cdot c_i(x),$$

where $p(x,0)$ has degree $2(n-1)$ such that it is impossible that p vanishes at the origin and where $\deg c_i(x) \leq n-k$ for $i \in [m-1]$. Such a $c_i(x)$ always exists since $n \geq k$. We can pick all $c_i(x)$ to be equal to the polynomial given by Lemma 63 with $\ell = n-k$, such that the first $n-k+1$ terms of $p(x,0)$ are equal to the first $n-k+1$ terms of $p(x,t_1)$ and such that every slice at $y \neq 0$ is equal.
At first, it might seem arbitrary to follow this approach. But in the following steps, it will become clear that these slices again provide support for making a conjecture on the threshold for tightness of the Ball-Serra Bound. Keeping the same notation as before, we write

$$p(x,y) = a_{2(n-1)}(y) \cdot x^{2(n-1)} + a_{2(n-1)-1}(y) \cdot x^{2(n-1)-1} + \cdots + a_1(y) \cdot x + a_0(y).$$

Since we have matched the first $n-k+1$ terms, we let

$$\begin{aligned} a_{2(n-1)-i}(y) &= p(x,t_1)_{2(n-1)-i} \\ &= p(x,0)_{2(n-1)-i}, \end{aligned}$$

for $0 \leq i \leq n-k$.

Now we consider $a_{2(n-1)-(n-2)}(y)$, with formula

$$a_{2(n-1)-(n-k+1)}(y) = v_0^{(2(n-1)-(n-k+1))} + v_1^{(2(n-1)-(n-k+1))}y + \dots + v_{n-k+1}^{(2(n-1)-(n-k+1))}y^{n-k+1}.$$

The slice at the x-axis requires that $v_0^{(2(n-1)-(n-k+1))} = p(x,0)_{2(n-1)-(n-k+1)}$. Plugging this in and and adding the other conditions of the slices, we obtain the system of equations

$$\begin{aligned}
t_1 v_1^{(2(n-1)-(n-k+1))} + \dots + t_1^{n-k+1} v_{n-k+1}^{(2(n-1)-(n-k+1))} &= p(x,t_1)_{2(n-1)-(n-2)} - p(x,0)_{2(n-1)-(n-2)} \\
t_2 v_1^{(2(n-1)-(n-k+1))} + \dots + t_2^{n-k+1} v_{n-k+1}^{(2(n-1)-(n-k+1))} &= p(x,t_1)_{2(n-1)-(n-2)} - p(x,0)_{2(n-1)-(n-2)} \\
&\vdots \\
t_{m-1} v_1^{(2(n-1)-(n-k+1))} + \dots + t_{m-1}^{n-k+1} v_{n-k+1}^{(2(n-1)-(n-k+1))} &= p(x,t_1)_{2(n-1)-(n-2)} - p(x,0)_{2(n-1)-(n-2)}.
\end{aligned}$$

This is a system of $m-1$ linearly independent equations and $n-k+1$ variables. Hence, this system only has a solution whenever $m \leq n-k+2$. Looking at Table 3.3, this is the same threshold when the generated covers are tight with respect to the Ball-Serra Bound. Thus, we make the following bold conjecture.

❇ **Conjecture 2.** ❇ *Let n, m, k be integers such that $m \leq n-(k-2)$ and $k \leq n$. Consider two arbitrary sets $S_1, S_2 \subseteq \mathbb{R}_{\geq 0}$ such that $S_1 = \{0, s_1, \dots, s_{n-1}\}$ and $S_2 = \{0, t_1, \dots, t_{m-1}\}$. Let $\Gamma = S_1 \times S_2$. Then there exists a polynomial f of degree $k(n-1)+(m-1)$ that covers every non-zero point of Γ k times while avoiding the origin. Furthermore, f contains every vertical line $x = s_i$ for $i \in [n-1]$ at least $k-2$ times and every horizontal line $y = t_i$ for $i \in [m-1]$ at least once.*

The especially bold part of this conjecture lies in its second half. It does not feel very efficient that the optimal cover contains the vertical lines this many times. However, the horizontal slices of this construction neatly show where the threshold pops up. Moreover, if we allow p to have a degree one higher than the Ball-Serra Bound, the 'critical' system of equations has one more linearly independent equation, allowing it to have a solution for $m \geq n-(k-2)+1$, which is exactly the behaviour we expect based on the numerical data. Also, having a lot of copies of the vertical lines resembles the construction in the proof of the threshold for line k-covers in Theorem 48. In general, we expect the following behaviour of the optimal value of k-covers outside the threshold.

❇ **Conjecture 3.** ❇ *Let n, m, k be integers such that $m > n-(k-2)$. Consider two arbitrary sets $S_1, S_2 \subseteq \mathbb{R}_{\geq 0}$ such that $S_1 = \{0, s_1, \dots, s_{n-1}\}$ and $S_2 = \{0, t_1, \dots, t_{m-1}\}$. Let $\Gamma = S_1 \times S_2$ and let j be the integer such that $m-j = n-(k-2)$. Then there exists a polynomial f of degree $k(n-1)+(m-1)+j$ that covers every non-zero point of Γ k times while avoiding the origin.*

Further research is required to prove both conjectures. It should still be investigated how the vertical slices have to be chosen to find a cover. Moreover, we have seen that the construction with multiple vertical lines is not always optimal. There are grids that can be covered more efficiently with the lines only included once. Table 3.3 is an example where both conjectures hold, but where Conjecture 3 is not tight for the $(n \times n)$-grids. In the same table, we have also seen that there exist $(n \times m)$-grids such that $m-1 = n-(4-2)$ and for which the polynomial 4-covering number is equal to $4(n-1)+(m-1)+1$. Hence, the conjectured threshold $m \leq n-(k-2)$ cannot be sharpened for all grids. It would be interesting to investigate when the conjectured upper bound on the optimal degree is tight and whether we can characterise the grids for which it is not. Perhaps the notion of generic grids can be extended such that it also holds for k-covers. Furthermore, perhaps there are specific grids that do not adhere to the threshold but for which there exists a k-cover that is tight with respect to the Ball-Serra Bound. This would imply that there are is a level as to how non-generic certain grids are for polynomial covers, based on by how much their optimal covers beat the expected degree in Conjecture 3.

3.4. Conclusion

This chapter investigated whether the Ball-Serra Bound is tight for various types of grids. As part of this investigation, we distinguished between hyperplane and polynomial k-covers. We have seen that for the hypercube, the Ball-Serra Bound is only tight for 1-covers and 2-covers. Once the multiplicity of the cover is greater than or equal to three, the bound is not even tight for polynomial covers, as shown by Sauermann and Wigderson [19]. They improved the lower bound for a k-cover of Q^n from $n+(k-1)$ to $n+2k-3$. On one hand, this bound is tight for polynomial k-covers of the hypercube. On the other, it has been conjectured by Clifton and Huang [11] that the minimal size of a hyperplane k-cover is $n+\binom{k}{2}$. If this conjecture holds, there is a clear difference between polynomial and hyperplane covers of the hypercube.
Secondly, in Section 3.2, we looked into covering the binary field. As the usual derivatives do not hold in finite fields, we started with a short introduction into Hasse derivatives. Using these derivatives, we showed that the optimal polynomial 4-cover of $\mathbb{F}_2^n$ has degree $n+4$ and is thus never tight with respect to the Ball-Serra Bound. So obviously the bound is not tight for the hyperplane 4-cover number either. Since we know the exact polynomial 4-covering number of $\mathbb{F}_2^n$, the question arises whether we can use this result to find improved lower bounds on polynomial covers of higher multiplicity. Perhaps the method with Hasse derivatives can be generalised, or there could be a recurrence relation between the multiplicities. However, it is unclear what such a relation would look like, as, contrary to the hyperplane cover, we cannot just delete a single degree of a polynomial cover. When looking at the hyperplane covering number rather than the polynomial one, the exact value is also unknown. When relaxing the problem to *almost* k-covers of $\mathbb{F}_2^n$, Bishnoi et al. [9] have proven the behaviour of the hyperplane cover number when n is fixed and k is large and the other way around. Yet again, this behaviour was different compared to the polynomial covering number. So, in the setting of the binary field, there are still plenty of interesting research directions.
The final and most extensive part of this section was spent on exploring how to cover grids in the Cartesian plane. For the hyperplane k-cover of an $(n \times m)$-grid, it is known that the Ball-Serra Bound is tight whenever $n \geq (k-1)(m-1)+1$, a result that was first proven in den Bakker's thesis [3]. If the grid does not satisfy this threshold, the gap with the Ball-Serra Bound becomes larger as n and m become larger even if their difference stays equal. When investigating the behaviour of the polynomial k-cover of the same $(n \times m)$-grid, a lot of differences could be noticed. We conjectured that the Ball-Serra Bound is tight whenever $n \geq m+(k-2)$. This threshold grows considerably slower than the threshold for the line covers. Moreover, it seems that the gap between the polynomial covering number and the Ball-Serra Bound is upper bounded by the difference $(m+(k-2))-n$. Hence, contrary to the hyperplane covering number, the gap does not seem to increase when n and m grow while their difference stays equal. It is sure that the conjectured upper bound on the gap is not always tight though. The grids $\{0,1,\dots,n-1\} \times \{0,1,\dots,n-1\}$ are an example for which the upper bound can be beaten. As further research, it would be interesting to investigate if we can determine the properties of grids that can be covered more efficiently than expected. This characterisation could be a generalisation of non-generic grids, but now for polynomial covers. At this point, it is unclear what this characterisation would look like. Another interesting open research question is whether there exists a grid that does not satisfy the threshold, but for which the Ball-Serra Bound is tight. Even tough we have found evenly-spaced grids that are efficient to cover with polynomials, none of the grids outside the threshold had a 4-cover of tight degree with respect to the Ball-Serra Bound. Hence, if such a grid exists, it should consists of a very specific structure to ensure enough linear dependencies in the constraints. A last proposed further research is maybe the most ambitious one. Until now, we have only considered two-dimensional grids in the Cartesian plane. The problem of course be generalised to finding polynomial k-covers of d-dimensional grids in $\mathbb{R}^d$, for $d \geq 3$. Since we had already difficulties of keeping track of the slices in two dimensions, adding more dimensions seems beyond our current capabilities. But this generalisation is for sure something to keep in mind when further refining the existing methods or when coming up with completely new techniques to find polynomial k-covers of grids in the plane.

4
Conclusion

Drawing this thesis to a close, the only remaining part is to look back at its important results and to look ahead to its most interesting further research directions. We investigated polynomial covers and hyperplane covers for all kinds of values of the multiplicity and for multiple grids. Even though, at first glance, the covering problem seems to be geometrical in nature, most of the arguments that we considered were based on the polynomial method. For covers with multiplicity one, the Alon-Füredi bound provided the optimal covering number.

