On Semialgebraic Range Reporting

Peyman Afshani ✉
Aarhus University, Denmark

Pingan Cheng ✉
Aarhus University, Denmark

Abstract

Semialgebraic range searching, arguably the most general version of range searching, is a fundamental problem in computational geometry. In the problem, we are to preprocess a set of points in $\mathbb{R}^D$ such that the subset of points inside a semialgebraic region described by a constant number of polynomial inequalities of degree Δ can be found efficiently.

Relatively recently, several major advances were made on this problem. Using algebraic techniques, "near-linear space" data structures [6, 18] with almost optimal query time of $Q(n) = O(n^{1-1/D+o(1)})$ were obtained. For "fast query" data structures (i.e., when $Q(n) = n^{o(1)}$), it was conjectured that a similar improvement is possible, i.e., it is possible to achieve space $S(n) = O(n^{D+o(1)})$. The conjecture was refuted very recently by Afshani and Cheng [3]. In the plane, i.e., $D = 2$, they proved that $S(n) = \Omega(n^{\Delta+1-o(1)}/Q(n)^{(\Delta+3)\Delta/2})$ which shows $\Omega(n^{\Delta+1-o(1)})$ space is needed for $Q(n) = n^{o(1)}$. While this refutes the conjecture, it still leaves a number of unresolved issues: the lower bound only works in 2D and for fast queries, and neither the exponent of n or $Q(n)$ seem to be tight even for $D = 2$, as the best known upper bounds have $S(n) = O(n^{\boldsymbol{m}+o(1)}/Q(n)^{(\boldsymbol{m}-1)D/(D-1)})$ where $\boldsymbol{m} = \binom{D+\Delta}{D} - 1 = \Omega(\Delta^D)$ is the maximum number of parameters to define a monic degree-Δ D-variate polynomial, for any constant dimension D and degree Δ.

In this paper, we resolve two of the issues: we prove a lower bound in D-dimensions, for constant D, and show that when the query time is $n^{o(1)} + O(k)$, the space usage is $\Omega(n^{\boldsymbol{m}-o(1)})$, which almost matches the $\tilde{O}(n^{\boldsymbol{m}})$ upper bound and essentially closes the problem for the fast-query case, as far as the exponent of n is considered in the pointer machine model. When considering the exponent of $Q(n)$, we show that the analysis in [3] is tight for $D = 2$, by presenting matching upper bounds for uniform random point sets. This shows either the existing upper bounds can be improved or to obtain better lower bounds a new fundamentally different input set needs to be constructed.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Computational Geometry, Range Searching, Data Structures and Algorithms, Lower Bounds

Digital Object Identifier 10.4230/LIPIcs.SoCG.2022.3

Related Version *Full Version*: https://arxiv.org/abs/2203.07096

Funding Supported by DFF (Det Frie Forskningsråd) of Danish Council for Independent Research under grant ID DFF−7014−00404.

1 Introduction

In the classical semialgebraic range searching problem, we are to preprocess a set of n points in $\mathbb{R}^D$ such that the subset of points inside a semialgebraic region, described by a constant number of polynomial inequalities of degree Δ can be found efficiently. Recently, two major advances were made on this problem. First, in 2019, Agarwal et al. [5] showed for polylogarithmic query time, it is possible to build a data structure of size $\tilde{O}(n^\beta)$ space[1], where β is the number of parameters needed to specify a query polynomial. For example, for $D = 2$,

[1] $\tilde{\Omega}(\cdot), \tilde{O}(\cdot), \tilde{\Theta}(\cdot)$ notations hide $\log^{o(1)} n$ factors; $\mathring{\Omega}(\cdot), \mathring{O}(\cdot), \mathring{\Theta}(\cdot)$ notations hide $n^{o(1)}$ factors.

a query polynomial is in the form of $\sum_{i+j\le\Delta} a_{ij}x^iy^j \le 0$ where a_{ij}'s are specified at the query time, and when $\Delta = 4$, β can be as large as 14 (technically, there are 15 coefficients but one coefficient can always be normalized to be 1). In this case, a major conjecture was that if this space bound could be improved to $\tilde{O}(n^D)$ (e.g., for $\Delta = 4$, from $\tilde{O}(n^{14})$ to $\tilde{O}(n^2)$). Very recently, Afshani and Cheng [3] refuted this conjecture by showing an $\overset{o}{\Omega}(n^{\Delta+1})$ lower bound. However, there are two major limitations of their lower bound. First, their lower bound only works in $\mathbb{R}^2$, while the upper bound in [5] holds for all dimensions. Second, their lower bound only works for queries of form $y - \sum_{i=0}^{\Delta} x^i \le 0$ and thus their lower bound does not give a satisfactory answer to the problem in the general case. For example, for $D = 2, \Delta = 4$, they show a $\overset{o}{\Omega}(n^5)$ lower bound whereas the current best upper bound is $\tilde{O}(n^{14})$. In general, their space lower bound is at most $\overset{o}{\Omega}(n^{\Delta+1})$ while the upper bound of [5] can be $\tilde{O}(n^{\Theta(\Delta^2)})$, which leaves an unsolved wide gap, even for $D = 2$. Another problem brought by [5] is the space-time tradeoff. When restricted to queries of the form $y - \sum_{i=0}^{\Delta} x^i \le 0$, the current upper bound tradeoff is $S(n) = \tilde{O}(n^{\Delta+1}/Q(n)^{2\Delta})$ [18, 5] while the lower bound in [3] is $S(n) = \overset{o}{\Omega}(n^{\Delta+1}/Q(n)^{(\Delta+3)\Delta/2})$. Even for $\Delta = 2$, we observe a discrepancy between an $S(n) = \tilde{O}(n^3/Q(n)^4)$ upper and an $S(n) = \overset{o}{\Omega}(n^3/Q(n)^5)$ lower bound.

Here, we make progress in both lower and upper bound directions. We give a general lower bound in D dimensions that is tight for all possible values of β. Our lower bound attains the maximum possible β value $\boldsymbol{m}_{D,\Delta} = \binom{D+\Delta}{D} - 1$, e.g., $\overset{o}{\Omega}(n^{14})$ for $D = 2, \Delta = 4$. Thus, our lower bounds almost completely settle the general case of the problem for the fast-query case, as far as the exponent of n is concerned. This improvement is quite non-trivial and requires significant new insights that are not avaiable in [3]. For the upper bound, we present a matching space-time tradeoff for the two problems studied in [3] for uniform random point sets. This shows their lower bound analysis is tight. Since for most range searching problems, a uniform random input instance is the hardest one, our results show that current upper bound based on the classical method might not be optimal. We develop a set of new ideas for our results which we believe are important for further investigation of this problem.

1.1 Background

In range searching, the input is a set of points in $\mathbb{R}^D$ for a fixed constant D. The goal is to build a structure such that for a query range, we can report or find the points in the range efficiently. This is a fundamental problem in computational geometry with many practical uses in e.g., databases and GIS systems. For more information, see surveys by Agarwal [14] or Matoušek [17]. We focus on a fundamental case of the problem where the ranges are semialgebraic sets of constant complexity which are defined by intersection/union/complementation of $O(1)$ polynomial inequalities of constant degree at most Δ in $\mathbb{R}^D$.

