



**HAL**  
open science

## Deciding Connectivity in Symmetric Semi-Algebraic Sets

Cordian Riener, Robin Schabert, Thi Xuan Vu

► **To cite this version:**

Cordian Riener, Robin Schabert, Thi Xuan Vu. Deciding Connectivity in Symmetric Semi-Algebraic Sets. 2026. ⟨hal-04995130v2⟩

**HAL Id: hal-04995130**

**<https://hal.science/hal-04995130v2>**

Preprint submitted on 15 May 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

# Deciding Connectivity in Symmetric Semi-Algebraic Sets

Cordian Riener

*Department of Mathematics and Statistics, UiT - The Arctic University of Norway, 9037 Tromsø,  
Norway*

Robin Schabert

*Fachbereich Mathematik und Statistik, Universität Konstanz, 78457 Konstanz, Germany*

Thi Xuan Vu

*Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France*

---

## Abstract

A *semi-algebraic set* is a subset of  $\mathbb{R}^n$  defined by a finite collection of polynomial equations and inequalities. In this paper, we investigate the problem of determining whether two points in such a set belong to the same connected component. We focus on the case where the defining equations and inequalities are invariant under the natural action of the symmetric group and where each polynomial has degree at most  $d$ , with  $d < n$  (where  $n$  denotes the number of variables). Exploiting this symmetry, we develop and analyze algorithms for two key tasks. First, we present an algorithm that determines whether the orbits of two given points are connected. Second, we provide an algorithm that decides connectivity between arbitrary points in the set. Both algorithms run in polynomial time with respect to  $n$ .

*Key words:* Multivariate Polynomial Systems, Symmetric Semi-Algebraic Set, Connectivity, Real Solutions, Computational Complexity

---

---

*Email addresses:* `cordian.riener@uit.no` (Cordian Riener), `robin.schabert@uni-konstanz.de` (Robin Schabert), `thi-xuan.vu@univ-lille.fr` (Thi Xuan Vu).

*Preprint submitted to Journal of Symbolic Computation*

## 1. Introduction

A subset of  $\mathbb{R}^n$  is called *semi-algebraic* if it is a finite union of sets defined by finitely many polynomial equalities and inequalities. One of the fundamental questions in the study of semi-algebraic sets is connectivity: given two points  $x, y \in S$ , is there a continuous path within  $S$  that connects them? This problem has been extensively studied in algorithmic geometry and is central to applications such as robot motion planning [43]. In motion planning, connectivity determines whether a robot can transition between two configurations without leaving the feasible space, a setting where semi-algebraic sets naturally describe collision-free regions. Recently, the significance of roadmap algorithms has extended beyond classical robotics. Capco et al. [17, 18] demonstrated that roadmap-based methods can be effectively applied to kinematic singularity analysis, emphasizing the broader scope of connectivity problems in computational geometry. Additionally, the increasing trend toward numerical implementations (see, e.g., [27]) highlights the practical necessity of developing robust connectivity algorithms with efficient computational performance.

In this article, we address the problem of deciding connectivity in symmetric semi-algebraic sets, that is, sets that are invariant under the action of the symmetric group  $\mathfrak{S}_n$ . Such sets appear naturally in many-body physics, algebraic statistics, and computational invariant theory, where symmetries reduce complexity but also introduce new algorithmic challenges. Our main contribution is the development of a polynomial-time algorithm for deciding connectivity within a given symmetric semi-algebraic set. The key insight enabling this approach is a dimension reduction technique, which exploits the structure of symmetric polynomials to simplify the connectivity problem.

The remainder of this paper is organized as follows. In Subsection 1.1, we provide an overview of previous results on connectivity in semi-algebraic sets. Section 2 introduces key preliminaries, including symmetric polynomials, the Weyl chamber, and Vandermonde maps. Section 3 presents essential algorithmic background on methods that will be referenced throughout the paper. In Section 4, we construct a framework that relates the connectivity of a union to connectivity in a bipartite graph. This construction, which follows directly from the Mayer-Vietoris exact sequence, plays a crucial role in our algorithms. Section 5 presents the first main result of this paper, establishing a connection between connectivity in symmetric semi-algebraic sets and *connectivity in orbits* (see Definition 5.2). The latter is a weaker notion, as orbit connectivity is a consequence of general connectivity. We study this concept in detail and provide an algorithm for deciding connectivity in orbits. Section 6 introduces the notion of mirrored spaces and establishes a result that complements orbit connectivity to recover full connectivity. We then use these insights to present our main algorithm. Finally, we conclude with a discussion of open questions and future directions in the last section.

### 1.1. Prior Results

The connectivity problem has received significant attention; however, to the best of our knowledge, no previous work has taken the symmetry into account.

*Computing Roadmaps and Deciding Connectivity.* Schwartz and Sharir [43] pioneered a solution based on Collins' cylindrical algebraic decomposition method. Their approach has a computational complexity that is polynomial in the degree  $d$  and the number of polynomials  $s$ , but doubly exponential in the number of variables  $n$ . Canny [14] introduced the concept of a roadmap for a semi-algebraic set. Roadmaps provide as a tool for counting connected components and verifying whether two points belong to the same connected component. Later improvements [15] enhanced this algorithm, enabling the construction of a roadmap for a semi-algebraic set defined by polynomials, where the sign-invariant sets provide a stratification of  $\mathbb{R}^n$ , with a complexity of  $s^n(\log s)d^{O(n^4)}$ . For arbitrary semi-algebraic sets, Canny applied perturbations to the defining polynomials, allowing the algorithm to determine whether two points lie in the same semi-algebraically connected component while maintaining the same complexity. However, this approach does not provide an explicit path between the points. Furthermore, a Monte Carlo version of the algorithm achieves a complexity of  $s^n(\log s)d^{O(n^2)}$ . Grigor'ev and Vorobjov [25, 16] proposed an algorithm with complexity  $(sd)^{n^{O(1)}}$  for counting the connected components of a semi-algebraic set. Additionally, Heintz, Roy, and Solerno [26], as well as Gournay and Risler [24], developed algorithms capable of computing a roadmap for any semi-algebraic set with the same complexity. Unlike Canny's algorithm, these approaches do not separate the complexity into a combinatorial part (depending on  $s$ ) and an algebraic part (depending on  $d$ ). Since a semi-algebraic set can potentially have up to  $(sd)^n$  connected components, the combinatorial complexity of Canny's algorithm is nearly optimal. In [4], a deterministic algorithm is introduced to construct a roadmap for any semi-algebraic set contained within an algebraic set of dimension  $k$ , achieving a complexity of  $s^{k+1}d^{O(k^2)}$ . This algorithm is particularly significant in robot motion planning, where the robot's configuration space is often modeled as a lower-dimensional algebraic set embedded in a higher-dimensional Euclidean space. A key advantage of this approach is that its complexity depends on the dimension of the algebraic set rather than the ambient space. The algorithm also achieves nearly optimal combinatorial complexity by utilizing only a fixed number of infinitesimal quantities, which reduces the algebraic complexity to  $d^{O(k^2)}$ . Furthermore, it not only identifies connectivity but also computes a semi-algebraic path between input points if they belong to the same connected component, thereby providing the full scope of the problem. However, none of the aforementioned algorithms achieves a complexity lower than  $d^{O(n^2)}$ , nor do they produce a roadmap with a degree below  $d^{O(n^2)}$ . Safey El Din and Schost [41] proposed a probabilistic algorithm that extended Canny's original approach to compute a roadmap for a closed and bounded hypersurface, with a complexity of  $(nd)^{O(n^{1.5})}$ . Later, in [10], the same authors, along with Basu and Roy, introduced a deterministic algorithm for computing a roadmap of a general real algebraic set, achieving a complexity of  $d^{O(n^{1.5})}$ . Basu and Roy, in [9], introduced a divide-and-conquer strategy that reduces the dimension by half at each recursive step, producing a recursion tree of depth  $O(\log(n))$ . Their deterministic algorithm computes a roadmap for a hypersurface in time polynomial in  $n^{n \log^3(n)}d^{n \log^2(n)}$ , with an output size polynomial in  $n^{n \log^2(n)}d^{n \log(n)}$ . Notably, the algorithm's complexity is not polynomial in its output size. Furthermore, despite not assuming smoothness on the hypersurface, the algorithm can address systems of equations by considering the sum of squares of the polynomials. Following the divide-and-conquer strategy, and under smoothness and compactness assumptions, Safey El Din and Schost [42] proposed a probabilistic roadmap algorithm with both output degree and running

time polynomial in  $(nd)^{n \log(k)}$ , where  $k$  denotes the dimension of the considered algebraic set. Note that, the algorithm in [42] achieves a running time that is subquadratic in the output size, with explicitly provided complexity constants in the exponent. Recently, in [32], Prébet, Safey El Din, and Schost established a new connectivity result that extends the one in [42] to the unbounded case. In a separate work [33], they introduced an algorithm for computing roadmaps with the same complexity as that in [42].

*Systems of symmetric polynomials.* This paper explores the connectivity problem in cases where the polynomials describing a semi-algebraic set are symmetric and have low degree in relation to the number of variables. Several studies have highlighted the potential of leveraging the action of the symmetric group in such situations. Timofte [45] established that for symmetric polynomials of lower degree, positivity can be deduced from positivity on low-dimensional test sets, which consist of points with at most half-degree distinct coordinates. Building on this foundation, [34, 35], and [39] demonstrated that this approach can be widely applied to algorithmic methods for polynomial optimization, resulting in polynomial-time complexity. Basu and Riener used a similar approach to compute the equivariant Betti numbers of symmetric semi-algebraic sets [7], the Euler-Poincaré characteristic [6], and the first  $\ell$  Betti numbers [8] of such sets, achieving polynomial-time computation for fixed degree. Our approach is based on ideas inherent in these works as well, namely a reduction of dimension in the case of a degree which is lower than the number of variables. In more recent work, Faugère, Labahn, Safey El Din, Schost, and Vu [20], along with Labahn, Riener, Safey El Din, Schost, and Vu [30], demonstrated that even when the degree is not fixed in advance, significant improvements in complexity can be achieved for certain algorithmic problems in real algebraic geometry such as emptiness decision and polynomial optimization.

### 1.2. Our main results

The main results presented in this paper are described in Sections 5 and 6. On the one hand, we present the first *deterministic polynomial-time* algorithm for deciding connectivity between two  $\mathfrak{S}_n$ -orbits of points in a symmetric semi-algebraic set. This algorithm is described in Algorithm 4 and its complexity is bounded by  $O(n^{d^2})$  arithmetic operations in  $\mathbb{Q}$ . Building on this, we provide, on the other hand, the first *deterministic polynomial-time* algorithm for determining connectivity between two individual points in such a set. This algorithm is described in Algorithm 6 and has complexity  $O\left(s^d n^{d^2} d^{O(d^2)}\right)$  arithmetic operations in  $\mathbb{Q}$ . Both of the quantities describing the complexity of our algorithms are polynomial for fixed degree  $d$ .