Corollary 3 (The Alon-Füredi Bound)**.** *Let $S_1, \dots, S_n$ be subsets of an arbitrary field $\mathbb{F}$. Consider a polynomial p such that p vanishes on $S_1 \times S_2 \times \cdots \times S_n$ except at one point. Then $\deg(p) \geq \sum_i (|S_i| - 1)$.*

The Alon-Füredi Bound is tight for line covers and hence also for polynomial covers. We considered two ways to prove this bound. The first approach in Section 2.2 used Gröbner bases to prove the Footprint Bound, which estimates the number of common zeroes of an ideal, based on the number of standard monomials of that ideal. By choosing a specific ideal, the Alon-Füredi Bound elegantly follows from the Footprint Bound. The second approach in Section 2.3 involved one of the Cayley-Bacharach theorems. This is a set of theorems in algebraic geometry with a rich history dating back to the 19th century. They all involve a bound on the number of linear constraints imposed by a set of points on polynomials of fixed degree that vanish on these points. By letting the set of points be equal to the grid we want to cover, we could again retrieve the Alon-Füredi Bound. Investigating the linear constraints imposed by a grid on a polynomial of fixed degree continued to play a vital role throughout all remaining sections of the thesis. When increasing the multiplicity of a cover from one to some $k \geq 2$, the best lower bound on its degree is given by the Ball-Serra Bound.

Theorem 31 (The Ball-Serra Bound)**.** *Let $S_1, \dots, S_n$ and $D_1, \dots, D_n$ be arbitrary sets such that $D_i \subset S_i$ for all $i \in [n]$. If f vanishes at least k times at all elements of $S_1 \times \cdots \times S_n$, except at at least one point of $D_1 \times \cdots \times D_n$ where it does not vanish, then*

$$\deg(f) \geq (k-1)\max_j \left(|S_j| - |D_j|\right) + \sum_{i=1}^{n} (|S_i| - |D_i|).$$

We derived this from the Punctured Combinatorial Nullstellensatz in Section 2.4. While the Alon-Füredi Bound is tight for hyperplane and polynomial covers, it is unclear when the Ball-Serra Bound is tight for any type of cover. That is what we investigated in the next chapter, Chapter 3. We looked into three different grids: the hypercube, the binary field and grids in the Cartesian plane. In the case of hyperplane covers with multiplicity one, the hypercube Q^n was arguably the easiest setting to find the minimal size of the cover. When increasing the multiplicity however, it quickly became clear that even in this setting it is difficult to decide what the optimal size is. For hyperplane 2-covers, it was not too complicated to show that the Ball-Serra Bound is tight. Increasing the multiplicity, different methods were required, but Clifton and Huang were able to determine the minimum size of a hyperplane 3-cover and find a lower bound on the minimum sizes of hyperplane covers of multiplicity greater than or equal to four [11].

Theorem 37 (Clifton-Huang)**.** *Let $f(n,k)$ denote the minimum size of a hyperplane k-cover of Q^n. For $n \geq 2$,*

$$f(n,3) = n+3.$$

For $k \geq 4$ and $n \geq 3$,

$$n+k+1 \leq f(n,k) \leq n + \binom{k}{2}.$$

The lower bound was obtained by the simple realisation that removing one hyperplane from a k-cover yields a $(k-1)$-cover, which results in the recurrence relation $f(n,k) \geq f(n,k-1)+1$. This theorem shows that the Ball-Serra Bound is never tight for hyperplane k-covers of the hypercube for $k \geq 3$. But Clifton and Huang even conjectured that the actual hyperplane k-covering number is equal to $n + \binom{k}{2}$. Later, Sauermann and Wigderson were able to improve the lower bound, again using polynomials [19].

Theorem 38 (Sauermann-Wigderson)**.** *Let $k \geq 2$ and $n \geq 2k-3$. Then any polynomial $p \in \mathbb{R}[x_1, \dots, x_n]$ with $p(\bar{0}) \neq 0$ having zeroes of multiplicity at least k at all points in $\{0,1\}^n \backslash \{\bar{0}\}$ has degree $\deg p \geq n+2k-3$. Furthermore, there exists such a polynomial p with degree $\deg p = n+2k-3$.*

Due to this theorem, the polynomial k-covering problem of the hypercube is immediately solved. Moreover, it shows that there might be a separation in the minimum degree of a polynomial k-cover and the minimum size of a hyperplane k-cover of the hypercube. Hence, to prove the conjecture of Clifton and Huang, additional techniques are required that take into account that we are specifically considering a hyperplane cover. The polynomial method in itself will not suffice. Since the Sauermann-Wigderson Theorem only holds for fields with characteristic equal to zero, the next sensible step was to investigate the covering number of fields with positive characteristic. Specifically, we looked into how to cover the binary field. Since the usual derivatives no longer hold for finite fields, we first introduced Hasse derivatives, a generalisation of derivatives that enabled us to determine the multiplicities of roots of polynomials in finite fields. Using these, we were able to prove the polynomial 4-covering number of $\mathbb{F}_2^n$.

❇ **Theorem 46.** ❇ *The polynomial 4-covering number of $\mathbb{F}_2^n$ is equal to $n+4$.*

The 4-covering number is less than the lower bound on fields with zero characteristic in the Sauermann-Wigderson Theorem. But it is still greater than the Ball-Serra Bound, showing that the bound is not tight in this case. Since Clifton and Huang were able to derive bounds on hyperplane covers of all multiplicities of the hypercube with a recurrence relation, the question arose whether something similar is possible for the binary field.

❇ **Question 1.** ❇ *Can we find bounds on the polynomial k-covering number of $\mathbb{F}_2^n$ for $k \geq 5$, knowing that the optimal polynomial 4-cover has degree $n+4$?*

Finding a recurrence relation for polynomial covering numbers is unfortunately not as straightforward as for hyperplane covering numbers. In the latter, a hyperplane can easily be removed from a cover, while in the former, removing a single degree is impossible. Reverting back to hyperplane coverings, their minimum size in the binary field remain unknown at this moment. Because of its equivalence with a well-studied – and unsolved – problem, we regarded a relaxed version of the hyperplane k-covering problem where the origin is allowed to be covered at most $k-1$ times. In this case, a difference in hyperplane and polynomial coverings could be found.

Lastly, we considered grids in the Cartesian plane. The hyperplane k-covering problem of these grids has already been studied and the threshold for tightness of the Ball-Serra Bound for covering $(n \times m)$-grids is known [8].

Theorem 48. *Let $S_1, S_2 \subseteq \mathbb{R}$ have respective sizes $|S_1| = n$ and $|S_2| = m$. Assume $0 \in S_1 \cap S_2$ and let $\Gamma = S_1 \times S_2$. If for a positive integer k, we have $n \geq (k-1)(m-1)+1$,*

$$\mathrm{cov}_k(\Gamma) = k(n-1) + (m-1).$$

The polynomial k-covering problem of grids in the Cartesian plane had not been studied yet. To investigate this problem, we started by setting up an algorithm based on partial derivatives that finds 3-covers for small grids. These covers raised the idea that the Ball-Serra Bound is tight whenever $n \geq m+1$. Moreover, we found that for every grid, there was a 3-cover that contains every horizontal and vertical line once. And if the points on the y-axis are a subset of the points on the x-axis, the remaining factor of the polynomial cover could always be chosen to be symmetric in x and y. To further investigate the threshold and the properties of the polynomial covers, we stepped away from the derivatives and looked into the slices of the grid. On these slices we evaluated the polynomial cover, yielding a linear system of equations for every slice. The analysis on when these systems have a solution found the exact same threshold $n \geq m+1$. Unfortunately, we were not yet able to prove this threshold. Even for symmetric polynomials there are a lot of linear dependencies and it is not fully clear if we can always ensure that those do not yield a contradiction. But since the entire analysis of the polynomial 3-covers seems to support that this is indeed the correct threshold, we posed it as a conjecture.

❇ **Conjecture 1.** ❇ *Let $S_1, S_2 \subseteq \mathbb{R}_{\geq 0}$ such that $S_1 = \{0, s_1, \ldots, s_{n-1}\}$ and $S_2 = \{0, t_1, \ldots, t_{m-1}\}$ for integers $2 \leq m < n$. Let $\Gamma = S_1 \times S_2$. Then there exists a polynomial f of degree $3(n-1)+(m-1)$ that covers every non-zero point of Γ three times while avoiding the origin. Furthermore, f contains each vertical line $x = s_i$ for $i \in [n-1]$ and horizontal line $y = t_i$ for $i \in [m-1]$ at least once.*

Besides proving this conjecture, there are still some other interesting open research directions. As mentioned, the conditions for the existence of symmetric polynomials has to be further researched. Furthermore, we saw that the threshold cannot be improved for all grids. But perhaps there are specific grids outside the threshold that are very efficient to cover and for which the Ball-Serra bound is tight. Or maybe the threshold is actually tight. Additionally, we extended the analysis for polynomial 3-covers to k-covers with $k \geq 4$. Again, the algorithmic approach suggested a threshold that was also found in the analysis of the slices. Hence, the conjecture for 3-covers could be generalised to higher multiplicities too.

❇ **Conjecture 2.** ❇ *Let n, m, k be integers such that $m \leq n-(k-2)$ and $k \leq n$. Consider two arbitrary sets $S_1, S_2 \subseteq \mathbb{R}_{\geq 0}$ such that $S_1 = \{0, s_1, \ldots, s_{n-1}\}$ and $S_2 = \{0, t_1, \ldots, t_{m-1}\}$. Let $\Gamma = S_1 \times S_2$. Then there exists a polynomial f of degree $k(n-1)+(m-1)$ that covers every non-zero point of Γ k times while avoiding the origin. Furthermore, f contains every vertical line $x = s_i$ for $i \in [n-1]$ at least $k-2$ times and every horizontal line $y = t_i$ for $i \in [m-1]$ at least once.*

Analogous to the 3-covers, we investigated what happens outside the threshold for k-covers. The code and the slices suggest that the gap between the minimum degree of a polynomial k-cover of an $(n \times m)$-grid and the Ball-Serra Bound depends on the difference $n-m$.

❇ **Conjecture 3.** ❇ *Let n, m, k be integers such that $m > n-(k-2)$. Consider two arbitrary sets $S_1, S_2 \subseteq \mathbb{R}_{\geq 0}$ such that $S_1 = \{0, s_1, \ldots, s_{n-1}\}$ and $S_2 = \{0, t_1, \ldots, t_{m-1}\}$. Let $\Gamma = S_1 \times S_2$ and let j be the integer such that $m-j = n-(k-2)$. Then there exists a polynomial f of degree $k(n-1)+(m-1)+j$ that covers every non-zero point of Γ k times while avoiding the origin.*

There are grids for which this conjectured upper bound on the minimum degree of a k-cover is not tight. For instance, for $n \leq 7$, we found polynomial 4-covers of the grids $\{0, 1, \ldots, n-1\} \times \{0, 1, \ldots, n-1\}$ of degree $4(n-1)+(n-1)+1$, which is one degree lower than proposed in the conjecture. But we also found grids for which the conjecture is tight, so in general it cannot be improved. In any case, the conjecture proposes a very different behaviour of the threshold for tightness of the Ball-Serra Bound for hyperplane and for polynomial covers. To prove this distinction, some more work is required.
In conclusion, for none of the three grids there is a proof yet that there is a different behaviour between the hyperplane k-covering number and the polynomial k-covering number. Only for almost k-covers of the binary field such a distinction has actually been proven. However, in all three cases there are clear

indications that such a difference exists. Moreover, in the case of the hypercube and the binary field, we saw that the Ball-Serra Bound is often not tight, whereas in the Cartesian plane, the bound is tight for hyperplane k-covers of infinitely many grids. Furthermore, for fixed k, the threshold of tightness for polynomials k-covers is conjectured to be even smaller than the one for k-hyperplane covers. So, there are many different aspects to covering grids with multiplicities. In this thesis, we have highlighted these aspects and their history, made some progress in finding covering numbers and have proposed different tools to attack the open questions. Hopefully, this framework can enable further progress on this interesting and intricate problem.