The study of this problem dates back to at least 35 years ago [19]. A linear space and $O(n^{1-1/D+o(1)})$ query time structure is given by Agarwal, Matoušek, and Sharir [6], due to the recent "polynomial method" breakthrough [15]. However, it is not entirely clear what happens to the "fast-query" case: if we insist on polylogarithmic query time, what is the smallest possible space usage? Early on, some believed that the number of parameters plays an important role and thus $\tilde{O}(n^\beta)$ space could be a reasonable conjecture [17], but such a data structure was not found until 2019 [5]. However, after the "polynomial method" revolution, and specifically after the breakthrough result of Agarwal, Matoušek and Sharir [6], it could also be reasonably conjectured that $\tilde{O}(n^D)$ could also be the right bound. However, this was refuted recently by Afshani and Cheng [3] who showed that in 2D, and for polynomials for the form $y - \sum_{i=0}^{\Delta} x^i \le 0$, there exists an $\overset{o}{\Omega}(n^{\Delta+1})$ space lower bound for data structures with query time $\overset{o}{O}(1)$. However, this lower bound does not go far enough, even in 2D, where a semialgebraic range can be specified by bivariate monic polynomial inequalities[2] of form $\sum_{i,j:i+j\le\Delta} a_{ij}x^iy^j \le 0$ with $a_{\Delta 0} = -1$. In this case, β can be as large as $\boldsymbol{m}_{2,\Delta} = \binom{\Delta+2}{2} - 1 = \Theta(\Delta^2)$, and much larger than $\Delta + 1$ even for moderate Δ (e.g., for $\Delta = 4$, "5" versus "14", for $\Delta = 5$, "6" versus "20" and so on). Another main weakness is that their lower bound is only in 2D, but the upper bound [5] works in arbitrary dimensions.

The correct upper bound tradeoff seems to be even more mysterious. Typically, the tradeoff is obtained by combining the linear space and the polylogarithmic query time solutions. For simplex range searching (i.e., when $\Delta = 1$), the tradeoff is $S(n) = \tilde{O}(n^D/Q(n)^D)$ [16], which is a natural looking bound and it is also known to be optimal. The tradeoff bound becomes very mysterious for semialgebraic range searching. For example, for $D = 2$ and when restricted to queries of the form $y - \sum_{i=0}^{\Delta} x^i \le 0$, combining the existing solutions yields the bound $S(n) = \tilde{O}(n^{\Delta+1}/Q(n)^{2\Delta})$ whereas the known lower bound [3] is $S(n) = \overset{o}{\Omega}(n^{\Delta+1}/Q(n)^{(\Delta+3)\Delta/2})$. One possible reason for this gap is that the lower bound construction is based on a uniform random point set, while in practice, the input can be pathological. But in general the uniform random point set assumption is not too restrictive for range searching problems. Almost all known lower bounds rely on this assumption: e.g., half-space range searching [9, 7, 8], orthogonal range searching [11, 12, 2], simplex range searching [10, 13, 1].

1.2 Our Results

Our results consist of two parts. First, we study a problem that we call "the general polynomial slab range reporting". Formally, let $P(X)$ be a monic D-variate polynomial of degree at most Δ, a general polynomial slab is defined to be the region between $P(X) = 0$ and $P(X) = w$ for some parameter w specified at the query time. Unlike [3], our construction can reach the maximum possible parameter number $\boldsymbol{m}_{D,\Delta}$. For simplicity, we use $\boldsymbol{m}$ instead of $\boldsymbol{m}_{D,\Delta}$ when the context is clear. We give a space-time tradeoff lower bound of $S(n) = \overset{o}{\Omega}(n^{\boldsymbol{m}}/Q(n)^{\Theta((\Delta^2+D\Delta)\boldsymbol{m})})$, which is (almost) tight when $Q(n) = n^{o(1)}$.

For the second part, we present data structures that match the lower bounds studied in the work by Afshani and Cheng [3]. We show that their lower bounds for 2D polynomial slabs and 2D annuli are tight for uniform random point sets. Our bound shows that current tradeoff given by the classical method of combining extreme solutions [18, 5] might not be tight. We shred some lights on the upper bound tradeoff and develop some ideas which could be used to tackle the problem. Our results are summarized in Table 1.

1.3 Technical Contributions

Compared to the previous lower bound in [3], we need to wrestle with many complications that stem from the algebraic geometry nature of the problem. In Section 3, we cover them in greater detail, but briefly speaking, the technical heart of the results in [3] is that "two univariate polynomials $P_1(x)$ and $P_2(x)$ that have sufficiently different leading coefficients, cannot pass close to each other for too long. However, this claim is not true for even bivariate polynomials, since $P_1(x, y)$ and $P_2(x, y)$ could have infinitely many roots in common and thus we can have $P_1(x, y) - P_2(x, y) = 0$ in an unbounded region of $\mathbb{R}^2$. Overcoming this requires significant innovations.

---

[2] We define that a D-variate polynomial $P(X_1, X_2, \cdots, X_D)$ is monic if the coefficient of X_2^{Δ} is -1.

Table 1 Our Results (marked by *). Our upper bounds are for uniform random point sets.

Query Types	Lower Bound	Upper Bound
General Polynomial Slabs $\left(m = m_{D,\Delta} = \binom{D+\Delta}{D} - 1\right)$	$S(n) = \overset{o}{\Omega}\left(\frac{n^m}{Q(n)^{\Theta(m)}}\right)^*$	$S(n) = \tilde{O}\left(\frac{n^m}{Q(n)^{\Theta(m)}}\right)$ [18, 5]
When $Q(n) = \overset{o}{O}(1)$	$\boldsymbol{S(n) = \overset{o}{\Omega}\left(n^m\right)^*}$	$\boldsymbol{S(n) = \tilde{O}\left(n^m\right)}$ [18, 5]
2D Semialgebraic Sets $\left(m = m_{2,\Delta} = \binom{2+\Delta}{2} - 1\right)$	$S(n) = \overset{o}{\Omega}\left(\frac{n^m}{Q(n)^{m+m^2(m-1)-1}}\right)^*$	$S(n) = \tilde{O}\left(\frac{n^m}{Q(n)^{2m-2}}\right)$ [18, 5] $\boldsymbol{S(n) = \tilde{O}\left(\frac{n^m}{Q(n)^{3m-4}}\right)^*}$
2D Polynomial Slabs	$\boldsymbol{S(n) = \overset{o}{\Omega}\left(\frac{n^{\Delta+1}}{Q(n)^{(\Delta+3)\Delta/2}}\right)}$ [3]	$S(n) = \tilde{O}\left(\frac{n^{\Delta+1}}{Q(n)^{2\Delta}}\right)$ [18, 5] $\boldsymbol{S(n) = \tilde{O}\left(\frac{n^{\Delta+1}}{Q(n)^{(\Delta+3)\Delta/2}}\right)^*}$
2D Annuli	$\boldsymbol{S(n) = \overset{o}{\Omega}\left(\frac{n^3}{Q(n)^5}\right)}$ [3]	$S(n) = \tilde{O}\left(\frac{n^3}{Q(n)^4}\right)$ [18, 5] $\boldsymbol{S(n) = \tilde{O}\left(\frac{n^3}{Q(n)^5}\right)^*}$

2 Preliminaries

In this section, we introduce some tools we will use in this paper. We will mainly use the lower bound tools used in [3]. For more detailed introduction, we refer the readers to [3].

2.1 A Geometric Lower Bound Framework

We present a lower bound framework in the pointer machine model of computation. It is a streamlined version of the framework by Chazelle [11] and Chazelle and Rosenberg [13]. In essence, this is an encapsulation of the way the framework is used in [3].

In a nutshell, in the pointer machine model, the memory is represented as a directed graph where each node can store one point and it has two pointers to two other nodes. Given a query, starting from a special "root" node, the algorithm explores a subgraph that contains all the input points to report. The size of the explored subgraph is the query time.

Intuitively, for range reporting, to answer a query fast, we need to store its output points close to each other. If each query range contains many points to report and two ranges share very few points, some points must be stored multiple times, thus the total space usage must be big. We present the framework, and refer the readers to the full version of the paper for the proof.

▶ **Theorem 1.** *Suppose the D-dimensional geometric range reporting problems admit an $S(n)$ space and $Q(n)+O(k)$ query time data structure, where n is the input size and k is the output size. Let $\mu^D(\cdot)$ denote the D-dimensional Lebesgue measure. (We call this D-measure for short.) Assume we can find $m = n^c$ ranges $\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_m$ in a D-dimensional cube $\boldsymbol{C}^D$ of side length $|l|$ for some constant c such that (i) $\forall i = 1, 2, \cdots, m, \mu^D(\mathcal{R}_i \cap \boldsymbol{C}^D) \geq 4c|l|^D Q(n)/n$; and (ii) $\mu^D(\mathcal{R}_i \cap \mathcal{R}_j) = O(|l|^D/(n2^{\sqrt{\log n}}))$ for all $i \neq j$. Then, we have $S(n) = \overset{o}{\Omega}(mQ(n))$.*

2.2 A Lemma for Polynomials

Given a univariate polynomial and some positive value w, the following lemma from [3] upper bounds the length of the interval within which the absolute value of the polynomial is no more than w. We will use this lemma as a building block for some of our proofs.