### 1.3. Relation to a previous version

This paper is an extended version of our previous work [38], which appeared in the proceedings of the ISSAC 2024 conference. In this version, we provide more detailed explanations, expand upon our previous results, and introduce new findings. In the conference version, we focused exclusively on the case where the points  $x = (x_1, \dots, x_n)$  and  $y = (y_1, \dots, y_n)$  were in canonical form, meaning they satisfied the ordering constraints  $x_1 \leq x_2 \leq \dots \leq x_n$  and  $y_1 \leq y_2 \leq \dots \leq y_n$ . In contrast, the current paper shows how to remove this restriction and provides an algorithm for general real points  $x$  and  $y$ , thereby broadening the applicability of our approach. In particular, Section 6 introduces

new results, including an algorithm for determining the connectivity of arbitrary points in symmetric semi-algebraic sets. Additionally, Sections 3, 4, and 5 extend and generalize certain notations and foundational results.

## 2. Algorithmic preliminaries

This section introduces key algorithmic techniques for analyzing connectivity in semi-algebraic sets. We first discuss the encoding of real algebraic points using univariate representations and Thom encodings. We then provide an overview of roadmap algorithms, summarizing essential results on computing roadmaps and deciding connectivity, which form the basis for the algorithms developed later in the paper.

### 2.1. Encoding of real algebraic points

Our algorithms manipulate points obtained through subroutines, with the outputs having coordinates represented as real algebraic numbers. We encode such a point using univariate polynomials and a Thom encoding.

Let  $\mathcal{F} \subseteq \mathbb{R}[\mathbf{X}]$  and  $x \in \mathbb{R}^n$ , where  $\mathbf{X} = (X_1, \dots, X_n)$ . A mapping  $\zeta : \mathcal{F} \rightarrow \{-1, 0, 1\}$  is called a *sign condition* on  $\mathcal{F}$ . The *sign condition realized by  $\mathcal{F}$*  on  $x$  is defined as  $\text{sign}(\mathcal{F}, x) : \mathcal{F} \rightarrow \{-1, 0, 1\}$ , where  $f \mapsto \text{sign}(f(x))$ . We say  $\mathcal{F}$  *realizes*  $\zeta \in \{-1, 0, 1\}^{\mathcal{F}}$  if and only if  $\text{sign}(\mathcal{F}, x) = \zeta$ .

Let  $\text{Der}(q) := \{q, q^{(1)}, q^{(2)}, \dots, q^{(\deg(q))}\}$  denote a list of polynomials, where  $q^{(i)}$ , for  $i > 0$ , is the formal  $i$ -th order derivative of  $q$ . A mapping  $\zeta$  is called a *Thom encoding* of  $\vartheta$  if  $\text{Der}(q)$  realizes  $\zeta$  on  $x$  and  $\zeta(q) = 0$ . Note that distinct roots of  $q$  in  $\mathbb{R}$  correspond to distinct Thom encodings [5, Proposition 2.28].

**Definition 2.1.** A *real univariate representation* representing  $x \in \mathbb{R}^n$  consists of:

- a zero-dimensional parametrization

$$\mathcal{Q} = (q(T), q_0(T), q_1(T), \dots, q_n(T)),$$

where  $q, q_0, \dots, q_n$  lie in  $\mathbb{Q}[T]$  with  $\gcd(q, q_0) = 1$ , and

- a Thom encoding  $\zeta$  representing an element  $\vartheta \in \mathbb{R}$  such that

$$q(\vartheta) = 0 \quad \text{and} \quad x = \left( \frac{q_1(\vartheta)}{q_0(\vartheta)}, \dots, \frac{q_n(\vartheta)}{q_0(\vartheta)} \right) \in \mathbb{R}^n.$$

We refer readers to [5, Chapters 2 and 12] for details about Thom encodings and univariate representations.

**Example 2.2.** A real univariate representation representing  $x = \left( \frac{1}{2}, \frac{3}{2} - \frac{\sqrt{2}}{4} \right)$  is  $(\mathcal{Q}, \zeta)$ , where

$$\mathcal{Q} = (q, q_0, q_1, q_2) = (T^2 - 2, 2T, T, 3T - 1)$$

and  $\zeta = (1, 1)$  (equivalently,  $\zeta = (+, +) = (q' > 0 \wedge q'' > 0)$ ).

Indeed, let  $\vartheta = \sqrt{2}$  be a root of  $q$ . Then  $x = \left( \frac{q_1(\vartheta)}{q_0(\vartheta)}, \frac{q_2(\vartheta)}{q_0(\vartheta)} \right)$ . Moreover,  $q'(T) = 2T$  and  $q''(T) = 2$ , implying that  $q'(\vartheta) > 0$  and  $q''(\vartheta) > 0$ . Thus,  $\zeta = (+, +)$  serves as a Thom encoding of  $\vartheta$ .

Given a non-zero polynomial  $q \in \mathbb{R}[T]$ , we consider the routine

$$\text{ThomEncoding}(q),$$

which returns the ordered list of Thom encodings of the roots of  $q$  in  $\mathbb{R}$ . This routine can be executed using [5, Algorithm 10.14] with a complexity of  $O(\delta_q^4 \log(\delta_q))$ , where  $\delta_q = \deg(q)$ , involving arithmetic operations in  $\mathbb{Q}$ .

Let  $q \in \mathbb{R}[T]$  be a polynomial of degree  $\delta_q$  and  $p \in \mathbb{R}[T]$  be a polynomial of degree  $\delta_p$ . Additionally, let  $\text{Thom}(q)$  denote the list of Thom encodings of the set of roots of  $q$  in  $\mathbb{R}$ . We define the routine

$$\text{Sign\_ThomEncoding}(q, p, \text{Thom}(q)),$$

which, for each  $\zeta \in \text{Thom}(q)$  specifying the root  $\vartheta$  of  $q$ , returns the sign of  $p(\vartheta)$ . This routine can be implemented using [5, Algorithm 10.15] with a complexity of

$$O(\delta_q^2 (\delta_q \log(\delta_q) + \delta_p))$$

arithmetic operations in  $\mathbb{Q}$ .

## 2.2. Roadmaps and connectivity

In general, a roadmap for a semi-algebraic set is a curve that has a non-empty and connected intersection with all of its connected components. We give a formal definition in the following.

**Definition 2.3.** Let  $S \subset \mathbb{R}^n$  be a semi-algebraic set. A roadmap for  $S$  is a semi-algebraic set  $\text{RM}(S) \subset S$  of dimension at most one, which satisfies the following conditions:

- RM<sub>1</sub>. For every semi-algebraic connected component  $C$  of  $S$ , the intersection  $C \cap \text{RM}(S)$  is a semi-algebraic connected set.
- RM<sub>2</sub>. For every  $x \in \mathbb{R}$  and for every semi-algebraic connected component  $C'$  of  $\{y \in \mathbb{R}^{n-1} : (x, y) \in S\}$ , we have  $C' \cap \text{RM}(S) \neq \emptyset$ .

Let  $\mathcal{M} \subset \mathbb{R}^n$  be a finite set of points. A roadmap for  $(S, \mathcal{M})$  is a semi-algebraic set  $\text{RM}(S, \mathcal{M})$  such that  $\text{RM}(S, \mathcal{M})$  is a roadmap of  $S$ , and  $\mathcal{M} \subset \text{RM}(S, \mathcal{M})$ . Roadmaps can be used to decide the connectivity of semi-algebraic sets. We summarize below the results we use in this paper.

**Theorem 2.4** ([4, Theorem 3] and [5, Theorem 16.14]). Let  $F$  be a sequence of polynomials in  $\mathbb{Q}[\mathbf{X}]$  with the algebraic set  $Z \subset \mathbb{R}^n$  defined by  $F$  of dimension  $k$ . Consider  $G = (g_1, \dots, g_s) \subset \mathbb{Q}[\mathbf{X}]$ . Let  $S$  be a semi-algebraic subset of  $Z$  defined by  $G$ . Let  $d$  be a bound for the degrees of  $F$  and  $G$ . Consider a finite set of points  $\mathcal{M} \subset Z$ , with cardinality  $\delta$ , described by real univariate representations of degree at most  $d^{O(n)}$ . Then the following holds:

- There is an algorithm whose output is exactly one point in every semi-algebraically connected component of  $S$ . The complexity of the algorithm is  $s^{k+1} d^{O(n^2)}$  arithmetic operations in  $\mathbb{Q}$ .
- There exists an algorithm to compute a roadmap for  $(Z, \mathcal{M})$  using

$$\delta^{O(1)} d^{O(n^{1.5})}$$

arithmetic operations in  $\mathbb{Q}$ .

As a consequence, there exists an algorithm for deciding whether two given points, described by real univariate representations of degree at most  $d^{O(n)}$ , belong to the same connected component of  $Z$ , using

$$s^{k+1}d^{O(n^2)}$$

arithmetic operations in  $\mathbb{Q}$ .

In the following sections, we use the notation

$$\text{Normal\_Point\_Connected}(S) \text{ and } \text{Normal\_Connect}(S, x, y)$$

to denote the algorithms mentioned in the theorem above. `Normal_Point_Connected` computes exactly one point per connected component of a semi-algebraic set  $S$  described by polynomials in  $n$  variables of degree at most  $d$ . This algorithm returns at most  $O((sd)^n)$  many points and its complexity is bounded by  $s^n d^{n^2}$  operations in  $\mathbb{Q}$ . Given a semi-algebraic set  $S$  and two real points  $x$  and  $y$ , the algorithm `Normal_Connect`( $S, x, y$ ) returns TRUE if  $x$  and  $y$  belong to the same connected component of  $S$ ; otherwise, it returns FALSE. The complexity of this algorithm is also bounded by  $s^n d^{n^2}$  operations in  $\mathbb{Q}$ .

Note also that there are some other prior results concerning the problem of computing roadmaps computing one point per each connected component, and deciding the connectivity of two given points (see Subsection 1.1), with better complexities. For example, when the input polynomials define an algebraic set, the complexities of these two procedure are  $(nd)^{n \log n}$  by using a deterministic algorithm given in [9]. The procedure returns  $(nd)O(n \log n)$  many points. However, the result stated in the theorem above is sufficient for our main algorithms in Section 5 and Section 6 and in the context of any semi-algebraic set.