Bibliography

[1] N. Alon. “Combinatorial nullstellensatz”. In: *Combinatorics, Probability and Computing* 8.1-2 (1999), pp. 7–29.

[2] N. Alon and Z. Füredi. “Covering the cube by affine hyperplanes”. In: *European journal of combinatorics* 14.2 (1993), pp. 79–83.

[3] Y. den Bakker. “Hyperplane covering problems”. TU Delft, 2021. URL: `https://repository.tudelft.nl/islandora/object/uuid:b7edc7e3-a094-4f32-93fd-1fc67a91562e`.

[4] S. Ball and O. Serra. “Punctured combinatorial nullstellensätze”. In: *Combinatorica* 29.5 (2009), pp. 511–522.

[5] E. Bézout. *Théorie générale des équations algébriques*. Ph.-D. Pierres, 1779.

[6] A. Bishnoi. “Some contributions to incidence geometry and the polynomial method”. PhD thesis. Ghent University, 2016.

[7] A. Bishnoi. *The footprint bound*. Mar. 25, 2018. URL: `https://anuragbishnoi.wordpress.com/2018/03/25/the-footprint-bound/`.

[8] A. Bishnoi, S. Boyadzhiyska, S. Das, and Y. d. Bakker. “Covering grids with multiplicity”. In: *arXiv preprint arXiv:2305.00825* (2023).

[9] A. Bishnoi, S. Boyadzhiyska, S. Das, and T. Mészáros. “Subspace coverings with multiplicities”. In: *Combinatorics, Probability and Computing* 32.5 (2023), pp. 782–795.

[10] M. Chasles. *Traité des sections coniques: faisant suite au traité de géométrie supérieure... Première partie*. Gauthier-Villars, 1865.

[11] A. Clifton and H. Huang. “On almost k-covers of hypercubes”. In: *Combinatorica* 40.4 (2020), pp. 511–526.

[12] D. Cox, J. Little, D. O’shea, and M. Sweedler. *Ideals, varieties, and algorithms*. Vol. 3. Springer, 1997.

[13] D. Eisenbud, M. Green, and J. Harris. “Cayley-Bacharach theorems and conjectures”. In: *Bulletin of the American Mathematical Society* 33.3 (1996), pp. 295–324.

[14] P. Griffiths and J. Harris. *Principles of algebraic geometry*. John Wiley & Sons, 1978.

[15] D. Hilbert. “Ueber die vollen Invariantensysteme”. In: *Mathematische Annalen* 42 (1893), pp. 313–373. DOI: `https://doi.org/10.1007/BF01444162`.

[16] T. Høholdt. “On (or in) the Blahut footprint”. In: *Codes, Curves, and Signals: Common Threads in Communications* (1998), pp. 3–7.

[17] D. Husemöller. *Elliptic curves*. Vol. 2. Springer New York, NY, 2010.

[18] R. Karasev. “Residues and the combinatorial Nullstellensatz”. In: *Periodica Mathematica Hungarica* 78 (2019), pp. 157–165.

[19] L. Sauermann and Y. Wigderson. “Polynomials that vanish to high order on most of the hypercube”. In: *Journal of the London Mathematical Society* 106.3 (2022), pp. 2379–2402.

[20] B. Sturmfels. “What is... a grobner basis?” In: *Notices-American Mathematical Society* 52.10 (2005), p. 1199.

[21] T. Tao. *Pappus’s theorem and elliptic curves*. July 15, 2011. URL: `https://terrytao.wordpress.com/2011/07/15/pappuss-theorem-and-elliptic-curves/`.

[22] D. B. West et al. *Introduction to graph theory*. Vol. 2. Prentice hall Upper Saddle River, 2001.

A

Plots of Covers

This appendix contains all generated polynomials for 3-covers and 4-covers. By Lemma 60, if there is a tight polynomial cover of an $(n \times m)$-grid that contains all horizontal lines at least once, there is also a tight cover for every $(n \times \ell)$-grid for $\ell < m$. These covers can be obtained by just removing the redundant horizontal lines. On all plots, the grid points we want to cover are indicated by blue dots. The origin, which has to remained uncovered is indicated by a red cross. The polynomial cover f is given by the orange curve. We also indicate the covered grid and $\deg f$ in the caption of each figure.

A.1. 3-covers

Already for 3-covers, the formulas of the polynomials quickly become too large to fit on one page. Therefore we only give the plots of the covers. These plots are sufficient to see that for the grids $\{0, 1, \dots, n-1\} \times \{0, 1, \dots m-1\}$, there is indeed always a cover that consists of all horizontal and vertical lines and a symmetric component.