▶ **Lemma 2** (Afshani and Cheng [3]). *Given a degree-Δ univariate polynomial $P(x) = \sum_{i=0}^{\Delta} a_i x^i$ where $|a_\Delta| > 0$ and $\Delta > 0$. Let w be any positive value. If $|P(x)| \le w$ for all $x \in [x_0, x_0 + t]$ for some parameter x_0, then $t = O((w/|a_\Delta|)^{1/\Delta})$.*

2.3 Useful Properties about Matrices

In this section, we recall some useful properties about matrices. We first recall some properties of the determinant of matrices. One important property is that the determinant is mutilinear:

▶ **Lemma 3.** *Let $A = \begin{bmatrix}\mathbf{a}_1 & \cdots & \mathbf{a}_n\end{bmatrix}$ be a $n \times n$ matrix where $\mathbf{a}_i$'s are vectors in $\mathbb{R}^n$. Suppose $\mathbf{a}_j = r \cdot \mathbf{w} + \mathbf{v}$ for some $r \in \mathbb{R}$ and $\mathbf{w}, \mathbf{v} \in \mathbb{R}^n$, then the determinant of A, denoted $\det(A)$, is*

$$\begin{aligned}\det(A) &= \det\left(\begin{bmatrix}\mathbf{a}_1 & \cdots & \mathbf{a}_{j-1} & \mathbf{a}_j & \mathbf{a}_{j+1} & \cdots & \mathbf{a}_n\end{bmatrix}\right)\\ &= r \cdot \det\left(\begin{bmatrix}\mathbf{a}_1 & \cdots & \mathbf{a}_{j-1} & \mathbf{w} & \mathbf{a}_{j+1} & \cdots & \mathbf{a}_n\end{bmatrix}\right)\\ &\quad + \det\left(\begin{bmatrix}\mathbf{a}_1 & \cdots & \mathbf{a}_{j-1} & \mathbf{v} & \mathbf{a}_{j+1} & \cdots & \mathbf{a}_n\end{bmatrix}\right).\end{aligned}$$

One of the special types of matrices we will use is the Vandermonde matrix which is a square matrix where the terms in each row form a geometric series, i.e., $V_{ij} = x_i^{j-1}$ for all indices i and j. The determinant of such a matrix is $\det(V) = \prod_{1 \le i < j \le n}(x_j - x_i)$.

Given an n-tuple $\lambda = (\lambda_1, \lambda_2, \cdots, \lambda_n)$ where $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n \ge 0$, we can define a generalized Vandermonde matrix V^* defined by λ, where $V^*_{ij} = x_i^{\lambda_{n-j+1}+j-1}$. The determinant of V^* is known to be the product of the determinant of the induced Vandermonde matrix V_{V^*} with $V_{ij} = x_i^{j-1}$ and the Schur polynomial $s_\lambda(x_1, x_2, \cdots, x_n) = \sum_T x_1^{t_1} \cdots x_n^{t_n}$, where the summation is over all semistandard Young tableaux [20] T of shape λ. The exponents $t_1, t_2, \cdots, t_n$ are all nonnegative numbers. The following lemma bounds the determinant of a generalized Vandermonde matrix.

▶ **Lemma 4.** *Let V^* be a generalized Vandermonde matrix defined by $\lambda = (\lambda_1, \lambda_2, \cdots, \lambda_n)$ where $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n \ge 0$. If $n, \lambda_1 = \Theta(1)$, and for all i, $x_i = \Theta(1)$, then $\det(V^*) = \Theta(\det(V_{V^*}))$, where V_{V^*} is the induced Vandermonde matrix with $V_{ij} = x_i^{j-1}$.*

3 Lower Bound for Range Reporting with General Polynomial Slabs

In this section, we prove our main lower bound for general polynomial slabs.

▶ **Definition 5.** *A general polynomial slab in $\mathbb{R}^D$ is a triple (P, a, b) where $P \in \mathbb{R}[X]$ is a degree-Δ D-variate polynomial and a, b are two real numbers such that $a < b$. A general polynomial slab is defined as $\{X \in \mathbb{R}^D : a \le P(X) \le b\}$. Note that due to rescaling, we can assume that the polynomial is monic.*

Before presenting our results, we first describe the technical challenges of this problem. We explain why the construction used in [3] cannot be generalized in an obvious way and give some intuition behind our lower bound construction.

3.1 Technical Challenges

Our goal is a lower bound of the form $\overset{o}{\Omega}(n^{\boldsymbol{m}}/Q(n)^{\Theta(\boldsymbol{m})})$. To illustrate the challenges, consider the case $D = 2$ and the unit square $\boldsymbol{U} = \boldsymbol{U}^2 = [0,1] \times [0,1]$. To use Theorem 1, we need to generate about $\overset{o}{\Omega}(n^{\boldsymbol{m}})$ polynomial slabs such that each slab should have width approximately $\Omega(Q(n)/n)$, and any two slabs should intersect with area approximately $O(1/n)$. Intuitively, this means two slabs cannot intersect over an interval of length $\Omega(1/Q(n))$.

In Lemma 2, for univariate polynomials, the observation behind their construction is that when the leading coefficients of two polynomials differ by a large number, the length of the interval in which two polynomials are close to each other is small. However, when we consider general bivariate polynomials in $\mathbb{R}^2$, this observation is no longer true. For example, consider $P_1(x,y) = (x+1)(1000x^2+y)$ and $P_2(x,y) = (x+1)(x^2+1000y)$. The leading coefficients are 1000 and 1 respectively, but since P_1, P_2 have a common factor $(x+1)$, their zero sets have a common line. Thus any slab of width $Q(n)/n$ generated for these two polynomial will have infinite intersection area, which is too large to be useful.

At first glance, it might seem that this problem can be fixed by picking the polynomials randomly, e.g., each coefficient is picked independently and uniformly from the interval $[0,1]$, as a random polynomial in two or more variables is irreducible with probability 1. Unfortunately, this does not work either but for some very nontrivial reasons. To see this, consider picking coefficients uniformly at random from range $[0,1]$ for bivariate polynomials $P(x,y) = \sum_{i+j\le\Delta} a_{ij}x^iy^j$. The probability of pick a polynomial with $0 \le a_{0j} \le \frac{1}{n}$ for all a_{0j} is $\frac{1}{n^{\Delta+1}}$. For such polynomials, $0 \le P(0,y) \le \frac{\Delta+1}{n}$ for $y \in [0,1]$. Suppose we sampled two such polynomials, then the two slabs generated using them will contain $x=0$ for $y \in [0,1]$, meaning, the two slabs will have too large of an area $(\Omega(Q(n)/n))$ in common, so we cannot have that. Unfortunately, if we sample more than $n^{\Delta+1}$ polynomials, this will happen with probability close to one, and there seems to be no easy fix. A deeper insight into the issue is given below.