### 3. Preliminaries on symmetric polynomials

Symmetric polynomials play a fundamental role in our study of connectivity in semi-algebraic sets. In this section, we provide essential definitions and results on symmetric polynomials, their invariance properties under the symmetric group, and their representations using power sums. We also introduce Weyl chambers and their associated combinatorial structures, which allow us to analyze connectivity in symmetric semi-algebraic sets. Finally, we discuss Vandermonde maps, which provide a useful tool for encoding and understanding these spaces.

#### 3.1. Symmetric polynomials

Throughout the article, we fix integers  $n, d \in \mathbb{N}$  with  $d \leq n$  and denote by  $\mathbb{R}[\mathbf{X}] := \mathbb{R}[X_1, \dots, X_n]$  the real polynomial ring in  $n$  variables. The *basic closed semi-algebraic set* given by polynomials  $f_1, \dots, f_s$  in  $\mathbb{R}[\mathbf{X}]$  is

$$\{x \in \mathbb{R}^n \mid f_i(x) \geq 0 \text{ for all } i = 1, \dots, s\}.$$

A semi-algebraic set is a set generated by a finite sequence of union, intersection and complement operations on basic semi-algebraic sets.

For  $n \in \mathbb{N}_0$ , we denote by  $\mathfrak{S}_n$  the symmetric group on  $n$  elements, which has order  $n!$ . The symmetric group  $\mathfrak{S}_n$  acts naturally on  $\mathbb{R}^n$  by permuting coordinates, and this

action extends to  $\mathbb{R}[\mathbf{X}]$  by permuting the variables. A semi-algebraic set  $S \subset \mathbb{R}^n$  is called *symmetric* if for every  $x \in S$  and  $\sigma \in \mathfrak{S}_n$  we have  $\sigma(x) \in S$ . We say that a polynomial  $f \in \mathbb{R}[\mathbf{X}]$  is symmetric if

$$f(\mathbf{X}) = f(\sigma^{-1}(\mathbf{X})) \quad \text{for all } \sigma \in \mathfrak{S}_n,$$

and we denote the  $\mathbb{R}$ -algebra of symmetric polynomials by  $\mathbb{R}[\mathbf{X}]^{\mathfrak{S}_n}$ .

**Remark 3.1.** Note that clearly every semi-algebraic set which is exclusively defined by symmetric polynomials, is symmetric. While a symmetric semi-algebraic set can sometimes be described using polynomials that are not themselves symmetric, results by Bröcker [13] show that every symmetric semi-algebraic set can always be described by symmetric polynomials. Throughout this article, we assume that any symmetric semi-algebraic set is given by a description involving only symmetric polynomials.

**Definition 3.2.** For  $1 \leq i \leq n$ , we define the  *$i$ -th power sum* as

$$p_i := X_1^i + X_2^i + \cdots + X_n^i.$$

The fundamental theorem of symmetric polynomials states that every symmetric polynomial can be uniquely written in terms of the first  $n$  power sums, i.e., for every symmetric polynomial  $f$ , there is a unique polynomial  $g \in \mathbb{R}[Z_1, \dots, Z_n]$ , where  $Z_1, \dots, Z_n$  are new variables, such that

$$f = g(p_1, \dots, p_n).$$

Furthermore, due to the uniqueness of this representation, it follows that for a symmetric polynomial of degree  $d < n$ , this representation cannot contain power sums of degree higher than  $d$ , i.e., we have:

**Proposition 3.3.** Any symmetric polynomial  $f \in \mathbb{R}[\mathbf{X}]^{\mathfrak{S}_n}$  of degree  $d \leq n$  can be uniquely written as

$$f = g(p_1, \dots, p_d),$$

where  $g$  is a polynomial in  $\mathbb{R}[Z_1, \dots, Z_d]$ .

Note that there is no intrinsic preference for using the power-sum polynomials  $p_i$ . Indeed, any family of algebraically independent symmetric polynomials with degree sequence  $\{1, \dots, n\}$ , for instance, the elementary symmetric polynomials or the complete homogeneous symmetric polynomials, leads to the same result. Moreover, the polynomial  $g$  can be computed algorithmically by using the results of [12, 22] for symmetric polynomials, and the approach of [46] for arbitrary families of algebraically independent polynomials  $p_1, \dots, p_n$ .

### 3.2. Weyl chambers and restriction to faces

We denote by  $\mathcal{W}_c$  the cone defined by  $X_1 \leq \cdots \leq X_n$ , known as the *canonical Weyl chamber* of the action of  $\mathfrak{S}_n$ . It serves as a fundamental domain for this action. The walls of  $\mathcal{W}_c$  contain points where the inequalities are nonstrict at some positions, i.e., points  $x = (x_1, \dots, x_n) \in \mathcal{W}_c$  for which  $x_i = x_{i+1}$  for some  $i \in \{1, \dots, n\}$ . More precisely:

**Definition 3.4.** Let  $n \in \mathbb{N}$  be a fixed integer.

- (1) A sequence of positive integers  $\lambda = (\lambda_1, \dots, \lambda_\ell)$  with  $|\lambda| := \sum_{i=1}^{\ell} \lambda_i = n$  is called a *composition of  $n$  into  $\ell$  parts*. Here,  $\ell$  is called the *length*  $\ell(\lambda)$  of  $\lambda$ , and we denote by  $\text{Comp}(n)$  the set of compositions of  $n$  and by  $\text{Comp}(n, \ell)$  the set of compositions of  $n$  of length  $\ell$ .
- (2) For  $n \in \mathbb{N}$  and  $\lambda = (\lambda_1, \dots, \lambda_\ell) \in \text{Comp}(n)$ , we denote by  $\mathcal{W}_c^\lambda$  the subset of  $\mathcal{W}_c$  defined by

$$X_1 = \dots = X_{\lambda_1} \leq X_{\lambda_1+1} = \dots = X_{\lambda_1+\lambda_2} \leq \dots \leq X_{\lambda_1+\dots+\lambda_{\ell-1}+1} = \dots = X_n.$$

- (3) For every  $\lambda \in \text{Comp}(n, d)$ , the set  $\mathcal{W}_c^\lambda$  defines a  $d$ -dimensional face of the cone  $\mathcal{W}_c$ , and every face is obtained in this way from a composition and denote by  $L_\lambda$  the linear span of  $\mathcal{W}_c^\lambda$ . Note that

$$\dim L_\lambda = \dim \mathcal{W}_c^\lambda = \ell(\lambda).$$

- (4) In the case when  $\ell = n - 1$ , there is a unique  $i \in \{1, \dots, n - 1\}$  such that  $\lambda_i > 1$ . With a slight abuse of notation, we denote the corresponding facet of the canonical Weyl chamber,  $\mathcal{W}_c^\lambda$ , also as  $\mathcal{W}_c^{(i, i+1)}$ .
- (5) Finally, for every  $d$ , we define the set of *alternate odd* compositions  $\text{CompMax}(n, d)$  as

$$\{\lambda = (\lambda_1, \dots, \lambda_d) \in \text{Comp}(n) \mid \lambda_{2i+1} = 1 \text{ for all } 0 \leq i < d/2\}.$$

Let  $n \in \mathbb{N}$ , and let  $\lambda, \mu \in \text{Comp}(n)$ . We write  $\lambda \prec \mu$  if  $\mathcal{W}_c^\lambda \supset \mathcal{W}_c^\mu$ . Equivalently,  $\lambda \prec \mu$  if  $\mu$  can be obtained from  $\lambda$  by replacing some of the commas in  $\lambda$  with  $+$  signs. It follows that  $\prec$  defines a *partial order* on  $\text{Comp}(n)$ , making  $\text{Comp}(n)$  a *poset*. For  $\lambda, \mu \in \text{Comp}(n)$ , the *smallest composition greater than both  $\lambda$  and  $\mu$*  (with respect to  $\prec$ ) is called their *join*. The face corresponding to the join of  $\lambda$  and  $\mu$  is given by  $\mathcal{W}_c^\lambda \cap \mathcal{W}_c^\mu$ .

For  $x \in \mathcal{W}_c$ , we denote by  $\mathcal{C}(x)$  the largest composition  $\lambda$  of  $n$ , such that  $x$  can be written as

$$x = (\underbrace{z_1, \dots, z_1}_{\lambda_1\text{-times}}, \underbrace{z_2, \dots, z_2}_{\lambda_2\text{-times}}, \dots, \underbrace{z_\ell, \dots, z_\ell}_{\lambda_\ell\text{-times}}),$$

where  $z = (z_1, \dots, z_\ell) \in \mathbb{R}^\ell$ , and  $\ell$  is the length of  $\lambda$ . For a composition  $\lambda = (\lambda_1, \dots, \lambda_\ell) \in \text{Comp}(n)$  and  $f \in \mathbb{R}[\mathbf{X}]$ , we define the polynomial

$$f^{[\lambda]} := f(\underbrace{X_1, \dots, X_1}_{\lambda_1\text{-times}}, \underbrace{X_2, \dots, X_2}_{\lambda_2\text{-times}}, \dots, \underbrace{X_\ell, \dots, X_\ell}_{\lambda_\ell\text{-times}}).$$

Notice that  $f^{[\lambda]}$  is a polynomial in  $\ell$  variables and its image is exactly the image of  $f$  when restricted to  $\mathcal{W}_c^\lambda$ . More generally, given a symmetric semi-algebraic set  $S \subset \mathbb{R}^n$  we denote by  $S_c = S \cap \mathcal{W}_c$  and given a partition  $\lambda \in \text{Comp}(n)$  we denote  $S_c^{[\lambda]}$  the intersection of  $S_c$  with  $\mathcal{W}_c^\lambda$ . If  $S$  is described by symmetric polynomials, then  $S_c^{[\lambda]}$  is just obtained by replacing every polynomial  $f$  in the description by  $f^{[\lambda]}$ .

**Example 3.5.** Consider  $n = 4$ . There exist eight compositions of 4 listed as

$$\{(4), (3, 1), (1, 3), (2, 2), (2, 1, 1), (1, 2, 1), (1, 1, 2), (1, 1, 1, 1)\}.$$

Notably, for  $\mu$  in the set  $\{(3, 1), (1, 3), (4)\}$ , we have  $(1, 2, 1) \prec \mu$ . Furthermore, the join of  $(2, 1, 1)$  and  $(1, 1, 1, 1)$  is  $(2, 1, 1)$ .

Given  $x = (-1, 5, 5, -3)$ , we determine  $\mathcal{C}(x) = (1, 2, 1)$ . Lastly, considering the polynomial  $f = x_3^3 + x_1x_2 - x_4$  and the composition  $\lambda = (1, 2, 1)$ , it follows that  $f^{[\lambda]} = x_1x_2 + x_2^3 - x_3$ .