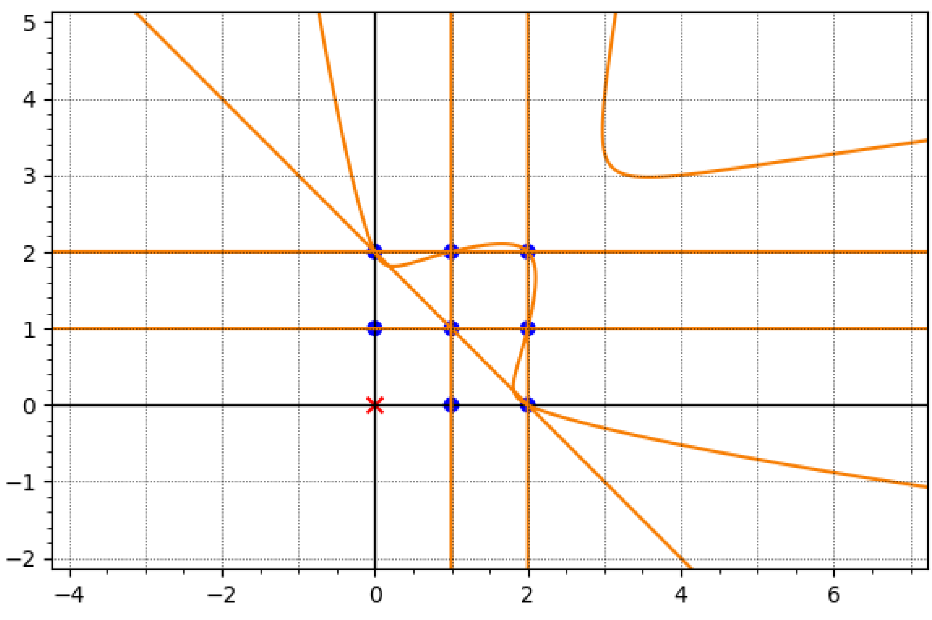


Figure A.1: $\{0, 1, 2\} \times \{0, 1, 2\}$ - $\deg(f) = 9$

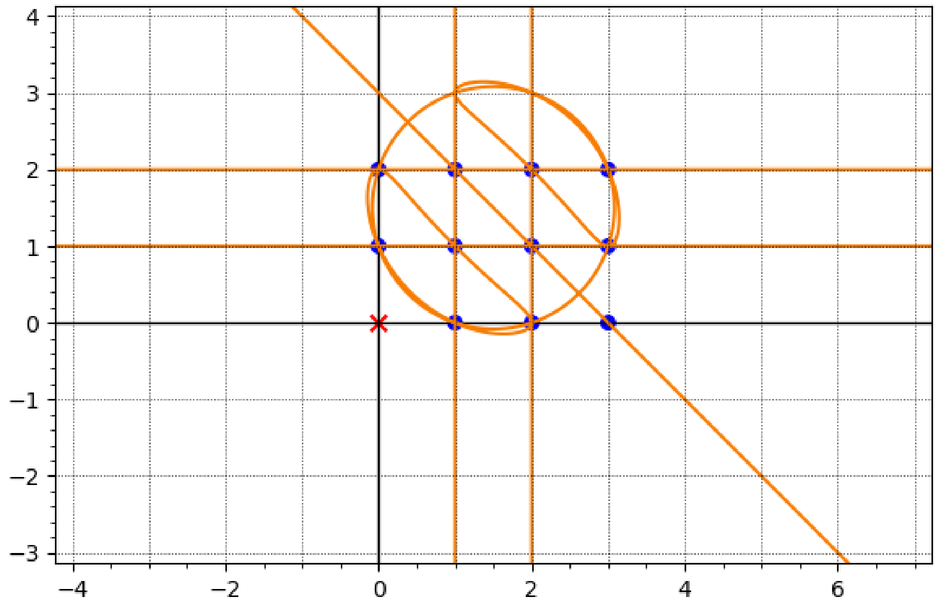


Figure A.2: $\{0, 1, 2, 3\} \times \{0, 1, 2\}$ - $\deg(f) = 11$

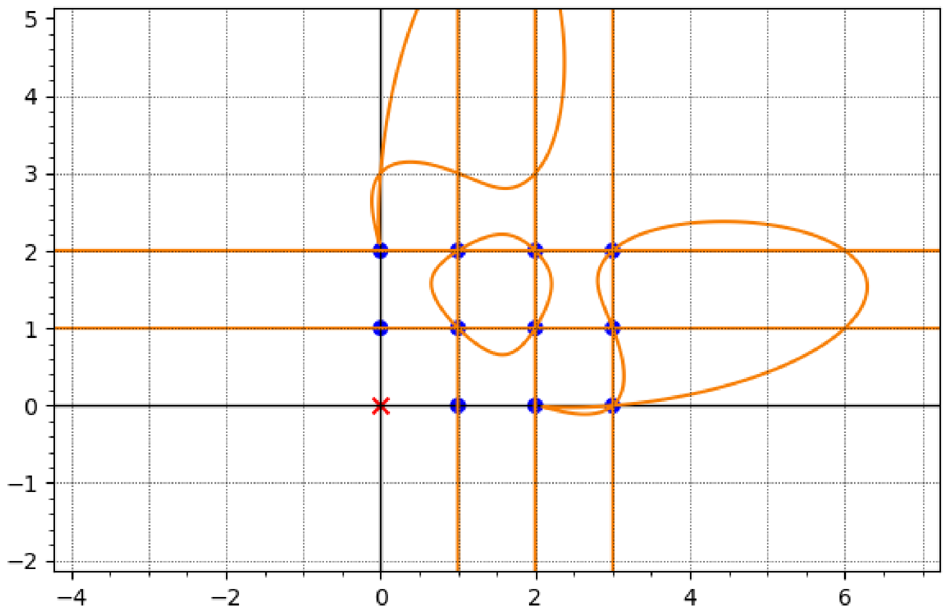


Figure A.3: $\{0, 1, 2, 3\} \times \{0, 1, 2\}$ - $\deg(f) = 11$

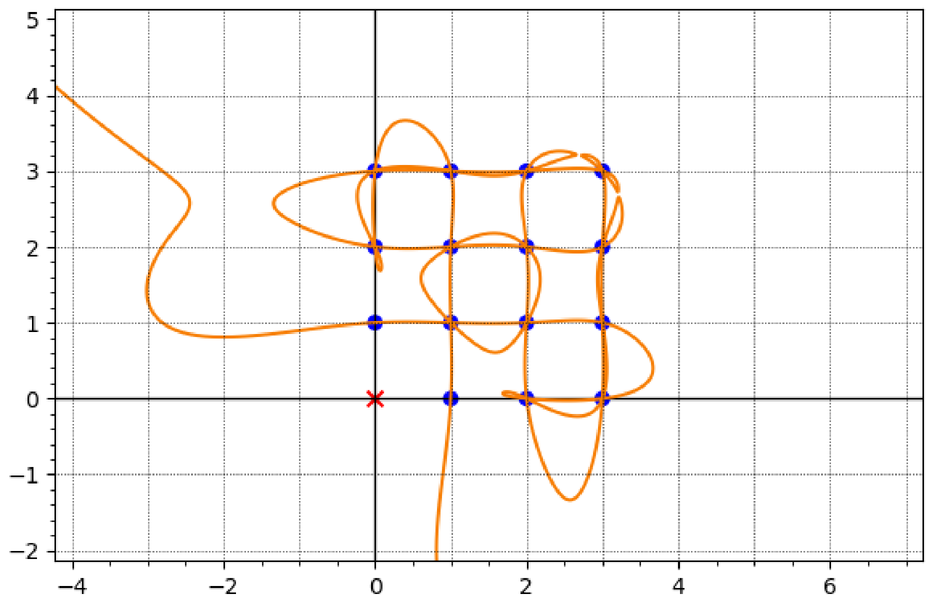


Figure A.4: $\{0, 1, 2, 3\} \times \{0, 1, 2, 3\}$ - $\deg(f) = 13$

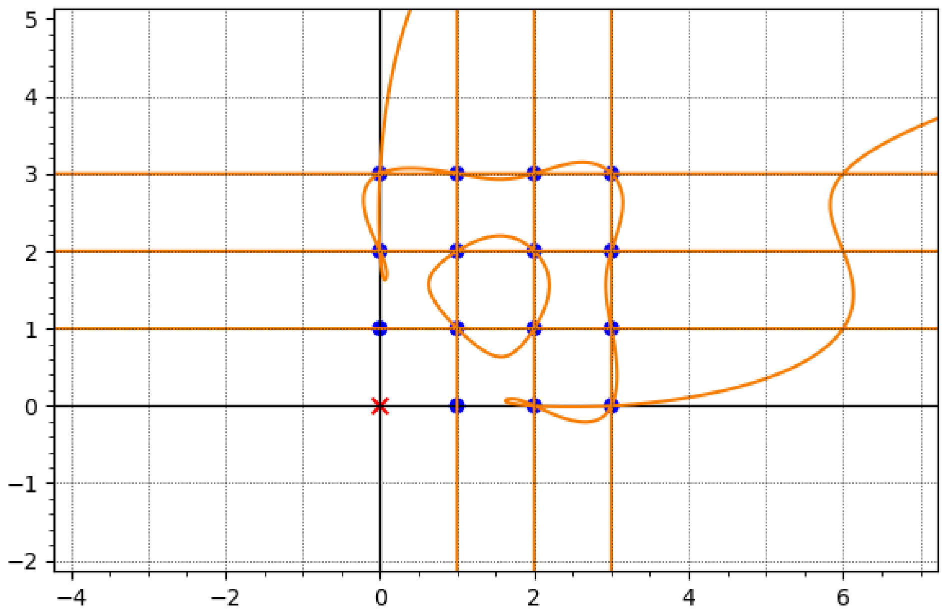


Figure A.5: $\{0, 1, 2, 3\} \times \{0, 1, 2, 3\}$ - $\deg(f) = 13$

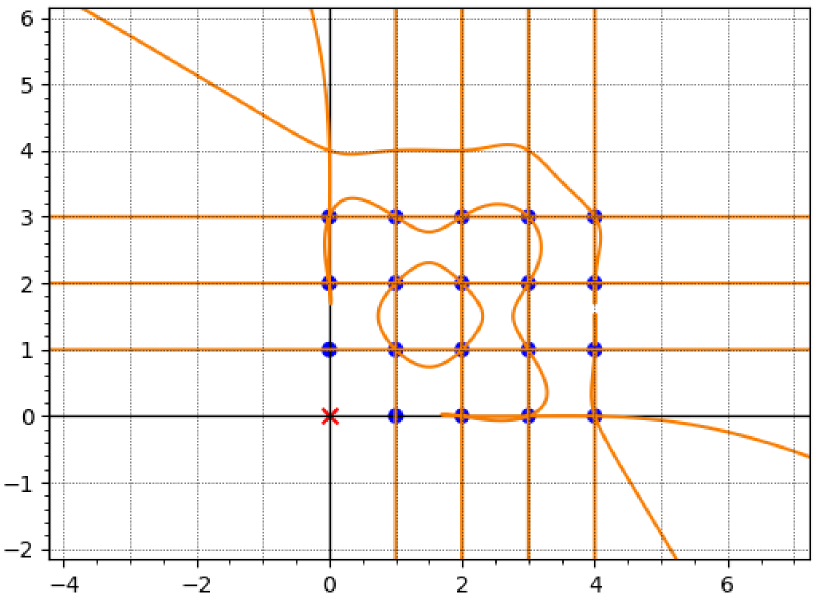


Figure A.6: $\{0,1,2,3,4\} \times \{0,1,2,3\}$ - $\deg(f) = 15$

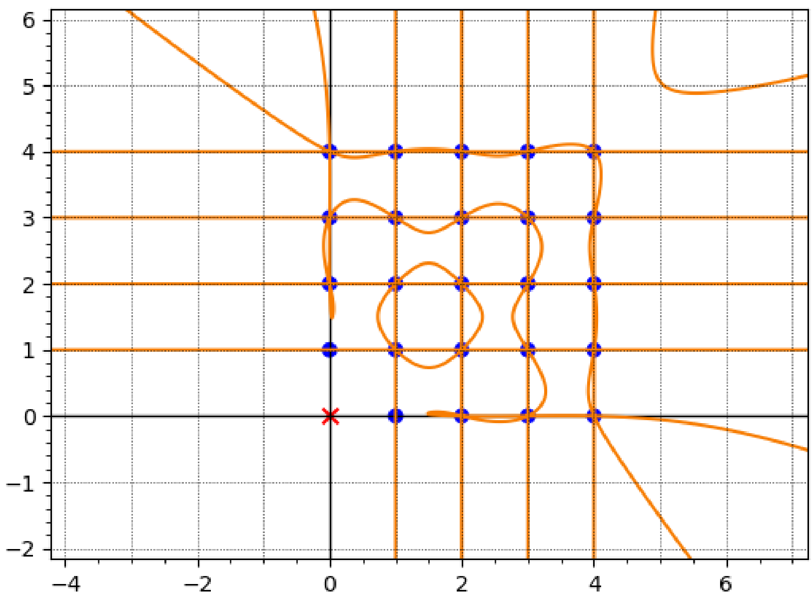


Figure A.7: $\{0,1,2,3,4\} \times \{0,1,2,3,4\}$ - $\deg(f) = 17$

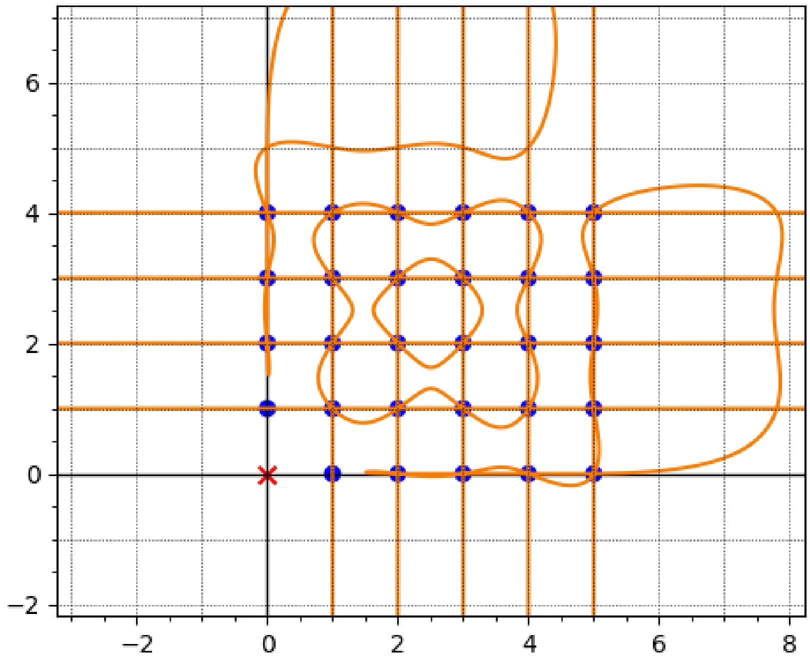


Figure A.8: $\{0,1,2,3,4,5\} \times \{0,1,2,3,4\}$ - $\deg(f) = 19$

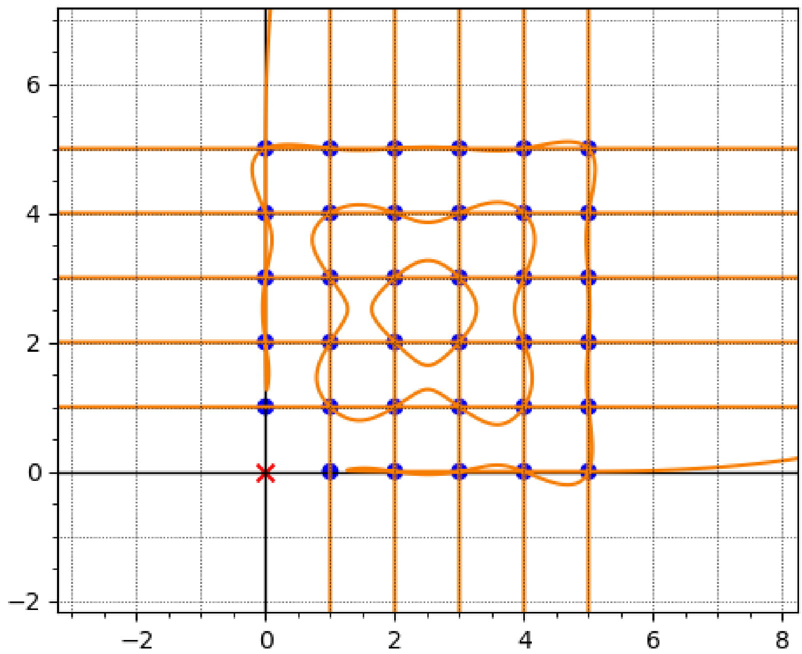


Figure A.9: $\{0,1,2,3,4,5\} \times \{0,1,2,3,4,5\}$ - $\deg(f) = 21$

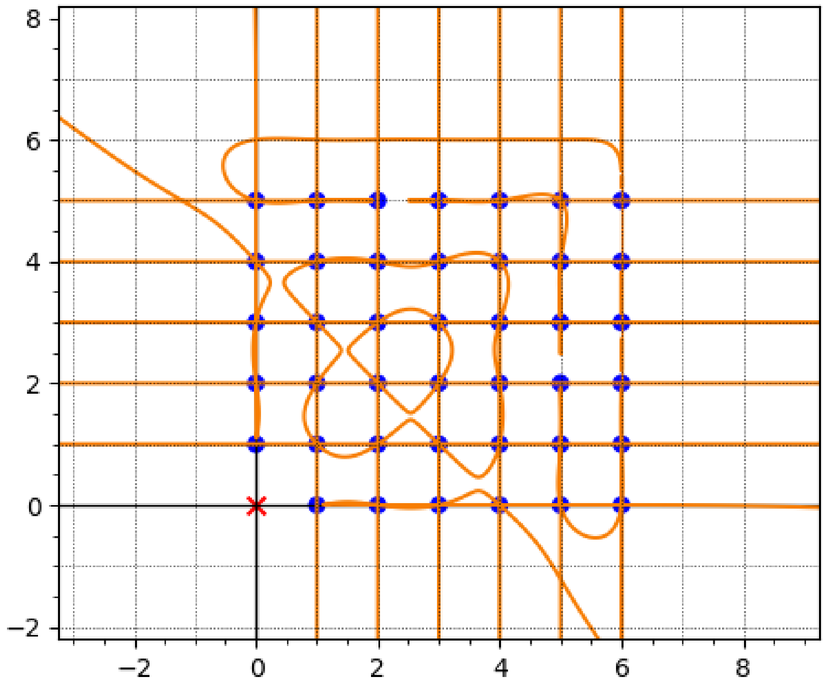