Map a polynomial $\sum_{i+j\le\Delta} a_{ij}x^iy^j$ to the point $(a_{00}, a_{01}, \cdots, a_{\Delta 0})$ in $\mathbb{R}^{\boldsymbol{m}}$. The above randomized construction corresponds to picking a random point from the unit cube $\boldsymbol{U}$ in $\mathbb{R}^{\boldsymbol{m}}$. Now consider the subset Γ of $\mathbb{R}^{\boldsymbol{m}}$ that corresponds to reducible polynomials. The issue is that Γ intersects $\boldsymbol{U}$ and thus we will sample polynomials that are close to reducible polynomials, e.g., a sampled polynomial with $a_{0j} = 0 \in [0, \frac{1}{n}]$ is close to the reducible polynomial with $a_{0j} = 0$. Pick a large enough sample and two points will lie close to the same reducible polynomial and thus they will produce a "large" overlap in the construction. Our main insight is that there exists a point $\mathbf{p}$ in $\boldsymbol{U}$ that has a "fixed" (i.e., constant) distance to Γ; thus, we can consider a neighborhood around $\mathbf{p}$ and sample our polynomials from there. However, more technical challenges need to be overcome to even make this idea work but it turns out, we can simply pick our polynomials from a grid constructed in the small enough neighborhood of some such point $\mathbf{p}$ in $\mathbb{R}^{\boldsymbol{m}}$.

3.2 A Geometric Lemma

In this section, we show a geometric lemma which we will use to establish our lower bound. In a nutshell, given two monic D-variate polynomials P_1, P_2 and a point $p = (p_2, p_3, \cdots, p_D) \in \mathbb{R}^{D-1}$ in the $(D-1)$-dimensional subspace perpendicular to the X_1-axis, we define the distance between $Z(P_1)$[3] and $Z(P_2)$ along the X_1-axis at point p to be $|a-b|$, where $(a, p_2, \cdots, p_D) \in Z(P_1)$ and $(b, p_2, \cdots, p_D) \in Z(P_2)$. In general, this distance is not well-defined as there could be multiple a and b's satisfying the definition. But we can show that for a specific set of polynomials, a, b can be made unique and thus the distance is well-defined. For P_1, P_2 with "sufficiently different" coefficients, we present a lemma which upper bounds the $(D-1)$-measure of the set of points p at which the distance between $Z(P_1)$ and $Z(P_2)$ is "small". Intuitively, this can be viewed as a generalization of Lemma 2. We first prove the lemma in 2D for bivariate polynomials, and then extend the result to higher dimensions.

[3] $Z(P)$ denotes the zero set of polynomial P.

First, we define the notations we will use for general D-variate polynomials.

▶ **Definition 6.** *Let $I^D \subseteq \{(i_1, i_2, \cdots, i_D) \in \mathbb{N}^D\}$[4], $D \geq 1$, be a set of D-tuples where each tuple consists of nonnegative integers. We call I^D an index set (of dimension D). Let $X^D = (X_1, X_2, \cdots, X_D)$ be a D-tuple of indeterminates. When the context is clear, we use X for simplicity. Given an index set I^D, we define*

$$P(X) = \sum_{i \in I^D} A_i X^i,$$

where $A_i \in \mathbb{R}$ is the coefficient of X^i and $X^i = X_1^{i_1} X_2^{i_2} \cdots X_D^{i_D}$, to be a D-variate polynomial. For any $i \in I^D$, we define $\sigma(i) = \sum_{j=1}^{D} i_j$. Let Δ be the maximum $\sigma(i)$ with $A_i \neq 0$, and we say P is a degree-Δ polynomial. Given a D-tuple T, we use $T_{:j}$ to denote a j-tuple by taking only the first j components of T. Also, we use notation T_j to specify the j-th component of T. Conversely, given a $(D-1)$-tuple t and a value v, we define $t \oplus v$ to be the D-tuple formed by appending v to the end of t.

We will consider polynomials of form

$$P(X) = X_1 - X_2^{\Delta} + \sum_{i \in I^D} A_i X^i,$$

where $0 \leq A_{ij} = O(\epsilon) = o(1)$ for all $\sigma(i) \leq \Delta$ except that $A_i = 0$ for $i = (0, \Delta, 0, \cdots, 0)$. Intuitively, these are monic polynomials packed closely in the neighborhood of $P(X) = X_1 - X_2^D$. For simplicity, we call them "packed" polynomials. We will prove a property for packed polynomials that are "sufficiently distant". More precisely,

▶ **Definition 7.** *Given two distinct packed degree-Δ D-variate polynomials P_1, P_2, we say P_1, P_2 are "distant" if each coefficient of $P_1 - P_2$ has absolute value at least $\xi_D = \delta\tau^{\mathcal{B}}(\eta\tau)^{(D-2)\Delta} > 0$ if not zero for parameters $\delta, \eta, \tau > 0$ and $\eta\tau = O((1/\epsilon)^{1/\mathcal{B}})$, where $\mathcal{B} = \binom{\boldsymbol{b}}{2}$ and $\boldsymbol{b} = \boldsymbol{m}_{2,\Delta}$ is the maximum number of coefficients needed to define a monic degree-Δ bivariate polynomial.*

We will use the following simple geometric observation. See the full version of the paper for the proof.

▶ **Observation 8.** *Let P be a packed D-variate polynomial and $a = (a_1, a_2, \cdots, a_D) \in Z(P)$. If $a_i \in [1, 2]$ for all $i = 2, 3, \cdots, D$, then there exists a unique a_1 such that $0 < a_1 = O(1)$.*

With this observation, we can define the distance between the zero sets of two polynomials along the X_1-axis at a point in $[1, 2]^{D-1}$ of the subspace perpendicular to the X_1 axis.

▶ **Definition 9.** *Given two packed polynomials P_1, P_2 and a point $p = (p_2, p_3, \cdots, p_D) \in [1, 2]^{D-1}$, we define the distance between $Z(P_1)$ and $Z(P_2)$ at p, denoted $\pi(Z(P_1), Z(P_2), p)$, to be $|a - b|$ s.t. $a, b > 0$, and $(a, p_2, p_3, \cdots, P_D) \in Z(P_1)$ and $(b, p_2, p_3, \cdots, P_D) \in Z(P_2)$.*

Now we show a generalization of Lemma 2 to distant bivariate polynomials in 2D.

▶ **Lemma 10.** *Let P_1, P_2 be two distinct distant bivariate polynomials. Let $I = \{y : \pi(Z(P_1), Z(P_2), y) = O(w) \wedge y \in [1, 2]\}$, where $w = \delta/\eta^{\mathcal{B}} = o(1)$. Then $|I| = O(\frac{1}{\eta\tau})$.*

---

[4] In this paper, $\mathbb{N} = \{0, 1, 2, \cdots\}$.

Proof. We prove it by contradiction. The idea is that if the claim does not hold, then we can "tweak" the coefficients of P_2 by a small amount such that the tweaked polynomial and P_1 have $\boldsymbol{b}$ common roots. Next, we show this implies that the tweaked polynomial is equivalent to P_1. Finally we reach a contradiction by noting that by assumption at least one of the coefficients of P_1 and P_2 is not close. Let $P_1(x,y) = x - y^\Delta + \sum_{i=0}^{\Delta}\sum_{j=0}^{\Delta-i} a_{ij}x^iy^j$ and $P_2(x,y) = x - y^\Delta + \sum_{i=0}^{\Delta}\sum_{j=0}^{\Delta-i} b_{ij}x^iy^j$ where by definition all a_{ij}'s and b_{ij}'s are $O(\epsilon)$. Suppose for the sake of contradiction that $|I| = \omega(\frac{1}{\eta\tau})$. We pick $\boldsymbol{b}$ values $y_1, y_2, \cdots, y_{\boldsymbol{b}}$ in I s.t. $|y_i - y_j| \geq |I|/\boldsymbol{b}$ for all $i \neq j$. Let $x_1, x_2, \cdots, x_{\boldsymbol{b}}$ be the corresponding values s.t. $(x_k, y_k) \in Z(P_1)$ in the first quadrant, i.e., $P_1(x_k, y_k) = 0$ for $k = 1, 2, \cdots, \boldsymbol{b}$. Note that

$$P_1(x_k, y_k) = 0 \equiv x_k - y_k^\Delta + \sum_{i=0}^{\Delta}\sum_{j=0}^{\Delta-i} a_{ij}x_k^iy_k^j = 0 \implies x_k = y_k^\Delta - O(\epsilon),$$