Given a general  $x \in \mathbb{R}^n$  we know that there is a unique element  $x' \in \mathcal{W}_c$  in the  $\mathfrak{S}_n$ -orbit of  $x$ . This  $x'$  is obtained by ordering the coordinates of  $x$  increasingly. This can be done algorithmically with the classical sorting algorithm bubble sort. This algorithm can also be used to obtain the unique permutation  $\sigma \in \mathfrak{S}_n$  that transforms  $x$  into  $x'$ . Since  $\sigma \in \mathfrak{S}_n$  we know that we can write  $\sigma$  as a product of pairwise transposition  $s_i = (i, i+1)$  for  $i \in \{1, \dots, n\}$  and we will use the following variant of bubble sort to obtain the minimal set of transpositions necessary to describe  $\sigma$ .

---

**Algorithm 1** Minimal\_Adjacent\_Transpositions

---

**Input:** Vector  $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ .

**Output:** A minimal sequence  $T$  of adjacent transpositions sorting  $x$ , and the sorted vector  $x' \in \mathbb{R}^n$  of  $x$ .

- (1) Initialize an empty list  $T$  of transpositions.
  - (2) **for**  $i = 1$  to  $n - 1$  **do**
    - (a) **for**  $j = 1$  to  $n - i$  **do**
      - (i) **if**  $x_j > x_{j+1}$  **then**
        - (A) Swap  $x_j$  and  $x_{j+1}$ .
        - (B) Append transposition  $(j, j + 1)$  to  $T$ .
  - (3) Return  $(T, x)$ .
- 

**Theorem 3.6.** The above Algorithm correctly computes the minimal decomposition of  $T$  into adjacent transpositions. Its complexity is  $O(n^2)$ .

*Proof.* Correctness follows from the observation that each adjacent transposition reduces the permutation's inversion number by exactly one. Since sorting requires eliminating all inversions, this process necessarily yields a minimal decomposition into adjacent transpositions. The complexity result is immediate, as the procedure is equivalent to bubble sort, which is well-known to have a worst-case complexity of  $O(n^2)$ .  $\square$

**Remark 3.7.** Note that there are other sorting algorithms with a better complexity. However, while bubble sort is not the most efficient sorting algorithm in general, the task of our Algorithm 1 is to obtain the minimal sequence of adjacent transpositions that give the sorted vector. As more efficient sorting algorithms use swaps of non-adjacent coordinates, the inherent complexity of the requirement – specifically, determining the minimal sequence of adjacent transpositions – necessitates an  $O(n^2)$  complexity.

### 3.3. Vandermonde maps

**Definition 3.8.** For  $m = (m_1, \dots, m_n) \in \mathbb{N}^n$ , we define the weighted power sum  $p_j^{(m)} := \sum_{i=1}^n m_i X_i^j$ .

**Remark 3.9.** Clearly, the usual power sum polynomials correspond to trivial weights  $m = (1, \dots, 1)$ . Furthermore, given a composition  $\lambda$  of  $n$  we directly see that the restriction of the  $i$ -th power sum polynomial  $p_i$  to the Wall  $\mathcal{W}_c^\lambda$ , denoted by  $p_j^{[\lambda]}$ , as defined above coincided with  $p_j^{(\lambda)}$ .

**Definition 3.10.** For  $d \in \{1, \dots, n\}$  and  $m \in \mathbb{N}^n$ , we define the (weighted)  $d$ -Vandermonde map:

$$\begin{aligned} \nu_{n,d,m} : \mathbb{R}^n &\longrightarrow \mathbb{R}^n \\ x &\longmapsto (p_1^{(m)}(x), \dots, p_d^{(m)}(x)) \end{aligned}$$

When  $m = (1, \dots, 1)$ , we denote the map by  $\nu_{n,d}$ . When  $d = n$ , and  $m = (1, \dots, 1)$  we simply refer to it as the Vandermonde map.

For  $a = (a_1, \dots, a_d) \in \mathbb{R}^d$ , the fiber of the  $d$ -Vandermonde map, i.e., the set

$$V(a) := \left\{ x \in \mathbb{R}^n \mid p_1^{(m)}(x) = a_1, \dots, p_d^{(m)}(x) = a_d \right\},$$

is called the  $m$ -weighted *Vandermonde variety* with respect to  $a$ . Note that, when  $n = d$  and  $m = (1, \dots, 1)$ , every non-empty Vandermonde variety is precisely the orbit of a single point  $x \in \mathbb{R}^n$ .

Arnold, Givental, and Kostov emphasized the significance of the Vandermonde map in their research on hyperbolic polynomials, and this topic has been explored further by [1]. The required properties for our studies are summarized as follows.

**Theorem 3.11** ([1, 2, 23, 29]). The Vandermonde map (i.e., when  $d = n$  and  $m = (1, \dots, 1)$ ) gives rise to a homeomorphism between the canonical Weyl chamber  $\mathcal{W}_c$  and its image. Moreover, for any composition  $\lambda$ , the weighted Vandermonde map  $\nu_{n,\ell(\lambda),\lambda}$  provides a homeomorphism between  $\mathcal{W}_c^\lambda$  and its image. Furthermore, for every  $a \in \mathbb{R}^d$  and  $m \in \mathbb{Z}^d$  the  $m$ -weighted Vandermonde variety  $V_m(a)$  is either contractible or empty when intersected to the Weyl chamber  $\mathcal{W}_c$ .

#### 4. Graphs and connectivity of unions

A central part of our algorithm will require that we algorithmically understand the connectivity of unions of semi-algebraic sets. Our main tool for studying connectivity in a union is based on relating it to graphs, for which efficient algorithms for connectivity are available. The key here is that the connected components of a union  $\bigcup_i S_i$  are completely determined by the connected components of the individual sets  $S_i$  and their pairwise intersections  $S_i \cap S_j$ . It is closely related to the nerve theorem in combinatorial topology [3, 11] and appears implicitly in many algorithms for computing connected components of semi-algebraic sets, such as cylindrical algebraic decomposition and roadmap-based approaches [5, Chapter 15]. Here we make it explicit by encoding the connectivity of the union in a bipartite graph, whose vertices represent connected components of the individual sets and their pairwise intersections, and whose edges connect components that meet non-trivially. We give the detailed construct of this graph in Algorithm 2.

**Definition 4.1.** Let  $G = (V, E)$  be a graph. Given two vertices  $u, v \in V$  we say that they are *connected* if there exists a sequence of vertices  $v_1, v_2, \dots, v_k$  where  $v_1 = u$ ,  $v_k = v$ , and  $(v_i, v_{i+1}) \in E$  for all  $1 \leq i < k$ . Moreover, a *connected component* of  $G$  is a maximal set of vertices  $C \subseteq V(G)$  such that for every pair of vertices  $v, u$  in  $C$ , there is a path in  $G$  between  $v$  and  $u$ .

Consider a collection of semi-algebraic sets  $S_1, \dots, S_k \subset \mathbb{R}^n$ . Our goal is to analyze the connectivity properties of their union  $\bigcup_{i=1}^k S_i$ . The key observation is that the connected components of a union  $\bigcup_i S_i$  are completely determined by the connected components of the individual sets  $S_i$  and their pairwise intersections  $S_i \cap S_j$ . This idea is classical and appears implicitly in many algorithms for computing connected components of semi-algebraic sets, such as cylindrical algebraic decomposition and roadmap-based approaches. Here we make it explicit by encoding the connectivity of the union in a bipartite graph, whose vertices represent connected components of the individual sets and their pairwise intersections, and whose edges connect components that meet non-trivially. We give the detailed construct of this graph in Algorithm 2.

---

**Algorithm 2** Construct Bipartite Graph from Semi-Algebraic Sets

---

**Input:** a collection of semi-algebraic sets  $S_1, \dots, S_k$

**Output:** a bipartite graph  $G(A \cup B, E)$  where  $A$  and  $B$  are vertex sets representing connected components of the sets, and  $E$  is a set of edges indicating connectivity between these components

- (1) initialize  $A, B, E$  as empty sets
  - (2) **for** each pair of indices  $i, j$  with  $1 \leq i, j \leq k$  and  $i \neq j$  **do**
    - (a) compute one point per connected component of  $S_{(i,j)} = S_i \setminus S_j$  and  $S_{(j,i)} = S_j \setminus S_i$ . Add the points together with the ordered pairs  $(i, j)$  and  $(j, i)$  into  $A$
    - (b) compute one point per connected component of  $S^{i,j} = S_i \cap S_j$  and add each point to  $B$  together with the tuple  $(i, j)$
  - (3) **for** each  $u \in A$  **do**
    - (a) **for** each  $w \in B$  **do**
      - (i) determine the semi-algebraic set  $S_i$  to which both  $u$  and  $w$  belong by inspecting the associated ordered pair  $(i, j)$  and the associated tuple  $\{u, w\}$
      - (ii) **if**  $u, w$  are in the same set  $S_i$  **then**
        - (A) **if**  $\text{CONNECT}(u, w, S_i)$  **then** add edge  $(u, w)$  to  $E$
  - (4) **return**  $G(A \cup B, E)$
- 

**Theorem 4.2.** Let  $S_1, \dots, S_k \subset \mathbb{R}^n$  be semi-algebraic sets defined by polynomials of degree at most  $d$ . The connected components of the union  $S = \bigcup_{i=1}^k S_i$  are in one-to-one correspondence with the connected components of the bipartite graph  $G$  constructed by Algorithm 2.

We give a short independent proof for the convenience of the reader.

*Proof.* The vertex set  $A$  represents points from the differences  $S_i \setminus S_j$ ,  $B$  represents points from the intersections  $S_i \cap S_j$ , and  $E$  represents the connectivity between these points in  $A$  and  $B$ . To show that we have an injective pairing: assume two distinct connected components in  $S$ , say  $C_1$  and  $C_2$ , map to the same connected component in  $G$ . By construction, vertices in  $A$  and  $B$  correspond to unique connected components of  $S_i \setminus S_j$  and  $S_i \cap S_j$ , respectively. Since  $C_1$  and  $C_2$  are distinct in  $S$ , they must yield distinct sets of points in  $A$  and  $B$ , contradicting the assumption that they map to the same component in  $G$ . Hence, the mapping is injective. To show the surjectivity of the pairing: consider a connected component in  $G$ , represented by a subset of vertices in  $A \cup B$  and edges in

$E$ . By construction, each vertex in  $A$  and  $B$  corresponds to a connected component in the differences  $S_i \setminus S_j$  or intersections  $S_i \cap S_j$  of the semi-algebraic sets. Since edges in  $E$  represent connectivity between these components, the corresponding points in  $S$  form a connected subset. This subset corresponds to a connected component in  $S$ , ensuring that every component in  $G$  maps back to a component in  $S$ , establishing surjectivity.  $\square$

**Theorem 4.3** ([28, Chapter 8]). Let  $G = (V, E)$  be an undirected graph. There exists an algorithm that computes all connected components of  $G$  with complexity  $O(|V| + |E|)$ . Moreover, the space complexity is  $O(|V| + |E|)$ .