Figure A.10: $\{0,1,2,3,4,5,6\} \times \{0,1,2,3,4,5\}$ - $\deg(f) = 23$

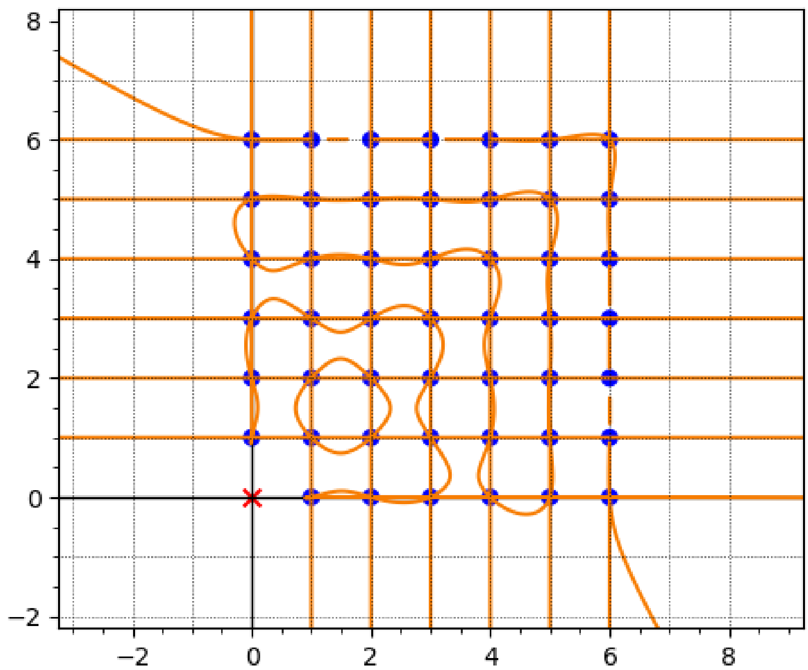


Figure A.11: $\{0,1,2,3,4,5,6\} \times \{0,1,2,3,4,5,6\}$ - $\deg(f) = 25$

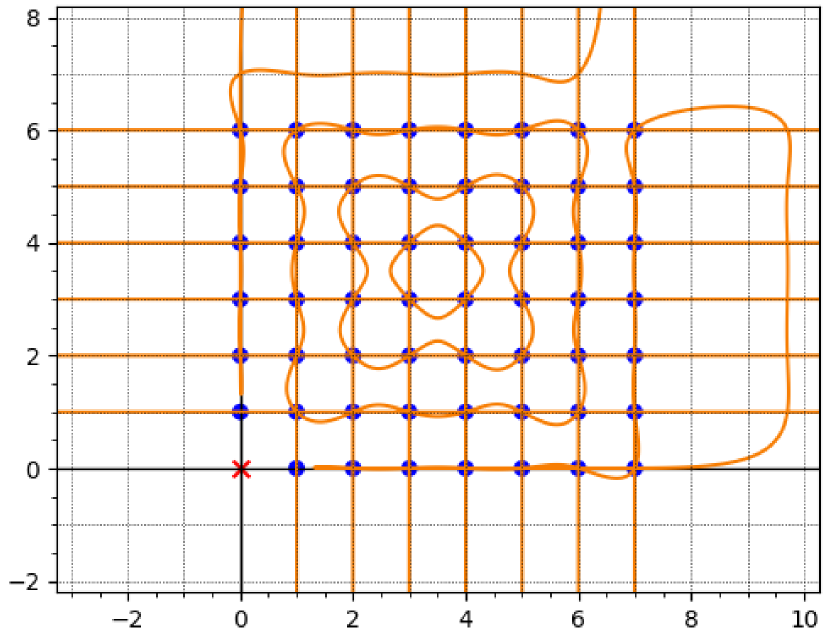


Figure A.12: $\{0, 1, 2, 3, 4, 5, 6, 7\} \times \{0, 1, 2, 3, 4, 5, 6\}$ - $\deg(f) = 27$

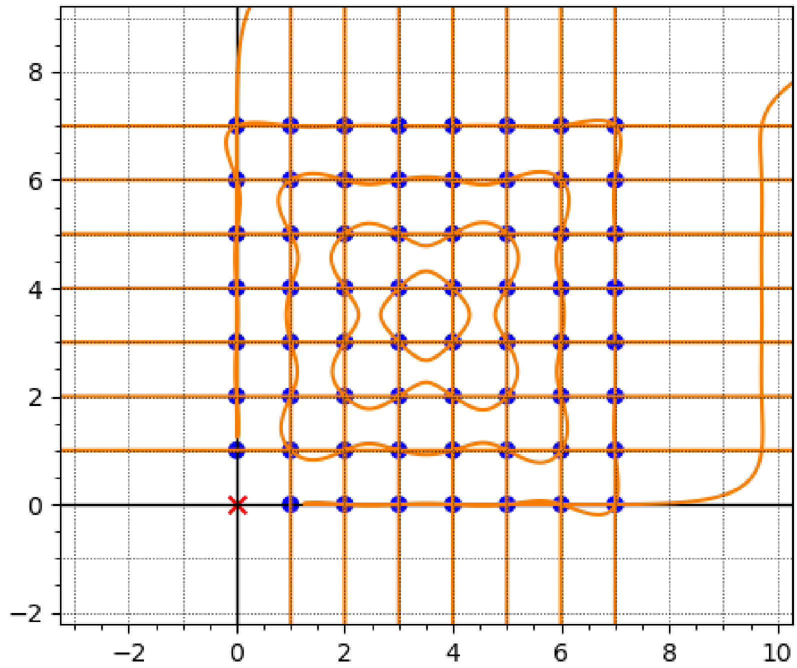


Figure A.13: $\{0, 1, 2, 3, 4, 5, 6, 7\} \times \{0, 1, 2, 3, 4, 5, 6, 8\}$ - $\deg(f) = 29$

A.2. 4-covers

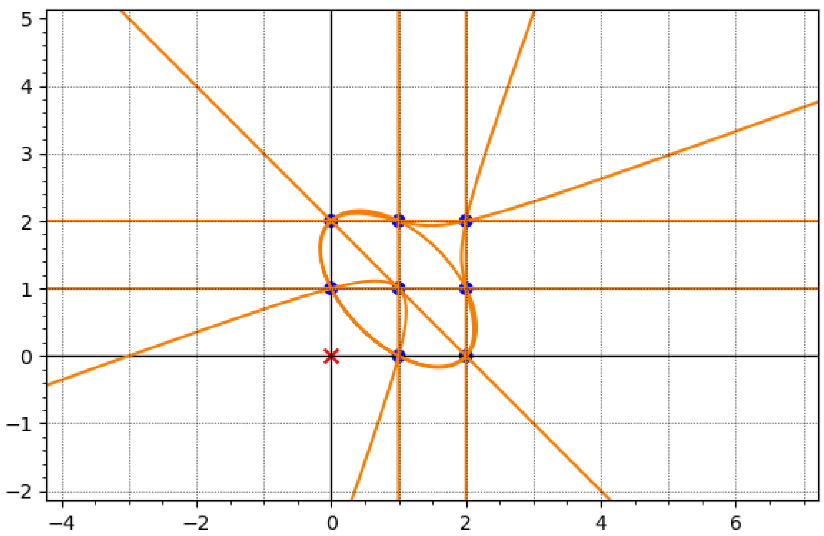


Figure A.14: $\{0,1,2\} \times \{0,1,2\}$ - $\deg(f) = 11$. This cover contains all vertical lines once.

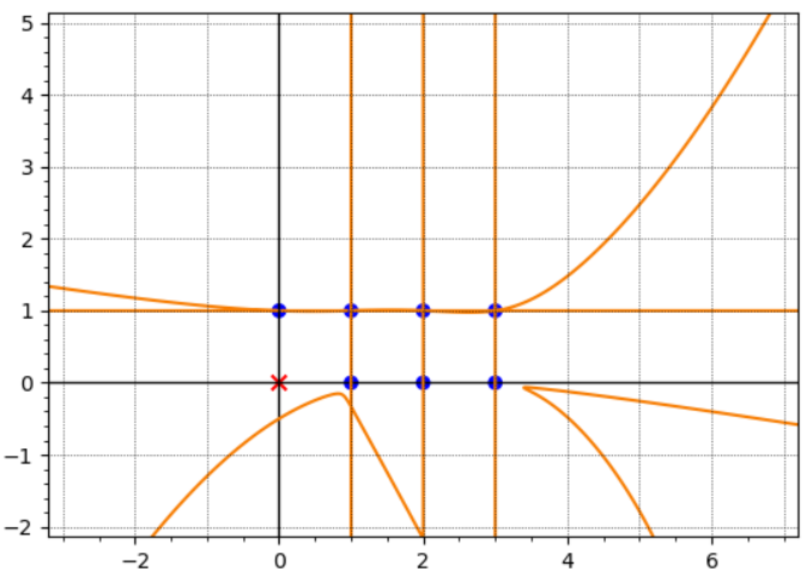


Figure A.15: $\{0,1,2,3\} \times \{0,1\}$ - $\deg(f) = 13$. This cover contains all vertical lines twice. The points on the x-axis are covered twice by the lines and twice by the other factor of the polynomial, but they are all singular points of this factor.

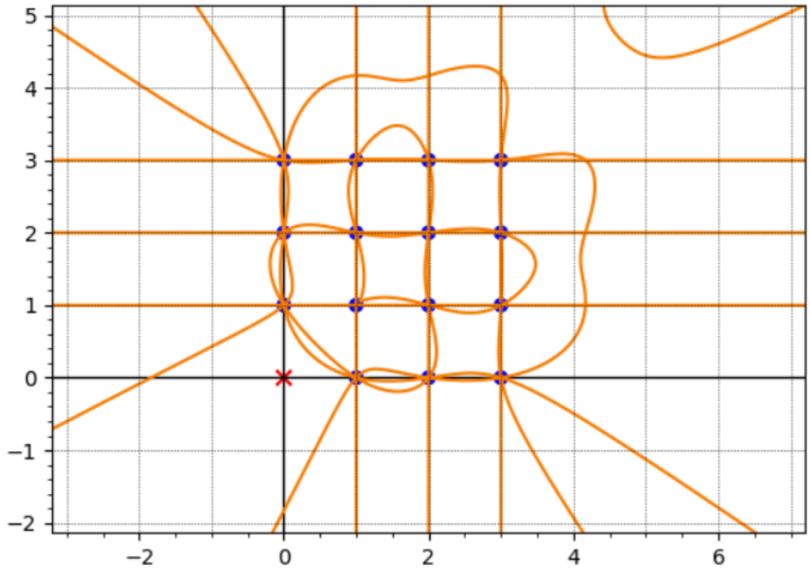


Figure A.16: $\{0,1,2,3\} \times \{0,1,2,3\}$ - $\deg(f) = 16$. This cover contains all vertical lines once.