since $a_{ij} = O(\epsilon)$ and $x_k, y_k = O(1)$ by Observation 8. Since $\pi(Z(P_1), Z(P_2), y_k) = O(w)$ for all $y_k \in I$, let $(x_k + \Delta x_k, y_k)$ be the points on $Z(P_2)$, we have $P_2(x_k + \Delta x_k, y_k) = P_2(x_k, y_k) + \Theta(\Delta x_k) = 0$. Since $|\Delta x_k| = O(w)$, $P_2(x_k, y_k) = \boldsymbol{\gamma}_k$ for some $|\boldsymbol{\gamma}_k| = O(w)$. We would like to show that we can "tweak" every coefficient b_{ij} of $P_2(x,y)$ by some value $\mathbf{d}_{ij}$, to turn P_2 into a polynomial Q s.t. $Q(x_k, y_k) = 0, \forall k = 1, 2, \cdots, \boldsymbol{b}$. If so, for every pair (x_k, y_k),

$$\begin{aligned} Q(x_k, y_k) &= x_k - y_k^\Delta + \sum_{i=0}^{\Delta}\sum_{j=0}^{\Delta-i}(b_{ij} + \mathbf{d}_{ij})x_k^iy_k^j \\ &= P_2(x_k, y_k) + \sum_{i=0}^{\Delta}\sum_{j=0}^{\Delta-i}\mathbf{d}_{ij}x_k^iy_k^j \\ &= \boldsymbol{\gamma}_k + \sum_{i=0}^{\Delta}\sum_{j=0}^{\Delta-i}\mathbf{d}_{ij}(y_k^\Delta - O(\epsilon))^iy_k^j \\ &= \boldsymbol{\gamma}_k + \sum_{i=0}^{\Delta}\sum_{j=0}^{\Delta-i}\mathbf{d}_{ij}(y_k^{i\Delta} - O(\epsilon))y_k^j, \end{aligned}$$

where the last equality follows from $\epsilon = o(1)$ and $1 \leq y_k \leq 2$. So to find $\mathbf{d}_{ij}$'s and to be able to tweak $P_2(x,y)$, we need to solve the following linear system

$$\begin{bmatrix} 1 & y_1 & y_1^2 & \cdots & y_1^{\Delta-1} & y_1^\Delta - O(\epsilon) & \cdots & y_1^{\Delta^2} - O(\epsilon) \\ 1 & y_2 & y_2^2 & \cdots & y_2^{\Delta-1} & y_2^\Delta - O(\epsilon) & \cdots & y_2^{\Delta^2} - O(\epsilon) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & y_{\boldsymbol{b}} & y_{\boldsymbol{b}}^2 & \cdots & y_{\boldsymbol{b}}^{\Delta-1} & y_{\boldsymbol{b}}^\Delta - O(\epsilon) & \cdots & y_{\boldsymbol{b}}^{\Delta^2} - O(\epsilon) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{d}_{00} \\ \mathbf{d}_{01} \\ \vdots \\ \mathbf{d}_{\Delta 0} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\gamma}_1 \\ -\boldsymbol{\gamma}_2 \\ \vdots \\ -\boldsymbol{\gamma}_{\boldsymbol{b}} \end{bmatrix},$$

where the exponents of y_k are generated by $i\Delta + j$ for $i, j \in \{0, 1, 2, \cdots, \Delta\}$, $j \neq \Delta$, and $i + j \leq \Delta$. Let us call the above linear system $A \cdot \mathbf{d} = \boldsymbol{\gamma}$.

By Lemma 3, $\det(A) = \det(A^*) + \sum_{l=1}^{\Theta(1)} \det(A_l)$, where A^* is a generalized Vandermonde matrix defined by an $\boldsymbol{b}$-tuple $\lambda = (\Delta^2 - \boldsymbol{b}, \ldots, 0)$, and each A_l is a matrix with some columns being $O(\epsilon)$. Since $\boldsymbol{b} = \binom{2+\Delta}{2} - 1$ is $\Theta(1)$, by Lemma 4, we can bound $\det(A^*)$ by $\Theta(\det(V_{A^*}))$, where V_{A^*} is the induced Vandermonde matrix. Since $|y_i - y_j| = \Omega(|I|)$ for $i \neq j$, $\det(V_{A^*}) = \prod_{1 \leq i < j \leq \boldsymbol{b}}(y_j - y_i)) = \Omega(|I|^{\mathcal{B}})$. On the other hand, for every matrix A_l, there is at least one column where the magnitude of all the entries is $O(\epsilon)$. Since all other entries are bounded by $O(1)$, by the Leibniz formula for determinants, $|\det(A_l)| = O(\epsilon) = O((\frac{1}{\eta\tau})^{\mathcal{B}})$. Since $|I|^{\mathcal{B}} = \omega((\frac{1}{\eta\tau})^{\mathcal{B}})$, we can bound $|\det(A)| = \Omega(|I|^{\mathcal{B}})$ and in particular $|\det(A)| \neq 0$

and thus the above system has a solution and the polynomial Q exists. Furthermore, we can compute $\mathbf{d} = A^{-1}\boldsymbol{\gamma} = \frac{1}{\det(A)} C \cdot \boldsymbol{\gamma}$, where C is the cofactor matrix of A. Since all entries of A are bounded by $O(1)$, then the entries of C, being cofactors of A, are also bounded by $O(1)$. Since $|\boldsymbol{\gamma}_k| = O(w)$ and $|I| = \omega(\frac{1}{\eta\tau})$, for every $k = 1, 2, \cdots, \boldsymbol{b}$, we have $|\mathbf{d}_{ij}| = O(w/|I|^{\mathcal{B}}) = o(w(\eta\tau)^{\mathcal{B}}) = o(\delta\tau^{\mathcal{B}})$.

However, since both $Z(P_1)$ and $Z(Q)$ pass through these $\boldsymbol{b}$ points, both P_1 and Q should satisfy $A \cdot \mathbf{c}_1 = 0$ and $A \cdot \mathbf{c}_2 = 0$, where $\mathbf{c}_1, \mathbf{c}_2$ are their coefficient vectors respectively. But since $\det(A) \neq 0$, $\mathbf{c}_1 = \mathbf{c}_2$, meaning, $P_1 \equiv Q$. This means for every $i, j = 0, 1, \cdots, \Delta$, where $j \neq \Delta$ and $i + j \leq \Delta$, $|a_{ij} - b_{ij}| = \mathbf{d}_{ij} = o(\delta\tau^{\mathcal{B}})$. However, by assumption, if two polynomials are not equal, then there exists at least one c_{ij} such that they differ by at least $\delta\tau^{\mathcal{B}}$, a contradiction. So $|I| = O(\frac{1}{\eta\tau})$. ◀

We now generalize Lemma 10 to higher dimensions.

▶ **Lemma 11.** *Let P_1, P_2 be two distinct distant D-variate polynomials. Let $S = \{X : \pi(Z(P_1), Z(P_2), X) = O(w) \wedge X \in [1,2]^{D-1}\}$, where $w = \delta/\eta^{\mathcal{B}} = o(1)$. Then $\mu^{D-1}(S) = O(\frac{1}{\eta\tau})$.*