Using this graph algorithm in combination with Theorem 4.2 we thus have a convenient way of computing the connected components of a union.

## 5. Connectivity in the orbit space

The main goal of this section is to study the connectivity between two points  $x = (x_1, x_2, \dots, x_n)$  and  $y = (y_1, y_2, \dots, y_n)$  in  $\mathbb{R}^n$ , with their coordinates arranged in ascending order, i.e.,  $x_1 \leq x_2 \leq \dots \leq x_n$  and  $y_1 \leq y_2 \leq \dots \leq y_n$ .

In the presence of group invariance, it is natural to reformulate algorithms in a symmetry-reduced setup. For example, one is no longer interested in all solutions to a system of equations but only in all solutions up to symmetry. This corresponds to reducing the set of solutions to the orbit space, by which we mean the following

**Definition 5.1** (Orbit Space). Let  $G$  be a finite group acting on  $\mathbb{R}^n$ . The *orbit space*, denoted by  $\mathbb{R}^n/G$ , is defined as the quotient space under the equivalence relation identifying points belonging to the same  $G$ -orbit. Formally,

$$\mathbb{R}^n/G := \{Gx : x \in \mathbb{R}^n\}, \quad \text{where } Gx = \{g(x) \mid g \in G\}.$$

In particular, the orbit space can be identified with a fundamental domain and thus in the case of a group  $G$  generated by reflections, with the *canonical Weyl chamber*. The notation of the orbit space motivates the following definition.

**Definition 5.2** (Connectivity up to Orbits). Let  $G$  be a finite group acting on  $\mathbb{R}^n$ , and let  $S \subseteq \mathbb{R}^n$  be a  $G$ -invariant semi-algebraic set. Two points  $x, y \in S$  are said to be *connected up to orbits* (or *orbit-connected*) if their orbits  $Gx$  and  $Gy$  lie in the same connected component of the orbit space  $S/G$ .

To relate this definition to the connectivity of two points we remark the following connection between the two notions.

**Remark 5.3.** Let  $S$  be a  $G$ -invariant semi-algebraic set and consider two points  $x, y \in S$ . If  $x, y$  are connected by a continuous path  $\gamma : [0, 1] \rightarrow S$ , the path  $\gamma$  itself descends naturally to a continuous path in the quotient space. Thus, the orbits  $Gx, Gy$  trivially lie within the same connected component in the orbit space  $S/G$ . On the other hand, two points  $x, y \in S$  may be connected up to orbits without necessarily being connected within the original set  $S$ . For instance, distinct points in the same orbit are always orbit-connected but need not lie in the same connected component of  $S$ .

**Example 5.4.** Let  $n \in \mathbb{N}$  and consider the symmetric polynomial

$$f_n := \sum_{i=1}^n X_i^2 (X_i - 1)^2$$

which has  $2^n$  zeros. We consider now the symmetric semi-algebraic set

$$S_n := \{x \in \mathbb{R}^n \mid f_n - \varepsilon \leq 0\},$$

which for small enough  $\varepsilon > 0$  consist of  $2^n$  connected components. Up to orbits,  $S_n$  has  $n + 1$  connected components.

So, while orbit connectivity of two points is not equivalent to the connectivity of the two points, the failure of orbit connectivity can be used to decide that the two points are not connected. For the setup of this article, we fix the canonical Weyl chamber  $\mathcal{W}_c$  as a model of the orbit space. For every point  $x \in \mathbb{R}^n$  obtain a unique representative of  $x$  in  $\mathcal{W}_c$  by ordering the coordinates of  $x$  increasingly. Thus, for two points  $x$  and  $y$  in  $S$  orbit connectivity is equivalent to stating that their unique representatives  $x'$  and  $y'$  are connected in  $S \cap \mathcal{W}_c$ . In the following, we will thus assume that the two points  $x = (x_1, \dots, x_n)$  and  $y = (y_1, \dots, y_n)$  are in the same fundamental domain of the action of the symmetric group, by assuming that the coordinates of the two given points are sorted in non-decreasing order. That is, throughout this section we will assume that  $x_1 \leq x_2 \leq \dots \leq x_n$  and  $y_1 \leq y_2 \leq \dots \leq y_n$ .

### 5.1. Restriction to subspaces

The key to our algorithmic improvements in the case of orbit connectivity lies in a reduction of dimension. This reduction of dimension rests on the following result of Arnold.

**Theorem 5.5** ([2, Theorem 7]). Let  $a \in \mathbb{R}^d$  and consider the  $d$ -Vandermonde variety  $V(a) \subset \mathbb{R}^n$ . A point  $x \in V(a) \cap \mathcal{W}_c$  is a minimizer of  $p_{d+1}$  if and only if  $x \in \mathcal{W}_c^\lambda$  for some  $\lambda \in \text{CompMax}(n, d)$ .

This result also motivates the following.

**Definition 5.6.** Given  $n, d \in \mathbb{N}$ , we denote

$$\mathcal{W}_c^d = \bigcup_{\lambda \in \text{CompMax}(n, d)} \mathcal{W}_c^\lambda.$$

Furthermore, for a semi-algebraic subset  $S \subset \mathbb{R}^n$ , we denote

$$S_{n, d} = S \cap \mathcal{W}_c^d,$$

which we call the  $d$ -dimensional orbit boundary of  $S$ . Notice that if  $d \geq n$ , then  $S_{n, d} = S \cap \mathcal{W}_c$ .

The first claim below arises directly from Theorem 5.5. Similarly, the subsequent statement can be deduced with Morse theoretic arguments built on Theorem 5.5 (see [7, Proposition 7] for further specifics).

**Theorem 5.7.** Let  $S \subset \mathbb{R}^n$  be a semi-algebraic set defined by polynomials of degree  $d$ . Then every connected component of  $S \cap \mathcal{W}_c$  intersects with  $S_{n,d}$ . In fact,  $S_{n,d}$  is a retraction of  $S \cap \mathcal{W}_c$ .

With the following Lemma and the subsequent Algorithm, we algorithmically obtain the dimension reduction.

**Lemma 5.8.** Let  $S \subset \mathbb{R}^n$  be a semi-algebraic set defined by symmetric polynomials of degree at most  $d$  with  $d \leq n$  and let  $x \in S \cap \mathcal{W}_c$ . Furthermore, let

$$a = (p_1(x), \dots, p_d(x))$$

and assume that

$$x' \in V(a) \cap \mathcal{W}_c$$

is a minimizer of  $p_{d+1}$  on  $V(a) \cap \mathcal{W}_c$ . Then  $x$  and  $x'$  are connected inside  $S \cap \mathcal{W}_c$ .

*Proof.* First observe that for every point

$$z \in V(a) \cap \mathcal{W}_c$$

we have  $z \in S$ . Indeed, since  $x, z \in V(a)$ ,

$$p_i(x) = p_i(z), \quad i = 1, \dots, d.$$

Since the defining polynomials of  $S$  are symmetric of degree at most  $d$ , Proposition 3.3 implies that each defining polynomial can be written as a polynomial in

$$p_1, \dots, p_d.$$

Hence all defining polynomials take the same values at  $x$  and  $z$ . Therefore  $z \in S$ .

Now, by the contractibility result for Vandermonde varieties inside the canonical Weyl chamber, the set

$$V(a) \cap \mathcal{W}_c$$

is contractible and therefore path-connected. Since

$$V(a) \cap \mathcal{W}_c \subseteq S \cap \mathcal{W}_c,$$

it follows that  $x$  and  $x'$  are connected inside  $S \cap \mathcal{W}_c$ .  $\square$

**Theorem 5.9.** Let  $V(a)$  be a non-empty Vandermonde variety, and let  $\mathcal{W}_c$  be the canonical Weyl-chamber. Then there is a unique minimizer of  $p_{d+1}$  on  $V(a) \cap \mathcal{W}_c \neq \emptyset$  with a multiplicity composition that is bigger or equal to a composition  $\lambda$  of length  $d$  and  $\lambda_d = \lambda_{d-2} = \lambda_{d-4} = \dots = 1$ .

Furthermore, there exists an algorithm  $\text{Min\_Canonical}(a, \mathcal{W}_c)$  that takes  $a \in \mathbb{R}^d$  and the canonical Weyl chamber  $\mathcal{W}_c$  as input and returns a real univariate representation representing this minimizer of  $p_{d+1}$  on  $V(a) \cap \mathcal{W}_c$  with a time complexity of

$$O\left(\binom{n - \lceil d/2 \rceil - 1}{\lfloor d/2 \rfloor - 1} d^{4d+1} \log(d)\right) = O\left(n^{d/2} d^d\right)$$

arithmetic operations in  $\mathbb{Q}$ .

---

**Algorithm 3** Min\_Canonical( $a, \mathcal{W}_c$ )

---

**Input:** a point  $a \in \mathbb{R}^d$  and the canonical Weyl-chamber  $\mathcal{W}_c$

**Output:** a real univariate representation representing the minimizer of  $p_{d+1}$  on  $V(a) \cap \mathcal{W}_c$

**for**  $\lambda \in \text{Comp}(n, d)$  with  $\lambda_d = \lambda_{d-2} = \dots = 1$  **do**

- (1) compute polynomials  $p^{[\lambda]} = (p_1^{[\lambda]} - a_1, \dots, p_d^{[\lambda]} - a_d)$
- (2) find a zero-dimensional parametrization  $\mathcal{Q}$  of  $V(p^{[\lambda]})$

$$\mathcal{Q} = (q(T), q_0(T), q_1(T), \dots, q_d(T)) \in \mathbb{Q}[T]^{d+2}$$

- (3) define  $\mathcal{Q}' \in \mathbb{Q}[T]^{n+2}$  with

$$\mathcal{Q}' = (q(T), q_0(T), \underbrace{q_1(T), \dots, q_1(T)}_{\lambda_1\text{-times}}, \dots, \underbrace{q_d(T), \dots, q_d(T)}_{\lambda_d\text{-times}})$$

- (4) compute  $\text{Thom}(q) = \text{ThomEncoding}(q)$
- (5) find  $\Sigma_0 = \text{Sign\_ThomEncoding}(q, q_0, \text{Thom}(q))$
- (6) **for**  $i = 1, \dots, d-1$  **do** find

$$\Sigma_i = \text{Sign\_ThomEncoding}(q, q_i - q_{i+1}, \text{Thom}(q))$$

- (7) **for**  $i = 1, \dots, d$  and  $(\zeta_\vartheta, \sigma_i) \in \Sigma_i$  **do**
  - (a) **if**  $\sigma_0 > 0$  and  $\sigma_i \leq 0$  for all  $i = 1, \dots, d-1$   
or  $\sigma_0 < 0$  and  $\sigma_i \geq 0$  for all  $i = 1, \dots, d-1$ 
    - (i) **return**  $(\mathcal{Q}', \zeta_\vartheta)$

---

*Proof.* The first part of the theorem is obtained from the results in [2, 31].