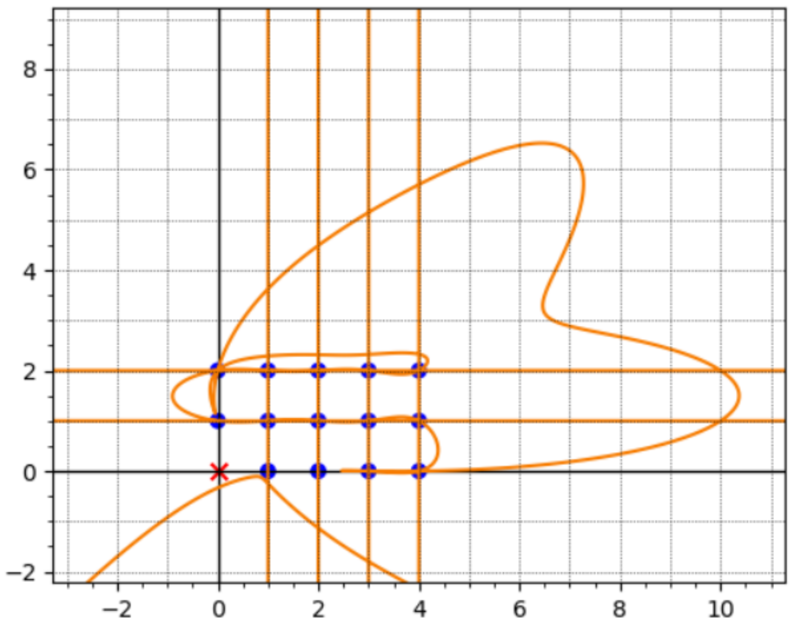


Figure A.17: $\{0,1,2,3,4\} \times \{0,1,2\}$ - $\deg(f) = 18$. This cover contains all vertical lines twice.

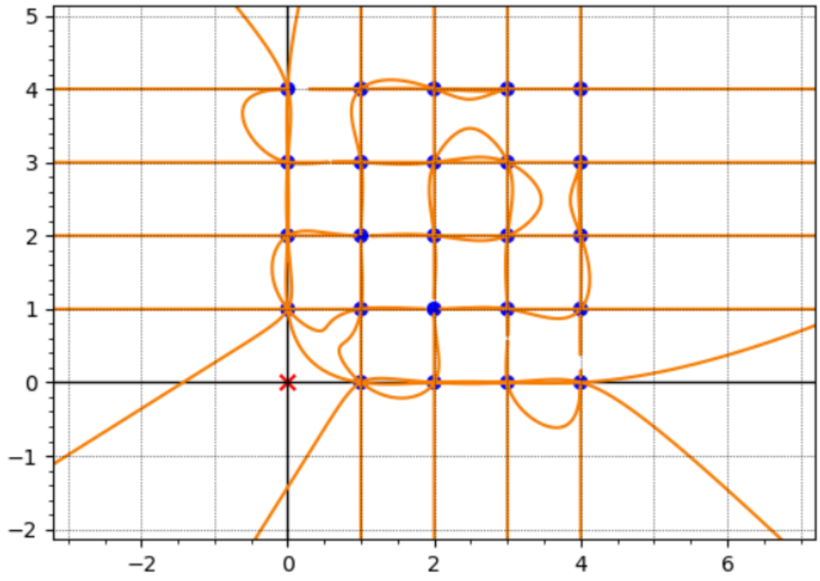


Figure A.18: $\{0, 1, 2, 3, 4\} \times \{0, 1, 2, 3, 4\}$ - $\deg(f) = 21$. This cover contains all vertical lines once.

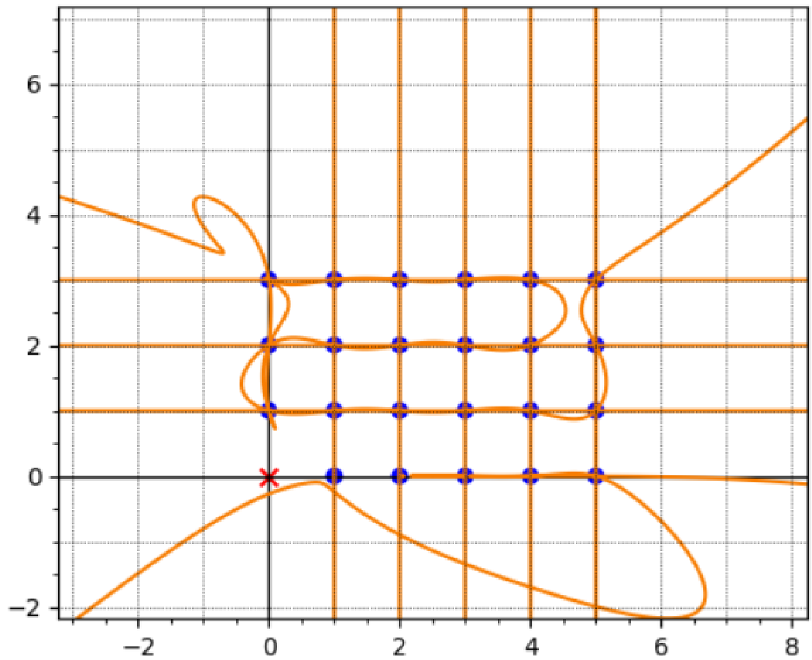


Figure A.19: $\{0, 1, 2, 3, 4, 5\} \times \{0, 1, 2, 3\}$ - $\deg(f) = 23$. This cover contains all vertical lines twice.

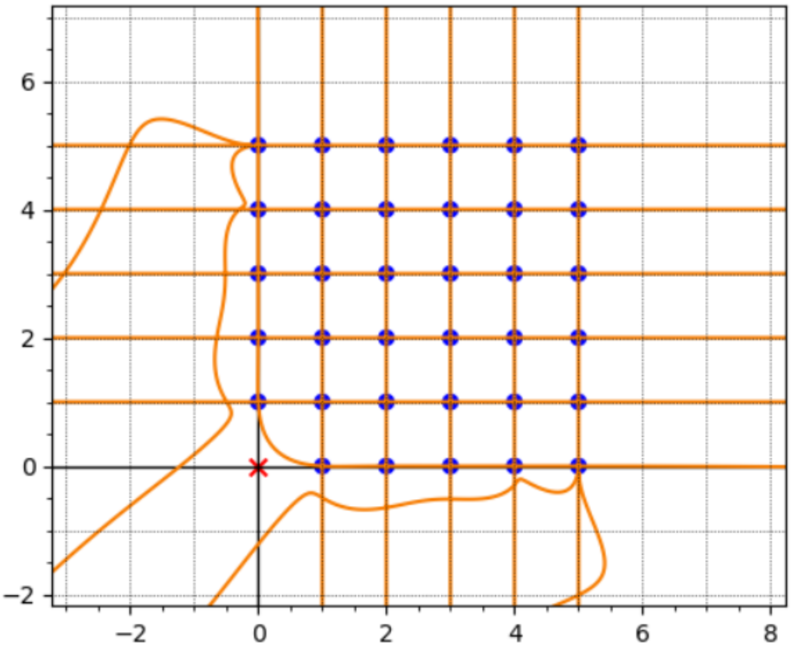


Figure A.20: $\{0,1,2,3,4,5\} \times \{0,1,2,3,4,5\}$ - $\deg(f) = 26$. This cover contains all vertical lines once. There are a lot of points that are singular points of factor of high degree of the cover

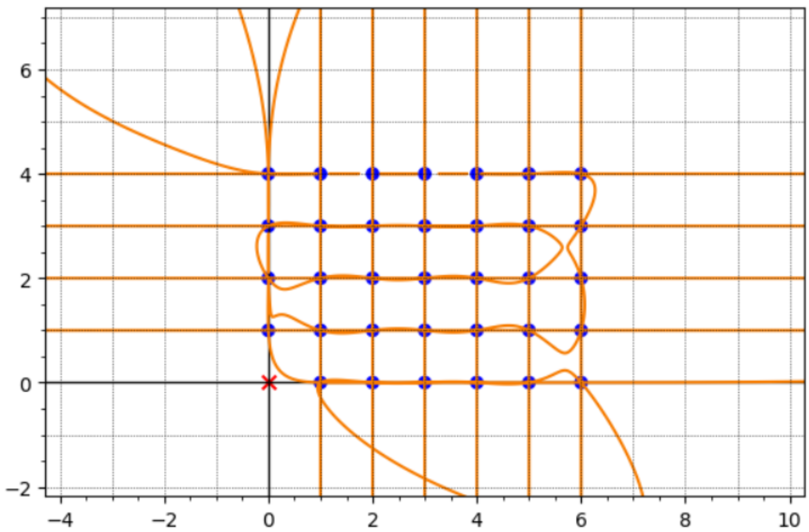


Figure A.21: $\{0,1,2,3,4,5,6\} \times \{0,1,2,3,4\}$ - $\deg(f) = 28$. This cover contains all vertical lines twice.

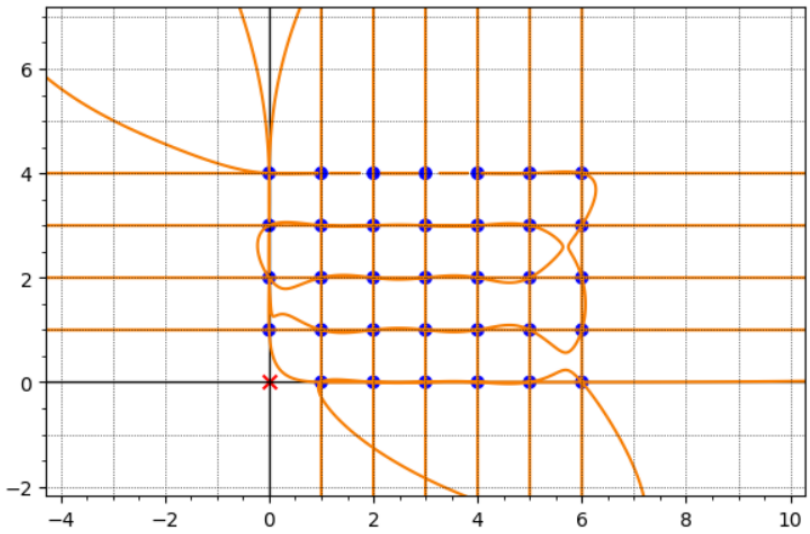


Figure A.22: $\{0, 1, 2, 3, 4, 5, 6\} \times \{0, 1, 2, 3, 4, 5, 6\}$ - $\deg(f) = 31$. This cover contains all vertical lines once.