Proof. We prove the lemma by induction. The base case when $D = 2$ is Lemma 10. Now suppose the lemma holds for dimension $D-1$, we prove it for dimension D. Observe that we can rewrite a D-variate polynomial $P(X) = X_1 - X_2^{\Delta} + \sum_{i \in I^D} A_i X^i$ as $P(X) = X_1 - X_2^{\Delta} + \sum_{j \in I^D_{:D-1}} (f_j(X_D)) X^j_{:D-1}$, where $f_j(X_D) = \sum_{k=0}^{\Delta - \sigma(j)} A_{j \oplus k} X_D^k$. Consider two distinct distant D-variate polynomials $P(X) = X_1 - X_2^{\Delta} + \sum_{i \in I^D} A_i X^i$ and $Q(X) = X_1 - X_2^{\Delta} + \sum_{i \in I^D} B_i X^i$. Let f_j, g_j be the corresponding coefficients for $X^j_{:D-1}$. Note that there exists some j such that $f_j \not\equiv g_j$ because P_1, P_2 are distinct. Let $h_j(X_D) = f_j(X_D) - g_j(X_D)$ and observe that h_j is a univariate polynomial in X_D. We show that the interval length of X_D in which $|h_j(X_D)| < \xi_{D-1}$ is upper bounded by $O(\frac{1}{\eta\tau})$ for any $h_j(X_D) \not\equiv 0$. Pick any $h_j(X_D) \not\equiv 0$ and note that this means there exists at least one coefficient of $h_j(X_D)$ that is nonzero. By assumption, each coefficient of $h_j(X_D)$ has absolute value at least ξ_D if not zero. If the constant term is the only nonzero term, then the interval length of X_D in which $|h_j(X_D)| < \xi_{D-1}$ is 0, since $|h_j(X_D)| \geq \xi_D > \xi_{D-1}$ by definition. Otherwise by Lemma 2, the interval length $|r|$ for X_D in which $|h_j(X_D)| < \xi_{D-1}$ is upper bounded by

$$|r| = O\left(\left(\frac{\xi_{D-1}}{\xi_D}\right)^{1/\Delta}\right) = O\left(\left(\frac{1}{(\eta\tau)^{\Delta}}\right)^{1/\Delta}\right) = O\left(\frac{1}{\eta\tau}\right).$$

Since the total number of different j's is $\Theta(1)$, the total number of $h_j(X_D)$ is then $\Theta(1)$. So the total interval length for X_D within which there is some nonzero $h_j(X_D)$ with $|h_j(X_D)| < \delta\tau_{D-1}$ is upper bounded by $\Theta(1) \cdot O(\frac{1}{\eta\tau}) = O(\frac{1}{\eta\tau})$. Since we are in a unit hypercube, we can simply upper bound $\mu^{D-1}(S)$ by $O(\frac{1}{\eta\tau}) \cdot \Theta(1) = O(\frac{1}{\eta\tau})$. Otherwise, by the inductive hypothesis, the $(D-2)$-measure of S in $[1,2]^{D-2}$ is upper bounded by $O(\frac{1}{\eta\tau})$. Integrating over all X_D, $\mu^{D-1}(S)$ is bounded by $O(\frac{1}{\eta\tau})$ in this case as well. ◀

3.3 Lower Bound for General Polynomial Slabs

Now we are ready to present our lower bound construction. We will use a set $\mathcal{S}$ of D-variate polynomials in $\mathbb{R}[X]$ of form:

$$P(X) = X_1 - X_2^{\Delta} + \sum_{i \in I^D} A_i X^i,$$

where X is a D-tuple of indeterminates, I^D is an index set containing all D-tuples i satisfying $\sigma(i) \leq \Delta$, and each $A_i \in \{k\xi_D : k = \lfloor \frac{\epsilon}{2\xi_D} \rfloor, \lfloor \frac{\epsilon}{2\xi_D} \rfloor + 1, \cdots, \lfloor \frac{\epsilon}{\xi_D} \rfloor\}$ for some $\xi_D = \delta\tau^{\mathcal{B}}(\eta\tau)^{(D-2)\Delta}$ to be set later, except for one special coefficient: we set $A_i = 0$ for $i = (0, \Delta, 0, \cdots, 0)$. Note that every pair of the polynomials in $\mathcal{S}$ is distant. A general polynomial slab is defined to be a triple $(P, 0, w)$ where $P \in \mathcal{S}$ and w is a parameter to be set later. We need $w = o(\epsilon)$ and $\epsilon = o(1)$.

We consider a unit cube $\boldsymbol{U}^D = \prod_{i=1}^{D}[1, 2] \subseteq \mathbb{R}^D$ and use Framework 1. Recall that to use Framework 1, we need to lower bound the intersection D-measure of each slab we generated and $\boldsymbol{U}^D$, and upper bound the intersection D-measure of two slabs.

Given a slab $(P, 0, w)$ in our construction, first note that both P and $P - w$ are packed polynomials. We define the width of $(P, 0, w)$ to be the distance between $Z(P)$ and $Z(P - w)$ along the X_1-axis. The following lemma shows that the width of each slab we generate will be $\Theta(w)$ in $\boldsymbol{U}^D$. See the full version of the paper for the proof.

▶ **Lemma 12.** *Let $P_1 \in \mathcal{S}$ and $P_2 = P_1 - r$ for any $0 \leq r = O(w)$. Then $\pi(Z(P_1), Z(P_2), X) = \Theta(r)$ for any $X \in [1, 2]^{D-1}$.*

The following simple lemma bounds the $(D - 1)$-measure of the projection of the intersection of the zero set of any polynomial in our construction and $\boldsymbol{U}^D$ on the $(D - 1)$-dimensional subspace perpendicular to X_1-axis. See the full version of the paper for the proof.

▶ **Lemma 13.** *Let $P \in \mathcal{S}$. The projection of $Z(P) \cap \boldsymbol{U}^D$ on the $(D - 1)$-dimensional space perpendicular to the X_1-axis has $(D - 1)$-measure $\Theta(1)$.*

Combining Lemma 12 and Lemma 13, we easily bound the intersection D-measure of any slab in our construction and $\boldsymbol{U}^D$.

▶ **Corollary 14.** *Any slab in our construction intersects $\boldsymbol{U}^D$ with D-measure $\Theta(w)$.*

Combining Lemma 12 and Lemma 11, we easily bound the intersection D-measure of two slabs in our construction in $\boldsymbol{U}^D$.

▶ **Corollary 15.** *Any two slabs in our construction intersect with D-measure $O(\frac{w}{\eta\tau})$ in $\boldsymbol{U}^D$.*

Since there are at most $\boldsymbol{m} = \binom{D+\Delta}{D} - 1$ parameters for a degree-Δ D-variate monic polynomial, the number of polynomial slabs we generated is then

$$\Theta\left(\left(\frac{\epsilon}{\xi_D}\right)^{\boldsymbol{m}}\right) = \Theta\left(\left(\frac{n}{Q(n)^{1+2\mathcal{B}+(D-2)\Delta}2^{((D-2)\Delta+2\mathcal{B})\sqrt{\log n}}}\right)^{\boldsymbol{m}}\right) = O(n^{\boldsymbol{m}}),$$

by setting $\delta = wQ(n)^{\mathcal{B}}$, $\eta = Q(n)$, $\tau = 2^{\sqrt{\log n}}$, $\epsilon = \frac{1}{Q(n)^{\mathcal{B}}2^{\mathcal{B}\sqrt{\log n}}}$, and $w = c_w Q(n)/n$ for a sufficiently large constant c_w. We pick c_w s.t. each slab intersects $\boldsymbol{U}^D$ with D-measure, by Corollary 14, $\Omega(w) \geq 4\boldsymbol{m}Q(n)/n$. By Corollary 15 the D-measure of the intersection of two slabs is upper bounded by $O(\frac{w}{Q(n)2^{\sqrt{\log n}}}) = O(\frac{1}{n2^{\sqrt{\log n}}})$. By Theorem 1, we get the lower bound $S(n) = \mathring{\Omega}\left(n^{\boldsymbol{m}}/Q(n)^{\boldsymbol{m}+2\boldsymbol{m}\mathcal{B}+\boldsymbol{m}(D-2)\Delta-1}\right)$. Thus we get the following result.