For the second part, we consider the polynomial optimization problem

$$(\mathbf{P}) : \min_{x \in V(a) \cap \mathcal{W}_c} p_{d+1}(x),$$

where  $V(a)$  is the Vandermonde variety with respect to  $a$ .

Let  $J$  denote the Jacobian matrix of  $(p_1(\mathbf{X}) - a_1, \dots, p_d(\mathbf{X}) - a_d, p_{d+1}(\mathbf{X}))$  with respect to  $\mathbf{X}$ , i.e.,

$$J = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 2X_1 & 2X_2 & \dots & 2X_n \\ \vdots & & & \vdots \\ (d+1)X_1^d & (d+1)X_2^d & \dots & (d+1)X_n^d \end{pmatrix}.$$

Any minimizer  $x' \in V(a) \cap \mathcal{W}_c$  lying in the relative interior of a face of  $\mathcal{W}_c$  is a critical point of  $p_{d+1}$  restricted to the corresponding stratum of  $V(a)$ . This means that  $x'$  belongs to the set of points where the Jacobian matrix  $J$  has rank at most  $d$ , i.e., all  $(d+1)$ -minors of  $J$  vanish. Furthermore, any  $(d+1)$ -minor of  $J$  has the form

$$\alpha \cdot \prod_{\substack{i_1 \leq i < j \leq i_{d+1}, \\ \{i_1, \dots, i_{d+1}\} \subset \{1, \dots, n\}}} (X_i - X_j),$$

where  $\alpha \in \mathbb{R}_{\neq 0}$ . Therefore, for any point  $x \in \mathbb{R}^n$  with more than  $d$  distinct coordinates, there exists a  $(d+1)$ -minor of  $J$  such that this minor does not vanish at  $x$ . This implies

that any optimizer has at most  $d$  distinct coordinates.

Let  $\lambda = (\lambda_1, \dots, \lambda_d)$  be a composition of  $n$  of length  $d$  with  $\lambda_d = \lambda_{d-2} = \dots = 1$ . Then the optimizers of the problem **(P)** of type  $\gamma$ , where  $\gamma$  is a composition of  $n$  with  $\lambda \preceq \gamma$ , have the form

$$x = (\underbrace{x_1, \dots, x_1}_{\lambda_1\text{-times}}, \underbrace{x_2, \dots, x_2}_{\lambda_2\text{-times}}, \dots, \underbrace{x_d, \dots, x_d}_{\lambda_d\text{-times}}),$$

where  $\bar{x} = (x_1, \dots, x_d)$  is a solution of

$$p_1^{[\lambda]} - a_1 = \dots = p_d^{[\lambda]} - a_d = 0. \quad (1)$$

Since the composition has length  $d$ , the system (1) is zero-dimensional in  $d$  variables, and its solution set is given by a zero-dimensional parametrization. We then check for the existence of a real point  $x = (x_1, \dots, x_d)$  in this solution set such that  $x_i \leq x_{i+1}$  for  $i = 1, \dots, d-1$ . To do this, we follow the following steps.

Let  $\mathcal{Q} = (q(T), q_0(T), q_1(T), \dots, q_d(T))$  be a zero-dimensional parametrization for the solution set of (1), and  $\text{Thom}(q)$  be the ordered list of Thom encodings of the roots of  $q$  in  $\mathbb{R}$ . Suppose  $\zeta_\vartheta \in \text{Thom}(q)$  is a Thom encoding of a real root  $\vartheta$  of  $q$ . Then

$$\bar{x} = \left( \frac{q_1(\vartheta)}{q_0(\vartheta)}, \dots, \frac{q_d(\vartheta)}{q_0(\vartheta)} \right) \in \mathbb{R}^d.$$

Since  $\gcd(q, q_0) = 1$ ,  $q_0$  does not vanish at  $\vartheta$ . If  $q_0(\vartheta) > 0$ , then  $x_1 \leq x_2 \leq \dots \leq x_d$  if and only if  $q_i(\vartheta) - q_{i+1}(\vartheta) \leq 0$  for all  $i = 1, \dots, d-1$ . Otherwise, when  $q_0(\vartheta) < 0$ ,  $x_1 \leq x_2 \leq \dots \leq x_d$  if and only if  $q_i(\vartheta) - q_{i+1}(\vartheta) \geq 0$  for all  $i = 1, \dots, d-1$ . Thus, it is sufficient to test the signs of polynomials  $q_0(T)$  and  $q_i(T) - q_{i+1}(T)$  at  $\vartheta$ . Computing  $\text{Thom}(q)$  can be done using the `ThomEncoding` procedure, while checking the signs of  $q_0$  and  $q_i - q_{i+1}$  is obtained using the `Sign_ThomEncoding` subroutine. For each admissible composition  $\lambda$ , the above procedure produces all real ordered solutions on the corresponding face. Among all candidates obtained in this way, we evaluate  $p_{d+1}$  and retain the one with minimal value. By the uniqueness statement in the first part of the theorem, this candidate is precisely the minimizer of  $p_{d+1}$  on  $V(a) \cap W_c$ . In summary, the `Min_Canonical` algorithm is given in Algorithm 3.

We finish the proof by analyzing the complexity of the `Min_Canonical` algorithm. First since  $\lambda_d = \lambda_{d-2} = \dots = 1$ , it is sufficient to consider compositions of  $n - \lceil d/2 \rceil$  of length  $\lfloor d/2 \rfloor$ , which is equal to  $\binom{n - \lceil d/2 \rceil - 1}{\lfloor d/2 \rfloor - 1}$ .

For a fixed composition  $\lambda$  of length  $d$ , finding a zero-dimensional parametrization for the solution set of (1) requires  $O(D^3 + dD^2)$  arithmetic operations in  $\mathbb{Q}$ , where  $D$  is the number of solutions of (1), which is also equal to the degree of  $q(T)$ . This can be achieved using algorithms such as those presented in [40]. Since the system (1) is zero-dimensional with  $d$  polynomials and  $d$  variables, Bézout's Theorem gives a bound for  $D \leq d^d$ . Therefore, the total complexity of finding a zero-dimensional parametrization for the solution set of (1) is  $O(d^{3d})$  arithmetic operations in  $\mathbb{Q}$ .

Finally, the subroutine `ThomEncoding(q)`, which returns the ordered list of Thom encodings of the roots of  $q$  in  $\mathbb{R}$ , requires

$$O(D^4 \log(D)) = O(d^{4d} \log(d^d)) = O(d^{4d+1} \log(d)).$$

Since the degrees of  $q_0$  and  $q_i$  (for  $i = 1, \dots, d$ ) are at most that of  $q$ , the procedure `Sign_ThomEncoding` to determine the sign of each of these polynomials at the real roots of  $q$  is

$$O(D^2(D \log(D) + D)) = O(d^{2d}(d^d \log(d^d) + d^d)) = O(d^{3d+1} \log(d))$$

operations in  $\mathbb{Q}$ .

Thus, the total complexity of the `Min_Canonical` algorithm is

$$O\left(\binom{n - \lceil d/2 \rceil - 1}{\lfloor d/2 \rfloor - 1} d^{4d+1} \log(d)\right) = O^\sim\left(n^{d/2} d^d\right)$$

arithmetic operations in  $\mathbb{Q}$ . Finally, among all ordered real solutions obtained from the candidate faces, we evaluate the value of  $p_{d+1}$ . Since Theorem 5.9 ensures that the minimizer is unique inside  $V(a) \cap W_c$ , the solution with minimal value of  $p_{d+1}$  is precisely the desired point. This finishes our proof.  $\square$

## 5.2. Algorithm for testing the connectivity within one Weyl chamber

We are now able to present the first main result of this article: an algorithm designed to determine the connectivity in orbits of two points, or equivalently for two points in  $\mathbb{R}^n$  with coordinates ordered increasingly.

Consider two points  $x = (x_1, x_2, \dots, x_n)$  and  $y = (y_1, y_2, \dots, y_n)$  in  $\mathbb{R}^n$ , with their coordinates arranged in ascending order,  $x_1 \leq x_2 \leq \dots \leq x_n$  and  $y_1 \leq y_2 \leq \dots \leq y_n$ . Prior to introducing the algorithm, we will define some notations.

**Notation 5.1.** Consider  $1 \leq d \leq n$  to be an integer, and let  $z \in W_c$ . Define  $a_1 = p_1(z), \dots, a_d = p_d(z)$ . We will denote the Vandermonde variety  $V(a_1, \dots, a_d)$  by  $V_{d,z}$ . Moreover, let  $z'$  denote the unique point within  $V_{d,z}$  that is guaranteed by Theorem 5.9.

**Theorem 5.10.** Let  $S \subset \mathbb{R}^n$  be a semi-algebraic set defined by symmetric polynomials of degree at most  $d$  with  $d \leq n$ , and take  $x, y \in S \cap W_c$ . Then,  $x$  and  $y$  are connected within  $S \cap W_c$  if and only if the corresponding points  $x'$  and  $y'$  are connected within  $S_{n,d}$ .

*Proof.* By Lemma 5.8,  $x$  and  $y$  are both connected to  $x'$  and  $y'$ , respectively. Furthermore, since by Theorem 5.7, the set  $S_{n,d}$  is a retraction of  $S \cap W_c$ , we have that  $x'$  and  $y'$  are connected within  $S \cap W_c$  if and only if they are connected within  $S_{n,d}$ .  $\square$

We can now formulate an algorithm for checking equivariant connectivity of semi-algebraic sets defined by symmetric polynomials of degree at most  $d$ .

**Theorem 5.11.** Given symmetric polynomials  $f_1, \dots, f_s \in \mathbb{R}[\mathbf{X}]^{\mathfrak{S}_n}$  defining a semi-algebraic set  $S$  and two points  $x, y \in S \cap W_c$ , the Algorithm 4 correctly decides connectivity. Moreover, for fixed  $d$  and  $s$ , its complexity can be estimated by  $O(n^{d^2})$ .

*Proof.* The correctness follows from Lemma 5.8 in combination with Theorem 4.2. For the complexity of the algorithm, we observe that the main cost lies in building the graph  $G$  with the help of Algorithm 2. Notice that this algorithm uses a roadmap algorithm for the  $d$ -dimensional semialgebraic sets defined by  $F^{[\lambda]}$ . Therefore, following Theorem 2.4, each call is made to compute the points per connected component or to decide the connectivity costs  $s^{d+1} d^{O(d^2)}$ , which are constant by assumption.