B

Codes to generate Polyomial Covers

B.1. SymPy Code for 3-covers

```
import sympy as sp
import itertools as itertools

def gridgenerator(xlist, ylist): #generates all non-zero points of a grid
    grid = []
    for i in xlist:
        for j in ylist:
            grid.append([i,j])
    grid.remove([0,0])

    return grid


def curvemaker(xlist, ylist, deg):   #computes the polynomial 3-cover
    indices = [i for i in itertools.product(range(deg+1), repeat=2)
              if  0 < sum(i) <=deg] #Set up the polynomial of the right degree
    a = sp.IndexedBase('a')
    i = sp.Idx('i')
    coeffs = {i: a[i] for i in indices}  #define all the coefficients
    coeffslist = [a[i] for i in indices]
    coeffs[(0,0)] = 1                   #the constant term in the polynomial = 1

    points = gridgenerator(xlist, ylist)

    x,y = sp.symbols('x, y')
    p = sp.Poly(coeffs, *[x,y]) #construct the polynomial

    px = sp.diff(p,x)    #compute all partial derivatives
    py = sp.diff(p,y)
    pxx = sp.diff(px, x)
    pxy = sp.diff(px, y)
    pyy = sp.diff(py,y)

    pols = [p, px, py, pxx, pxy, pyy]
    #construct the system of linear equations
    equations = [pol.eval(point) for pol in pols for point in points]
    solutions = sp.solve(equations, *coeffslist)

    if len(solutions) == 0:
        return "no solution", "no solution", "no solution"
    else:
        freevar = {}

        for i in coeffslist:
            if (i in solutions) == False:   #Set the free variables equal to 0
                freevar[i] = 0
                solutions[i] = 0
```

```
        sols={i: solutions[coeffs[i]] for i in indices}
        sols[(0,0)] = 1

        psolved=sp.Poly(sols, *[x,y]) #construct the formula of the polynomial
        psolved = sp.Poly(psolved.subs(freevar))
        pfactor = sp.factor(psolved.as_expr())  #factorise the cover

        return solutions, psolved, pfactor   #returns the coefficients, the expanded
    polynomial and the factorised polynomial


def curvemaker_symmetric(xlist, ylist, deg):  #same function as above, but we impose that p
    is symmetric
    indices = [i for i in itertools.product(range(deg+1), repeat=2)
               if  0 < sum(i) <=deg]
    a = sp.IndexedBase('a')
    i = sp.Idx('i')
    coeffs = {i: a[i] for i in indices}
    coeffslist = [a[i] for i in indices]
    coeffs[(0,0)] = 1

    points = gridgenerator(xlist, ylist)

    x,y = sp.symbols('x, y')
    p = sp.Poly(coeffs, *[x,y])

    px = sp.diff(p,x)
    py = sp.diff(p,y)
    pxx = sp.diff(px, x)
    pxy = sp.diff(px, y)
    pyy = sp.diff(py,y)

    pols = [p, px, py, pxx, pxy, pyy]

    equations = [pol.eval(point) for pol in pols for point in points]

    for i in range(deg+1):    #add constraints to make the curve symmetric
        for j in range(deg+1):
            if i<j:
                equations.append(a[i,j]-a[j,i])

    solutions = sp.solve(equations, *coeffslist)

    if len(solutions) == 0:
        return "no solution", "no solution", "no solution"
    else:
        freevar = {}

        for i in coeffslist:
            if (i in solutions) == False:
                freevar[i] = 0
                solutions[i] = 0


        sols={i: solutions[coeffs[i]] for i in indices}
        sols[(0,0)] = 1

        psolved=sp.Poly(sols, *[x,y])
        psolved = sp.Poly(psolved.subs(freevar))
        pfactor = sp.factor(psolved.as_expr())

        return solutions, psolved, pfactor


def bound_int_generator(xlist, ylist):
    grid = []
    for i in xlist:
        for j in ylist:
            grid.append([i,j])

    boundary = [] #make a distinction between boundary and interior points
```

```
    interior = []

    for i in grid:
        if 0 in i:
            boundary.append(i)
        else:
            interior.append(i)
    boundary.remove([0,0])

    return boundary, interior


def curvemaker_lines(xlist, ylist, deg): #variation on curvemaker that assumes that the
    horizontal and vertical lines are included in the cover

    #compute the number of horizontal and vertical lines
    linedeg = len(xlist) - 1 + len(ylist) - 1
    indices = [i for i in itertools.product(range(deg - linedeg +1), repeat=2)
               if  0 < sum(i) <=deg - linedeg]
    a = sp.IndexedBase('a')
    i = sp.Idx('i')
    coeffs = {i: a[i] for i in indices}
    coeffslist = [a[i] for i in indices]
    coeffs[(0,0)] = 1

    boundary, interior = bound_int_generator(xlist, ylist)

    x,y = sp.symbols('x, y')
    p = sp.Poly(coeffs, *[x,y])

    px = sp.diff(p,x)
    py = sp.diff(p,y)
    pols = [p, px, py]

    #the derivatives only have to vanish on the boundary:
    equations = [pol.eval(point) for pol in pols for point in boundary]

    for point in interior:
        equations.append(p.eval(point))

    solutions = sp.solve(equations, *coeffslist)

    if len(solutions) == 0:
        return "no solution", "no solution", "no solution"
    else:
        freevar = {}

        for i in coeffslist:
            if (i in solutions) == False:
                freevar[i] = 0
                solutions[i] = 0

        sols={i: solutions[coeffs[i]] for i in indices}
        sols[(0,0)] = 1

        psolved=sp.Poly(sols, *[x,y])
        psolved = sp.Poly(psolved.subs(freevar))
        pfactor = sp.factor(psolved.as_expr())

        return solutions, psolved, pfactor


def curvemaker_lines_symmetric(xlist, ylist, deg):   #function to ensure that the factor
    without the lines is symmetric
    linedeg = len(xlist) - 1 + len(ylist) - 1
    indices = [i for i in itertools.product(range(deg - linedeg +1), repeat=2)
               if  0 < sum(i) <=deg - linedeg]
    a = sp.IndexedBase('a')
    i = sp.Idx('i')
    coeffs = {i: a[i] for i in indices}
    coeffslist = [a[i] for i in indices]
```

```
coeffs[(0,0)] = 1

boundary, interior = bound_int_generator(xlist, ylist)

x,y = sp.symbols('x, y')
p = sp.Poly(coeffs, *[x,y])

px = sp.diff(p,x)
py = sp.diff(p,y)

pols = [p, px, py]

equations = [pol.eval(point) for pol in pols for point in boundary]

for point in interior:
    equations.append(p.eval(point))

for i in range(deg - linedeg +1):
    for j in range(deg - linedeg +1):
        if i<j:
            equations.append(a[i,j]-a[j,i])

solutions = sp.solve(equations, *coeffslist)

if len(solutions) == 0:
    return "no solution", "no solution", "no solution"
else:
    freevar = {}

    for i in coeffslist:
        if (i in solutions) == False:
            freevar[i] = 0
            solutions[i] = 0

    sols={i: solutions[coeffs[i]] for i in indices}
    sols[(0,0)] = 1

    psolved=sp.Poly(sols, *[x,y])
    psolved = sp.Poly(psolved.subs(freevar))
    pfactor = sp.factor(psolved.as_expr())

    return solutions, psolved, pfactor
```

B.2. Sage Code for 3-covers

```
import itertools as itertools
import numpy as np

def gridgenerator(xlist, ylist):  #generates all non-zero points of a grid
    grid = []
    for i in xlist:
        for j in ylist:
            grid.append([i,j])

    grid.remove([0,0])

    return grid


#Evaluate a point (x,y) in each monomial and construct a row filled with these evaluated
    monomials
def evp(x,y, indices):
    row = np.zeros(len(indices))

    for i in range(len(indices)):
        j = indices[i]
        row[i] = x**(j[0]) * y**(j[1])

    return row


#Evaluate a point (x,y) in the monomials after having taken the derivative wrt x
def evp_x(x,y, indices):
    row = np.zeros(len(indices))

    for i in range(len(indices)):
        j = indices[i]

        if j[0] != 0:
            row[i] = j[0] * x**(j[0]-1) * y**(j[1])

    return row


#Evaluate a point (x,y) in the monomials after having taken the derivative wrt x
def evp_y(x,y, indices):
    row = np.zeros(len(indices))

    for i in range(len(indices)):
        j = indices[i]

        if j[1] != 0:
            row[i] = j[1] * x**(j[0]) * y**(j[1]-1)

    return row


#Evaluate a point (x,y) in the monomials after having taken the second order derivative wrt
    xx
def evp_xx(x,y, indices):
    row = np.zeros(len(indices))

    for i in range(len(indices)):
        j = indices[i]
        if j[0] > 1:
            row[i] = j[0]*(j[0]-1) * x**(j[0]-2) * y**(j[1])

    return row


#Evaluate a point (x,y) in the monomials after having taken the second order derivative wrt
    yy
def evp_yy(x,y,indices):
    row = np.zeros(len(indices))
```

```
    for i in range(len(indices)):
        j = indices[i]
        if j[1] > 1:
            row[i] = j[1]*(j[1]-1) * x**(j[0]) * y**(j[1]-2)

    return row


#Evaluate a point (x,y) in the monomials after having taken the second order derivative wrt
    xy
def evp_xy(x,y, indices):
    row = np.zeros(len(indices))

    for i in range(len(indices)):
        j = indices[i]
        if j[0] != 0 and j[1] != 0:
            row[i] = j[0]*j[1] * x**(j[0]-1) * y**(j[1]-1)

    return row


#This function makes the system of linear equations based on the above functions
def matrixgenerator(xlist, ylist, deg):
    #Construct the monomials of the right degrees
    indices = [i for i in itertools.product(range(deg+1), repeat=2) if  0 <= sum(i) <= deg]

    #Place the constant term as final index in the list such that it becomes the last column
    in the system
    indices.remove((0,0))
    indices.append((0,0))

    grid = gridgenerator(xlist, ylist)
    print(grid)

    eqcount = 6*len(grid) #number of equations in the system for a 3-cover
    indcount = len(indices) #number of monomials

    syst = np.zeros((eqcount, indcount))

    #Constructing the matrix:
    teller = 0
    for j in grid:
        syst[teller, :] = evp(j[0], j[1], indices)
        teller = teller + 1

        syst[teller, :] = evp_x(j[0], j[1], indices)
        teller = teller + 1

        syst[teller, :] = evp_y(j[0], j[1], indices)
        teller = teller + 1

        syst[teller, :] = evp_xx(j[0], j[1], indices)
        teller = teller + 1

        syst[teller, :] = evp_xy(j[0], j[1], indices)
        teller = teller + 1

        syst[teller, :] = evp_yy(j[0], j[1], indices)
        teller = teller + 1

    #The domain of the matrix is ”ZZ”, i.e. the rational field
    syst = Matrix(QQ, syst)

    return syst, syst.rref(), syst.pivot_rows(), syst.pivots()


#Use the solution of the system to construct the corresponding polynomial
def curvegenerator(xlist, ylist, deg):
    indices = [i for i in itertools.product(range(deg+1), repeat=2) if  0 <= sum(i) <= deg]
    indices.remove((0,0))
```

```
    indices.append((0,0))

    syst, matrix, rows, pivot = matrixgenerator(xlist, ylist, deg)
    x,y = var('x, y')
    p = -1

    for i in range(len(pivot)):
        p =  p + matrix[rows[i], -1] * x**(indices[pivot[i]][0]) * y**(indices[pivot[i]][1])

    return p


#generate the non-zero points and make a distinction between interior and boundary points
def bound_int_generator(xlist, ylist):
    grid = []
    for i in xlist:
        for j in ylist:
            if i >= j:
                grid.append([i,j])

    boundary = []
    interior = []

    for i in grid:
        if 0 in i:
            boundary.append(i)
        else:
            interior.append(i)
    boundary.remove([0,0])

    return boundary, interior


##Evaluate a point (x,y) in each monomial and construct a row filled with the evaluated sums
    of symmetric monomials
def evp_symmetric(x,y, indices):
    row = np.zeros(len(indices))

    for i in range(len(indices)):
        j = indices[i]
        if j[0] == j[1]:        #All monomials that are symmetric to itself
            row[i] = x**(j[0]) * y**(j[1])
        else:                   #Sums of other monomials
            row[i] = x**(j[0]) * y**(j[1]) + x**(j[1]) * y**(j[0])

    return row


##Evaluate a point (x,y) in all sums of symmetric monomials after having taken the derivative
     wrt x
def evp_x_symmetric(x,y, indices):
    row = np.zeros(len(indices))

    for i in range(len(indices)):
        j = indices[i]
        if j[0] == j[1]:  #Symmetric to itself
            if j[0] != 0:
                row[i] = j[0] * x**(j[0]-1) * y**(j[1])
        else:
            if j[0] != 0 and j[1] != 0:  #None of the powers is zero
                row[i] = j[0] * x**(j[0]-1) * y**(j[1]) + j[1] * x**(j[1]-1) * y**(j[0])
            elif j[0] == 0 and j[1] != 0:  #Power of x is zero
                row[i] = j[1] * x**(j[1]-1) * y**(j[0])
            elif j[0] != 0 and j[1] == 0:  #Power of y is zero
                row[i] = j[0] * x**(j[0]-1) * y**(j[1])

    return row
```

```
##Evaluate a point (x,y) in all sums of symmetric monomials after having taken the derivative
     wrt y
def evp_y_symmetric(x,y, indices):
    row = np.zeros(len(indices))

    for i in range(len(indices)):
        j = indices[i]
        if j[0] == j[1]:
            if j[1] != 0:
                row[i] = j[1] * x**(j[0]) * y**(j[1]-1)

        else:
            if j[0] != 0 and j[1] != 0:
                row[i] = j[1] * x**(j[0]) * y**(j[1]-1) + j[0] * x**(j[1]) * y**(j[0]-1)
            elif j[0] == 0 and j[1] != 0:
                row[i] = j[1] * x**(j[0]) * y**(j[1]-1)
            elif j[0] != 0 and j[1] ==0:
                row[i] = j[0] * x**(j[1]) * y**(j[0]-1)

    return row

#Same function as the matrixgenerator above, but now for symmetric polynomials
def matrixgenerator_symmetric(xlist, ylist, deg):
    indices = [i for i in itertools.product(range(deg+1), repeat=2) if  0 <= sum(i) <= deg
    and i[1] <= i[0]]

    indices.remove((0,0))
    indices.append((0,0))

    boundary, interior = bound_int_generator(xlist, ylist)

    eqcount = len(interior) + 3* len(boundary)
    indcount = len(indices)

    syst = np.zeros((eqcount,indcount))
    teller = 0

    for j in boundary:
        syst[teller,:] = evp_symmetric(j[0], j[1], indices)
        teller = teller + 1

        syst[teller,:] = evp_x_symmetric(j[0],j[1], indices)
        teller = teller + 1

        syst[teller,:] = evp_y_symmetric(j[0],j[1], indices)
        teller = teller + 1

    for i in interior:
        syst[teller,:] = evp_symmetric(i[0], i[1], indices)
        teller = teller + 1

    syst = Matrix(QQ, syst)

    return syst, syst.rref(), syst.pivot_rows(), syst.pivots()

#Same function as the curvegenerator above, but now for symmetric polynomials
def curvegenerator_symmetric(xlist, ylist, deg):
    indices = [i for i in itertools.product(range(deg+1), repeat=2) if  0 <= sum(i) <= deg
    and i[1] <= i[0]]
    indices.remove((0,0))
    indices.append((0,0))

    syst, matrix, rows, pivot = matrixgenerator_symmetric(xlist, ylist, deg)
    x,y = var('x, y')
    p = -1

    for i in range(len(pivot)):
        if indices[pivot[i]][0] == indices[pivot[i]][1]:
            p =  p + matrix[rows[i], -1] * x**(indices[pivot[i]][0]) *
             y**(indices[pivot[i]][1])
```

```
        else:
            p = p + matrix[rows[i], -1] * (x**(indices[pivot[i]][0]) *
                y**(indices[pivot[i]][1]) + x**(indices[pivot[i]][1]) *
                y**(indices[pivot[i]][0]))

    return p
```