▶ **Theorem 16.** *Let $\mathcal{P}$ be a set of n points in $\mathbb{R}^D$, where $D \geq 2$ is an integer. Let $\mathcal{R}$ be the set of all D-dimensional generalized polynomial slabs $\{(P, 0, w) : \deg(P) = \Delta \geq 2, w > 0\}$ where $P \in \mathbb{R}[X_1, X_2, \cdots, X_D]$ is a monic degree-Δ polynomial. Let $\boldsymbol{b}$ (resp. $\boldsymbol{m}$) be the maximum number of parameters needed to specify a moinc degree-Δ bivariate (resp. D-variate) polynomial. Then any data structure for $\mathcal{P}$ that can answer generalized polynomial slab reporting queries from $\mathcal{R}$ with query time $Q(n) + O(k)$, where k is the output size, must use $S(n) = \mathring{\Omega}\left(\frac{n^{\boldsymbol{m}}}{Q(n)^{\boldsymbol{m}+2\boldsymbol{m}\mathcal{B}+\boldsymbol{m}(D-2)\Delta-1}}\right)$ space, where and $\mathcal{B} = \binom{\boldsymbol{b}}{2}$.*

4 Data Structures for Uniform Random Point Sets

In this section, we present data structures for an input point set $\mathcal{P}$ uniformly randomly distributed in a unit square $\boldsymbol{U} = [0,1] \times [0,1]$ for semialgebraic range reporting queries in $\mathbb{R}^2$. Our hope is that some of these ideas can be generalized to build more efficient data structures for general point sets. To this end, we show two approaches based on two different assumptions: one assumes the query curve has bounded curvature, and the other assumes bounded derivatives. We show that for any degree-Δ bivariate polynomial inequality, we can build a data structure with space-time tradeoff $S(n) = \tilde{O}(n^{\boldsymbol{m}}/Q(n)^{3\boldsymbol{m}-4})$, which is optimal for $\boldsymbol{m} = 3$ [3]. When the query curve has bounded derivatives for the first Δ orders within $\boldsymbol{U}$, this bound sharpens to $\tilde{O}(n^{\boldsymbol{m}}/Q(n)^{((2\boldsymbol{m}-\Delta)(\Delta+1)-2)/2})$, which matches the lower bound in [3] for polynomial slabs generated by inequalities of form $y - \sum_{i \le \Delta} a_i x^i \ge 0$. Since any polynomial can be factorized into a product of $O(1)$ irreducible polynomials, and we can show that any irreducible polynomial has bounded curvature (See the full version of the paper for details), we can express the original range by a semialgebraic set consisting of $O(1)$ irreducible polynomials. We mention that both data structures can be made multilevel, then by the standard result of multilevel data structures, see e.g., [16] or [4], it suffices for us to focus on one irreducible polynomial inequality. So the curvature-based approach works for all semialgebraic sets. For both approaches, the main ideas are similar: we first partition $\boldsymbol{U}$ into a $Q(n) \times Q(n)$ grid G, and then build a set of slabs in each cell of G to cover the boundary $\partial\mathcal{R}$ of a query range $\mathcal{R}$. The boundaries of each slab consist of the zero sets of lower degree polynomials. We build a data structure to answer degree-Δ polynomial inequality queries inside each slab, then use the boundaries of slabs to express the remaining parts of $\mathcal{R}$. This lowers the degree of query polynomials, and then we can use fast-query data structures to handle the remaining parts. We assume our data structure can perform common algebraic operations in $O(1)$ time, e.g., compute roots, compute derivatives, etc.

4.1 A Curvature-based Approach

The main observation we use is that when the total absolute curvature of $\partial\mathcal{R}$ is small, the curve behaves like a line, and so we can cover it using mostly "thin" slabs, and a few "thick" slabs when the curvature is big. See Figure 1 for an example. We use the curvature as a "budget": thin slabs have few points in them so we can afford to store them in a "fast" data structure and the overhead will be small. Doing the same with the thick slabs will blow up the space too much so instead we store them in "slower" but "smaller" data structures. The crucial observation here is that for any given query, we only need to use a few "thick" slabs so the slower query time will be absorbed in the overall query time.

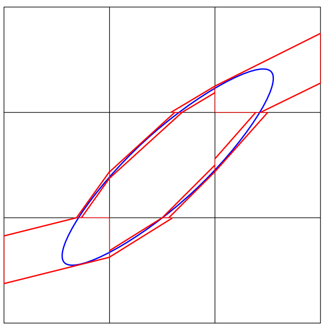

Figure 1 Cover an Ellipse with Slabs of Different Widths.

The high-level idea is to build a two-level data structure. For the bottom-level, we build a multilevel simplex range reporting data structure [16] with query time $\tilde{O}(1) + O(k)$ and space $S(n) = \tilde{O}(n^2)$. For the upper-level, for each cell C in G and a parameter $\alpha = 2^i/Q(n)$, for $i = 0, \cdots, \lfloor \log Q(n) \rfloor$, we generate a series of parallel disjoint slabs of width $\alpha/Q(n)$ such that they together cover C. Then we rotate these slabs by angle $\gamma = j/Q(n)$, for $j = 1, 2, \cdots, \lfloor 2\pi Q(n) \rfloor$. For each slab we generated during this process, we collect all the points in it and build a $\tilde{O}(Q(n)\alpha) + O(k)$ query time and $\tilde{O}((n/(Q(n)\alpha))^{\boldsymbol{m}})$ space data structure by linearization [19] to $\mathbb{R}^{\boldsymbol{m}}$ and using simplex range reporting [16].

The following lemma shows we can efficiently report the points close to $\partial\mathcal{R}$ using slabs we constructed. For the proof of this lemma, we refer the readers to the full version of the paper.

▶ **Lemma 17.** *We can cut $\partial\mathcal{R}$ into a set $\mathcal{S}$ of $O(Q(n))$ sub-curves such that for each sub-curve σ, we can find a set S_σ of slabs that together cover σ. Let P_σ be the subset of the input that lies inside the query and inside the slabs, i.e., $P_\sigma = \mathcal{R} \cap \mathcal{P} \cap (\cup_{s \in S_\sigma} s)$. P_σ can be reported in time $Q(n)\tilde{O}(\kappa_\sigma + 1/Q(n)) + O(|P_\sigma|)$, where κ_σ is the total absolute curvature of σ. Furthermore, for any two distinct $\sigma_1, \sigma_2 \in \mathcal{S}$, $s_1 \cap s_2 = \emptyset$ for all $s_1 \in S_{\sigma_1}, s_2 \in S_{\sigma_2}$.*

With Lemma 17, we can now bound the total query time for points close to $\partial\mathcal{R}$ by $\sum_\sigma Q(n)\tilde{O}(\kappa_\sigma + 1/Q(n)) + O(t_\sigma) = \tilde{O}(Q(n)) + O(t_1)$, where t_1 is the output size. An important observation is that after covering $\partial\mathcal{R}$, we can express the remaining regions by the boundaries of the slabs used and G, which are linear inequalities and so we can use simplex range reporting. Lemma 18 characterizes the remaining regions. See the full version of the paper for the proof.

▶ **Lemma 18.** *There are $O(Q(n))$ remaining regions and each region can be expressed using $O(1)$ linear inequalities. These regions can be found in time $O(Q(n))$.*

With Lemma 18, the query time for the remaining regions is $\tilde{O}(Q(n)) + O(t_2)$, where t_2 is the number of points in the remaining regions. Then the total query time is easily computed to be bounded by $\tilde{O}(Q(n)) + O(k)$, where $k = t_1 + t_2$.

To bound the space usage for the top-level data structure, note that we have $Q(n)^2$ cells, for each α, we generate $\Theta(\frac{1/Q(n)}{\alpha/Q(n)}) = \Theta(1/\alpha)$ slabs for each of the $\Theta(Q(n))$ angles. Since points are distributed uniformly at random, the expected number of points in a slab of width $\alpha/Q(n)$ in a cell C is $O(n \cdot \frac{1}{Q(n)} \cdot \frac{\alpha}{Q(n)})$. So the space usage for the top-level data structure is

$$S(n) = \sum_\alpha Q(n)^2 \cdot \Theta\left(\frac{1}{\alpha}\right) \cdot \Theta(Q(n)) \cdot \tilde{O}\left(\frac{O\left(n \cdot \frac{1}{Q(n)} \cdot \frac{\alpha}{Q(n)}\right)}{Q(n)\alpha}\right)^{\boldsymbol{m}} = \tilde{O}\left(\frac{n^{\boldsymbol{m}}}{Q(n)^{3\boldsymbol{m}-4}}\right).$$

On the other hand, we know that the space usage for the bottom-level data structure is $\tilde{O}(n^2)$. So the total space usage is bounded by $\tilde{O}(\frac{n^{\boldsymbol{m}}}{Q(n)^{3\boldsymbol{m}-4}})$ for $\boldsymbol{m} \geq 3$.