---

**Algorithm 4** Connectivity\_Symmetric\_Canonical( $F, (x, y)$ )

---

**Input:**  $F = (f_1, \dots, f_s) \in \mathbb{R}[\mathbf{X}]^{\mathfrak{S}_n}$  of degrees at most  $d$  defining a semi-algebraic set  $S$ ; two points  $x$  and  $y$  in  $\mathcal{W}_c$

**Output:** TRUE if  $x$  and  $y$  are in the same connected of  $S$ ; otherwise return FALSE

- (1) compute a list  $L$  of alternate odd compositions of  $n$  into  $d$  parts
  - (2) **for**  $\lambda$  in  $L$  **do** compute  $F^{[\lambda]} := (f_1^{[\lambda]}, \dots, f_s^{[\lambda]})$ .
  - (3) compute bipartite graph  $G = (A \cup B, E)$  for  $\bigcup_{\lambda \in L} S(F^{[\lambda]})$
  - (4) compute connected components of  $G$
  - (5) compute  $a = (\nu_{n,d}(x))$  and  $b = (\nu_{n,d}(y))$
  - (6) compute  $x' = \text{Min\_Canonical}(a, \mathcal{W}_c)$  and  $y' = \text{Min\_Canonical}(b, \mathcal{W}_c)$
  - (7) find the compositions  $\gamma = \mathcal{C}(x')$  of  $x'$  and  $\eta = \mathcal{C}(y')$  of  $y'$
  - (8) find  $v_x, v_y \in A$  with
    - $\text{Normal\_Connect}(F^{[\gamma]}, (v_x, x')) = \text{TRUE}$  and
    - $\text{Normal\_Connect}(F^{[\eta]}, (v_y, y')) = \text{TRUE}$ .
  - (9) **if**  $v_x$  and  $v_y$  are in the same connect component of  $G$  return TRUE, else return FALSE
- 

Moreover, the number of sets we consider

$$\kappa(d) = O\left(\binom{n - \lceil d/2 \rceil - 1}{\lfloor d/2 \rfloor - 1} d^{4d+1} \log(d)\right) = O\left(n^{d/2} d^d\right),$$

which is the number of alternate odd compositions of  $n$  with  $d$  parts. Thus, the first loop has length  $\binom{\kappa(d)}{2}$ . Each step of the loop has a fixed complexity, which is at most  $3s^{d+1}d^{O(d^2)}$ .

For the second loop, note that we can bound the number of connected components of each of the sets  $S(F^{[\lambda]})$  from above by  $O((sd)^d)$ . Therefore, the second loop has length at most  $\kappa(d)^2 O((sd)^{2d})$ , and each step has complexity at most  $s^{d+1}d^{O(d^2)}$ . Since  $s$  and  $d$  are fixed, we arrive at the announced complexity.  $\square$

## 6. Connectivity across Weyl chambers

In the previous section, we provided an effective method for determining connectivity within the canonical Weyl chamber  $\mathcal{W}_c$ , i.e., for deciding orbit connectivity. However, as highlighted in Remark 5.3, orbit connectivity of two points does not in itself guarantee that the two points are connected in  $S$ . In this section, we analyze how paths in a symmetric set  $S$  interact with the Weyl chamber decomposition of  $\mathbb{R}^n$  and derive a criterion that complements orbit connectivity to recover full connectivity. The geometric picture is that of a kaleidoscope: the walls of the Weyl chamber act as mirrors, and the reflected copies of  $S_c = S \cap \mathcal{W}_c$  build up the entire symmetric set  $S$ . Whether two points in  $S$  are connected depends on whether a path can be routed through the appropriate sequence of reflections.

**Remark 6.1.** The interplay between a fundamental domain and reflected copies is studied systematically in the theory of *mirrored spaces* and the *Basic Construction* of Davis [19]; see also [8]. In particular, [8, Proposition 4] shows that the Davis construction  $\mathcal{U}(\mathfrak{S}_n, S_c)$  is semi-algebraically homeomorphic to  $S$ . While the full machinery of mirrored

spaces provides decompositions of  $\mathbf{H}_*(\mathcal{U}(\mathcal{G}, Z))$  into Solomon modules [44, 19], the connectivity question we address here only involves  $\mathbf{H}_0$  and admits a direct path-based treatment, which we give below.

### 6.1. Paths and wall-crossings

The *Weyl chamber decomposition* of  $\mathbb{R}^n$  is defined by the reflection hyperplanes

$$H_{i,j} := \{x_i = x_j\} \text{ with } 1 \leq i < j \leq n.$$

Their complement in  $\mathbb{R}^n$  consists of  $n!$  open Weyl chambers  $\sigma(\mathcal{W}_c^\circ)$  for  $\sigma \in \mathfrak{S}_n$ , where  $\mathcal{W}_c^\circ = \{x_1 < \dots < x_n\}$ , and their closures  $\sigma(\mathcal{W}_c)$ , with  $\mathcal{W}_c = \{x_1 \leq \dots \leq x_n\}$ , partition  $\mathbb{R}^n$  into  $n!$  closed chambers forming the Coxeter complex of type  $A_{n-1}$ . In this complex two closed chambers  $\sigma_1(\mathcal{W}_c)$  and  $\sigma_2(\mathcal{W}_c)$  share a facet if and only if  $\sigma_1^{-1}\sigma_2$  is an adjacent transposition  $(i, i+1)$ ; the shared facet lies on the hyperplane  $\{x_{\sigma_1(i)} = x_{\sigma_1(i+1)}\}$ .

**Lemma 6.2** (Path decomposition). Let  $S \subset \mathbb{R}^n$  be a symmetric semi-algebraic set. Let  $x \in S \cap \mathcal{W}_c$  and for a  $\sigma \in \mathfrak{S}_n$  let  $y \in S \cap \sigma(\mathcal{W}_c)$  such that both  $x$  and  $y$  lie in the same connected component of  $S$ . Then:

- (i) There exist adjacent transpositions  $t_1, \dots, t_m$  such that  $\sigma = t_m \cdots t_1$ , and for each  $j$  there is a point  $q_j \in S$  on the wall corresponding to  $t_j$  that is connected to  $x$  in  $S$ .
- (ii) The points  $x$  and  $y' := \sigma^{-1}(y)$  are connected in  $S_c := S \cap \mathcal{W}_c$ .
- (iii) Each wall-crossing point  $q_j$ , reflected to its canonical representative  $q'_j := \sigma_{j-1}^{-1}(q_j) \in S_c$ , lies on the facet  $\mathcal{W}_c^{(i_j, i_j+1)} := \{x_{i_j} = x_{i_j+1}\}$  and is connected to  $x$  in  $S_c$ .

*Proof.* Since  $S$  is semi-algebraic, two points in the same connected component of  $S$  can be joined by a semi-algebraic path (see [5, Proposition 5.23]). Let  $\phi: [0, 1] \rightarrow S$  be such a path with  $\phi(0) = x$  and  $\phi(1) = y$ , and write  $\phi = (\phi_1, \dots, \phi_n)$ .

(i). For each pair  $1 \leq i < j \leq n$ , define  $f_{i,j}(t) = \phi_i(t) - \phi_j(t)$ . Each  $f_{i,j}$  is semi-algebraic, so its zero set  $f_{i,j}^{-1}(0) \subset [0, 1]$  is a finite union of points and intervals. In particular, the sign of each  $f_{i,j}$  changes only finitely many times. Since there are finitely many pairs  $(i, j)$ , there exists a finite subdivision

$$0 = \tau_0 < \tau_1 < \dots < \tau_{m+1} = 1$$

such that on each open interval  $(\tau_{j-1}, \tau_j)$  all signs of the  $f_{i,j}$  are constant (after removing repetitions, we may assume consecutive chambers are distinct). Thus

$$\phi((\tau_{j-1}, \tau_j)) \subset \sigma_{j-1}(\mathcal{W}_c^\circ)$$

for uniquely determined permutations  $\sigma_0, \dots, \sigma_m$ , with  $\sigma_0 = \text{id}$  and  $\sigma_m = \sigma$ . Each wall-crossing point  $q_j := \phi(\tau_j)$  lies in  $\overline{\sigma_{j-1}(\mathcal{W}_c)} \cap \overline{\sigma_j(\mathcal{W}_c)}$ . Since these two chambers share a facet, they differ by an adjacent transposition  $t_j = (i_j, i_j+1)$  with  $\sigma_j = t_j \sigma_{j-1}$ . Hence  $\sigma = t_m \cdots t_1 \in \langle t_1, \dots, t_m \rangle$ , and each  $q_j$  is connected to  $x$  in  $S$  via the sub-path  $\phi|_{[0, \tau_j]}$ .

(ii). Decompose  $\phi$  into sub-paths  $\phi_j := \phi|_{[\tau_j, \tau_{j+1}]}$  for  $j = 0, \dots, m$ . Define the reflected paths

$$\tilde{\phi}_j := \sigma_j^{-1} \circ \phi_j.$$

Since  $\phi_j$  takes values in  $\overline{\sigma_j(\mathcal{W}_c)}$ , the reflected path  $\tilde{\phi}_j$  takes values in  $\mathcal{W}_c$ , and by symmetry of  $S$ , in  $S_c$ . At each wall-crossing point  $q_{j+1} = \phi(\tau_{j+1})$ , the point  $q_{j+1}$  lies on the

hyperplane fixed by the adjacent transposition  $t_{j+1}$  (in  $\sigma_j$ -coordinates), so  $\sigma_j^{-1}(q_{j+1})$  is fixed by the conjugate  $\sigma_j^{-1}t_{j+1}\sigma_j$ , which gives

$$\sigma_j^{-1}(q_{j+1}) = \sigma_{j+1}^{-1}(q_{j+1}),$$

i.e.,  $\tilde{\phi}_j(1) = \tilde{\phi}_{j+1}(0)$ . The reflected segments therefore concatenate to a continuous path in  $S_c$  from  $x$  to  $y' = \sigma^{-1}(y)$ .