B.3. SymPy Code for 4-covers

```
import sympy as sp
import itertools as itertools

def gridgenerator(xlist, ylist): #generate all non-zero point of a grid
    grid = []
    for i in xlist:
        for j in ylist:
            grid.append([i,j])
    grid.remove([0,0])

    return grid


def curvemaker(xlist, ylist, deg):   #computes the polynomial 4-cover
    indices = [i for i in itertools.product(range(deg+1), repeat=2)
              if  0 < sum(i) <=deg] #Set up the polynomial of the right degree
    a = sp.IndexedBase('a')
    i = sp.Idx('i')
    coeffs = {i: a[i] for i in indices}  #define all the coefficients
    coeffslist = [a[i] for i in indices]
    coeffs[(0,0)] = 1

    points = gridgenerator(xlist, ylist)

    x,y = sp.symbols('x, y')
    p = sp.Poly(coeffs, *[x,y]) #construct the polynomial

    px = sp.diff(p,x)   #compute all partial derivatives
    py = sp.diff(p,y)
    pxx = sp.diff(px, x)
    pxy = sp.diff(px, y)
    pyy = sp.diff(py,y)
    pxxx = sp.diff(pxx, x)
    pxxy = sp.diff(pxx, y)
    pxyy = sp.diff(pxy, y)
    pyyy = sp.diff(pyy, y)

    pols = [p, px, py, pxx, pxy, pyy, pxxx, pxxy, pxyy, pyyy]
    #construct the system of linear equations
    equations = [pol.eval(point) for pol in pols for point in points]
    solutions = sp.solve(equations, *coeffslist)

    if len(solutions) == 0:
        return "no solution", "no solution", "no solution"
    else:
        freevar = {}

        for i in coeffslist:
            if (i in solutions) == False:   #Set the free variables to 0
                freevar[i] = 0
                solutions[i] = 0

        sols={i: solutions[coeffs[i]] for i in indices}
        sols[(0,0)] = 1

        psolved=sp.Poly(sols, *[x,y]) #construct the formula of the polynomial
        psolved = sp.Poly(psolved.subs(freevar))
        pfactor = sp.factor(psolved.as_expr())  #factorise the cover

        return solutions, psolved, pfactor

#generate the non-zero points and make a distinction between interior and boundary points
def bound_int_generator(xlist, ylist):
    grid = []
    for i in xlist:
        for j in ylist:
            grid.append([i,j])
```

```
    boundary = []
    interior = []

    for i in grid:
        if 0 in i:
            boundary.append(i)
        else:
            interior.append(i)
    boundary.remove([0,0])

    return boundary, interior


def curvemaker_oneline(xlist, ylist, deg): #computes the polynomial 4-cover assuming all
    horizontal and vertical lines to be included once in the cover

    #compute the number of horizontal and vertical lines:
    linedeg = len(xlist) - 1 + len(ylist) - 1
    indices = [i for i in itertools.product(range(deg - linedeg +1), repeat=2)
               if  0 < sum(i) <=deg - linedeg]
    a = sp.IndexedBase('a')
    i = sp.Idx('i')
    coeffs = {i: a[i] for i in indices}
    coeffslist = [a[i] for i in indices]
    coeffs[(0,0)] = 1

    boundary, interior = bound_int_generator(xlist, ylist)

    x,y = sp.symbols('x, y')
    p = sp.Poly(coeffs, *[x,y])

    px = sp.diff(p,x)
    py = sp.diff(p,y)
    pxx = sp.diff(px, x)
    pxy = sp.diff(px, y)
    pyy = sp.diff(py,y)
    pols = [p, px, py, pxx, pxy, pyy]

    #the second derivatives only have to vanish on the boundary:
    equations = [pol.eval(point) for pol in pols for point in boundary]

    for point in interior:
        equations.append(p.eval(point))
        equations.append(px.eval(point))
        equations.append(py.eval(point))

    solutions = sp.solve(equations, *coeffslist)

    if len(solutions) == 0:
        return "no solution", "no solution", "no solution"
    else:
        freevar = {}

        for i in coeffslist:
            if (i in solutions) == False:
                freevar[i] = 0
                solutions[i] = 0

        sols={i: solutions[coeffs[i]] for i in indices}
        sols[(0,0)] = 1

        psolved=sp.Poly(sols, *[x,y])
        psolved = sp.Poly(psolved.subs(freevar))
        pfactor = sp.factor(psolved.as_expr())

        return solutions, psolved, pfactor
```

```
#generate the non-zero points and make a distinction between interior points and points on
    the two axes
def x_y_generator(xlist, ylist):
    grid = []
    for i in xlist:
        for j in ylist:
            grid.append([i,j])

    x_boundary = []
    y_boundary = []
    interior = []

    grid.remove([0,0])

    for i in grid:
        if 0 == i[0]:
            y_boundary.append(i)
        elif 0 == i[1]:
            x_boundary.append(i)
        else:
            interior.append(i)

    return x_boundary, y_boundary, interior


def curvemaker_lines(xlist, ylist, deg): #computes the polynomial 4-cover assuming all
    horizontal to be included once and vertical lines to be included twice in the cover

    #compute the number of horizontal and vertical lines:
    linedeg = 2*(len(xlist) - 1) + len(ylist) - 1
    print(linedeg)
    indices = [i for i in itertools.product(range(deg - linedeg +1), repeat=2)
               if  0 < sum(i) <=deg - linedeg]
    print(indices)
    a = sp.IndexedBase('a')
    i = sp.Idx('i')
    coeffs = {i: a[i] for i in indices}
    coeffslist = [a[i] for i in indices]
    coeffs[(0,0)] = 1

    x_boundary, y_boundary, interior = x_y_generator(xlist, ylist)

    x,y = sp.symbols('x, y')
    p = sp.Poly(coeffs, *[x,y])

    px = sp.diff(p,x)
    py = sp.diff(p,y)
    pxx = sp.diff(px, x)
    pxy = sp.diff(px, y)
    pyy = sp.diff(py,y)
    pols = [p, px, py, pxx, pxy, pyy]

    #the highest derivatives only have to vanish on the y-axis:
    equations = [pol.eval(point) for pol in pols for point in y_boundary]

    for point in x_boundary:
        equations.append(p.eval(point))
        equations.append(px.eval(point))
        equations.append(py.eval(point))

    for point in interior:
        equations.append(p.eval(point))

    solutions = sp.solve(equations, *coeffslist)

    if len(solutions) == 0:
        return "no solution", "no solution", "no solution"
    else:
        freevar = {}
```

```
    for i in coeffslist:
        if (i in solutions) == False:
            freevar[i] = 0
            solutions[i] = 0

    sols={i: solutions[coeffs[i]] for i in indices}
    sols[(0,0)] = 1

    psolved=sp.Poly(sols, *[x,y])
    psolved = sp.Poly(psolved.subs(freevar))
    pfactor = sp.factor(psolved.as_expr())

    return solutions, psolved, pfactor
```