We therefore obtain the following theorem.

▶ **Theorem 19.** *Let $\mathcal{R}$ be the set of semialgebraic ranges formed by degree-Δ bivariate polynomials. Suppose we have a polynomial factorization black box that can factorize polynomials into the product of irreducible polynomials in time $O(1)$, then for any $\log^{O(1)} n \leq Q(n) \leq n^\epsilon$ for some constant ϵ, and a set $\mathcal{P}$ of n points distributed uniformly randomly in $\boldsymbol{U} = [0,1] \times [0,1]$, we can build a data structure of space $\tilde{O}(n^{\boldsymbol{m}}/Q(n)^{3\boldsymbol{m}-4})$ such that for any $\mathcal{R} \in \mathcal{R}$, we can report $\mathcal{R} \cap \mathcal{P}$ in time $\tilde{O}(Q(n)) + O(k)$ in expectation, where $\boldsymbol{m} \geq 3$ is the number of parameters needed to define a degree-Δ bivariate polynomial and k is the output size.*

4.2 A Derivative-based Approach

If we assume that the derivative of $\partial\mathcal{R}$ is $O(1)$, the previous curvature-based approach can be easily adapted to get a derivative-based data structure. See the full version of the paper for details. We can even do better by using slabs whose boundaries are the zero sets of higher degree polynomials instead of linear polynomials. Using Taylor's theorem, we show that we can cover the boundary of the query using "thin" slabs of lower degree polynomials, similar to the approach above. The full details are presented in the full version of the paper.

▶ **Theorem 20.** *Let $\mathcal{R}$ be the set of semialgebraic ranges formed by degree-Δ bivariate polynomials with bounded derivatives up to the Δ-th order. For any $\log^{O(1)} n \le Q(n) \le n^{\epsilon}$ for some constant ϵ, and a set $\mathcal{P}$ of n points distributed uniformly randomly in $\boldsymbol{U} = [0,1]\times[0,1]$, we can build a data structure which uses space $\tilde{O}(n^{\boldsymbol{m}}/Q(n)^{((2\boldsymbol{m}-\Delta)(\Delta+1)-2)/2})$ s.t. for any $\mathcal{R} \in \mathcal{R}$, we can report $\mathcal{P}\cap\mathcal{R}$ in time $\tilde{O}(Q(n)) + O(k)$ in expectation, where $\boldsymbol{m}$ is the number of parameters needed to define a degree-Δ bivariate polynomial and k is the output size.*

▶ Remark 21. We remark that our data structure can also be adapted to support semialgebraic range searching queries in the semigroup model.

5 Conclusion and Open Problems

In this paper, we essentially closed the gap between the lower and upper bounds of general semialgebraic range reporting in the fast-query case at least as far as the exponent of n is concerned. We show that for general polynomial slab queries defined by D-variate polynomials of degree at most Δ in $\mathbb{R}^D$ any data structure with query time $n^{o(1)} + O(k)$ must use at least $S(n) = \overset{o}{\Omega}(n^{\boldsymbol{m}})$ space, where $\boldsymbol{m} = \binom{D+\Delta}{D} - 1$ is the maximum possible parameters needed to define a query. This matches current upper bound (up to an $n^{o(1)}$ factor).

We also studied the space-time tradeoff and showed an upper bound that matches the lower bounds in [3] for uniform random point sets.

The remaining big open problem here is proving a tight bound for the exponent of $Q(n)$ in the space-time tradeoff. There is a large gap between the exponents in our lower bound versus the general upper bound. Our results show that current upper bound might not be tight. On the other hand, our lower bound seems to be suboptimal when the query time is $n^{\Omega(1)} + O(k)$. Both problems seem quite challenging, and probably require new tools.

References

1 Peyman Afshani. Improved pointer machine and I/O lower bounds for simplex range reporting and related problems. In *Proceedings of the Twenty-Eighth Annual Symposium on Computational Geometry*, SoCG '12, pages 339–346, New York, NY, USA, 2012. Association for Computing Machinery. doi:10.1145/2261250.2261301.

2 Peyman Afshani. A new lower bound for semigroup orthogonal range searching. In *35th International Symposium on Computational Geometry*, volume 129 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 3, 14. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2019.

3 Peyman Afshani and Pingan Cheng. Lower Bounds for Semialgebraic Range Searching and Stabbing Problems. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry (SoCG 2021)*, volume 189 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:15, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2021.8.

4 Pankaj K. Agarwal. Simplex range searching and its variants: a review. In *A journey through discrete mathematics*, pages 1–30. Springer, Cham, 2017.

5 Pankaj K. Agarwal, Boris Aronov, Esther Ezra, and Joshua Zahl. Efficient algorithm for generalized polynomial partitioning and its applications. *SIAM J. Comput.*, 50(2):760–787, 2021. doi:10.1137/19M1268550.

6 Pankaj K. Agarwal, Jiří Matoušek, and Micha Sharir. On range searching with semialgebraic sets. II. *SIAM J. Comput.*, 42(6):2039–2062, 2013. doi:10.1137/120890855.

7 Sunil Arya, Theocharis Malamatos, and David M. Mount. On the importance of idempotence. In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 564–573. ACM, New York, 2006. doi:10.1145/1132516.1132598.

8 Sunil Arya, David M. Mount, and Jian Xia. Tight lower bounds for halfspace range searching. *Discrete Comput. Geom.*, 47(4):711–730, 2012. doi:10.1007/s00454-012-9412-x.

9 Hervé Brönnimann, Bernard Chazelle, and János Pach. How hard is half-space range searching? *Discrete Comput. Geom.*, 10(2):143–155, 1993. doi:10.1007/BF02573971.

10 Bernard Chazelle. Lower bounds on the complexity of polytope range searching. *J. Amer. Math. Soc.*, 2(4):637–666, 1989. doi:10.2307/1990891.

11 Bernard Chazelle. Lower bounds for orthogonal range searching. I. The reporting case. *J. Assoc. Comput. Mach.*, 37(2):200–212, 1990. doi:10.1145/77600.77614.

12 Bernard Chazelle. Lower bounds for orthogonal range searching. II. The arithmetic model. *J. Assoc. Comput. Mach.*, 37(3):439–463, 1990. doi:10.1145/79147.79149.

13 Bernard Chazelle and Burton Rosenberg. Simplex range reporting on a pointer machine. *Comput. Geom.*, 5(5):237–247, 1996. doi:10.1016/0925-7721(95)00002-X.

14 Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Tóth, editors. *Handbook of discrete and computational geometry*. Discrete Mathematics and its Applications (Boca Raton). CRC Press, Boca Raton, FL, 2018. Third edition of [MR1730156].

15 Larry Guth and Nets Hawk Katz. On the Erdős distinct distances problem in the plane. *Ann. of Math. (2)*, 181(1):155–190, 2015. doi:10.4007/annals.2015.181.1.2.

16 Jiří Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.*, 10(2):157–182, 1993. doi:10.1007/BF02573972.

17 Jiří Matoušek. Geometric range searching. *ACM Comput. Surv.*, 26(4):421–461, 1994. doi:10.1145/197405.197408.

18 Jiří Matoušek and Zuzana Patáková. Multilevel polynomial partitions and simplified range searching. *Discrete Comput. Geom.*, 54(1):22–41, 2015. doi:10.1007/s00454-015-9701-2.

19 Andrew Chi-Chih Yao and F. Frances Yao. A general approach to d-dimensional geometric queries (extended abstract). In Robert Sedgewick, editor, *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 163–168. ACM, 1985. doi:10.1145/22145.22163.

20 A. Young. On Quantitative Substitutional Analysis. *Proc. Lond. Math. Soc.*, 33:97–146, 1901. doi:10.1112/plms/s1-33.1.97.