(iii). By the same reflection,  $q'_j := \sigma_{j-1}^{-1}(q_j)$  lies in  $\mathcal{W}_c$  (since  $q_j \in \overline{\sigma_{j-1}(\mathcal{W}_c)}$ ) and in  $S$  by symmetry, hence in  $S_c$ . The point  $q'_j$  lies on the facet  $\mathcal{W}_c^{(i_j, i_{j+1})}$  because  $q_j$  is fixed by the reflection  $\sigma_{j-1} \cdot (i_j, i_{j+1}) \cdot \sigma_{j-1}^{-1}$ , so  $q'_j$  is fixed by  $(i_j, i_{j+1})$ . Since the reflected path from (ii) passes through  $q'_j$  and connects to  $x$  inside  $S_c$ , the claim follows.  $\square$

## 6.2. The connectivity criterion

**Definition 6.3.** Let  $S \subseteq \mathbb{R}^n$  be a symmetric semi-algebraic set, and let  $S_c = S \cap \mathcal{W}_c$ . For a connected component  $\mathcal{C}$  of  $S_c$ , define the *wall subgroup*

$$G(\mathcal{C}) \subseteq \mathfrak{S}_n$$

to be the subgroup generated by all adjacent transpositions  $(i, i+1)$  such that  $\mathcal{C} \cap \mathcal{W}_c^{(i, i+1)} \neq \emptyset$ .

The value of this subgroup is made explicit in the following.

**Lemma 6.4.** With the definitions above we have that for any  $\tau \in G(\mathcal{C})$  and any  $z \in \mathcal{C}$ , the points  $z$  and  $\tau(z)$  are connected in  $S$ .

*Proof.* We proceed by induction, with the base case ( $\tau = \text{id}$ ) being trivially true. Now, assume first that  $\tau = t = (i, i+1)$  is a generator of  $G(\mathcal{C})$ . In this situation we know by definition that there exists  $q \in \mathcal{C} \cap \mathcal{W}_c^{(i, i+1)}$  such that there is a path  $\beta$  from  $z$  to  $q$  in  $S_c$ . Furthermore, since  $q$  lies on the wall  $\{x_i = x_{i+1}\}$ , we have  $t(q) = q$ . The image path  $t \circ \beta$  goes from  $t(z)$  to  $t(q) = q$  and lies in  $t(S_c) \subset S$ . Concatenating  $\beta$  with  $t \circ \beta$  (traversed in reverse) yields a path

$$z \xrightarrow{\beta} q = t(q) \xrightarrow{\overline{t \circ \beta}} t(z)$$

in  $S$ .

Now, to do the inductive step, we assume  $\tau = t \cdot \tau'$  where  $t$  is a generator of  $G(\mathcal{C})$  and  $\tau'$  has shorter word length. Let  $q \in \mathcal{C} \cap \mathcal{W}_c^{(i, i+1)}$  be the wall-touching point for the generator  $t$ , and let  $\beta_z$  be a path from  $z$  to  $q$  in  $S_c$ . By the inductive hypothesis applied to  $q$ , we find a path  $\gamma_q$  from  $q$  to  $\tau'(q)$  in  $S$ . Since  $t(q) = q$ , the image  $t(\gamma_q)$  is a path from  $t(q) = q$  to  $t(\tau'(q)) = \tau(q)$  in  $S$ . Furthermore,  $\tau(\beta_z) = t(\tau'(\beta_z))$  is a path from  $\tau(z)$  to  $\tau(q)$  in  $S$  (since  $\beta_z \subset S_c \subset S$  and  $S$  is  $\mathfrak{S}_n$ -invariant). Concatenating

$$z \xrightarrow{\beta_z} q \xrightarrow{t(\gamma_q)} \tau(q) \xrightarrow{\overline{\tau(\beta_z)}} \tau(z).$$

we arrive at a path in  $S$  that connects  $z$  and  $\tau(z)$ .  $\square$

Combining Lemma 6.2 and Lemma 6.4 we arrive at the following characterization for connectivity of two points

**Theorem 6.5.** Let  $S \subset \mathbb{R}^n$  be a symmetric semi-algebraic set and consider two points  $x \in S_c$  and  $y \in S$ . Let  $y' \in S_c$  be the canonical representative of  $y$  and let  $\sigma \in \mathfrak{S}_n$  a permutation satisfy  $y = \sigma(y')$ . Then the two points  $x \in S_c$  and  $y \in S$  belong to the same connected component of  $S$  if and only if the following two conditions are met:

- (i)  $x$  and  $y'$  lie in the same connected component  $\mathcal{C}$  of  $S_c$ , and
- (ii)  $\sigma \in G(\mathcal{C})$ .

*Proof.* The forward direction essentially follows directly from Lemma 6.2. Suppose that  $x$  and  $y$  belong to the same connected component of  $S$ . By Lemma 6.2(ii),  $x$  and  $y'$  are connected in  $S_c$ , thus condition (1) is satisfied. Furthermore, Lemma 6.2(i) tells us that there exist adjacent transpositions  $t_1, \dots, t_k$  with  $\sigma \in \langle t_1, \dots, t_k \rangle$ . By Lemma 6.2(iii), each reflected wall-crossing point  $q'_j$  lies in  $\mathcal{C} \cap \mathcal{W}_c^{(i_j, i_{j+1})}$ , so each  $t_j$  is a generator of  $G(\mathcal{C})$ , whence  $\sigma \in G(\mathcal{C})$ . This gives condition (ii). Now, to see the converse direction, we suppose that conditions (i) and (ii) hold. So we assume that there exists a path  $\alpha$  from  $x$  to  $y'$  in  $S_c \subset S$ . It therefore suffices to construct a path from  $y'$  to  $\sigma(y') = y$  in  $S$ . Now, applying Lemma 6.4 with  $z = y' \in \mathcal{C}$  (by condition (i)) and  $\tau = \sigma \in G(\mathcal{C})$  (by condition (ii)), we obtain a path from  $y'$  to  $\sigma(y') = y$  in  $S$ . Concatenating with  $\alpha$  gives a path from  $x$  to  $y$  in  $S$ .  $\square$

**Remark 6.6.** The inductive construction in the converse direction may produce a path whose length (in the sense of wall-crossings) is larger than the word length  $k$  of  $\sigma$ . This is unavoidable in general: the path must detour through wall-touching points  $q \in \mathcal{C}$  that may be far from the direct geometric route. However, the argument is purely existential; for the algorithmic applications in the next subsection, we only need to *decide* connectivity, not to construct an explicit connecting path.

**Corollary 6.7.** Let  $S \subset \mathbb{R}^n$  be a symmetric semi-algebraic set with  $S_c$  connected. Then  $S$  is connected if and only if  $S_c \cap \mathcal{W}_c^{(i, i+1)} \neq \emptyset$  for every  $i \in \{1, \dots, n-1\}$ .

*Proof.* If  $S_c \cap \mathcal{W}_c^{(i, i+1)} \neq \emptyset$  for every  $i \in \{1, \dots, n-1\}$ , then every adjacent transposition  $(i, i+1)$  belongs to  $G(S_c)$ . Since adjacent transpositions generate  $\mathfrak{S}_n$ , we obtain  $G(S_c) = \mathfrak{S}_n$ . Now let  $x, y \in S$ . Since  $S_c$  is connected, the canonical representatives  $x', y' \in S_c$  belong to the same connected component of  $S_c$ . Moreover, if  $y = \sigma(y')$ , then  $\sigma \in \mathfrak{S}_n = G(S_c)$ . Hence Theorem 6.5 implies that  $x$  and  $y$  lie in the same connected component of  $S$ . Therefore  $S$  is connected. Conversely, suppose that  $S$  is connected. Since  $S_c$  is connected, Theorem 6.5 implies that  $G(S_c) = \mathfrak{S}_n$ . In particular, every adjacent transposition  $(i, i+1)$  belongs to  $G(S_c)$ . By definition of  $G(\mathcal{C})$ , this means  $S_c \cap \mathcal{W}_c^{(i, i+1)} \neq \emptyset$  for every  $i \in \{1, \dots, n-1\}$ .  $\square$

### 6.3. Algorithm for checking connectivity

We are now in the position to give an algorithm that takes two points  $x, y \in S$  and decides if they are connected in  $S$ . Without loss of generality, we can assume that  $x$  has ordered coordinates, i.e.,  $x \in S_c$ , since otherwise we can reiterate the same argumentation by replacing the canonical Weyl chamber  $\mathcal{W}_c$  with the Weyl chamber that  $x$  is contained in. Our approach now builds on Theorem 6.5. For this, we need to find the set of facets of  $\mathcal{W}_c$  to which  $x$  can be connected in  $S_c$ . As an observation for this, we use the following reduction which is a consequence of applying Arnold's result to the intersection of  $S_c$  with the face  $\mathcal{W}_c^{(i, i+1)}$ .

**Definition 6.8.** Let  $n, d \in \mathbb{N}$  and let  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_d)$  be a composition of  $n$  into  $d$  parts. For a given index  $i \in \{1, \dots, n-1\}$ , we define  $\text{RES}_i(\lambda)$ , the composition of  $n$  that additionally incorporates the constraint  $x_i = x_{i+1}$ , as follows. Let

$$\pi_k = 1 + \sum_{j=1}^{k-1} \lambda_j, \quad \text{for } k \geq 2, \quad \pi_1 = 1,$$

and let  $k$  and  $k'$  be the unique indices such that

$$\pi_k \leq i < \pi_{k+1}, \quad \pi_{k'} \leq i+1 < \pi_{k'+1}.$$

Then,

$$\text{RES}_i(\lambda) = \begin{cases} \lambda, & \text{if } k = k', \text{ (i.e., } x_i = x_{i+1} \text{ is already implied)} \\ (\lambda_1, \dots, \lambda_{k-1}, \lambda_k + \lambda_{k'}, \lambda_{k'+1}, \dots, \lambda_d), & \text{otherwise.} \end{cases}$$

The above definition also gives a direct algorithm to compute this new composition out of a given  $\lambda$  in  $O(d)$ .

**Lemma 6.9.** Let  $S \subset \mathbb{R}^n$  be a symmetric semi-algebraic set defined by symmetric polynomials of degree at most  $d$  and  $S_c \subset \mathcal{W}_c$  be the intersection of  $S$  with  $\mathcal{W}_c$ . Then, every connected component of  $S_c \cap \mathcal{W}_c^{(i, i+1)}$  is connected to a point in  $S_c^{\text{RES}(\lambda)}$  for some  $\lambda \in \text{CompMax}(n, d)$ .

*Proof.* Consider the composition  $m \in \mathbb{N}^{n-1}$  defined by

$$m_j = \begin{cases} 2, & \text{if } j = i, \\ 1, & \text{otherwise.} \end{cases}$$

For any  $x \in S_c \cap \mathcal{W}_c^{(i, i+1)}$ , define the associated symmetric polynomial invariants

$$a_j = p_j^{(m)}(x).$$

The (weighted) Vandermonde variety  $V(a)$  contains  $x$  and encodes the structure of connected components.

By Theorem 5.5 (applied to the weighted Vandermonde map  $\nu_{n-1, d, m}$ ), the minimum of  $p_{d+1}^{(m)}$  on  $V(a) \cap \mathcal{W}_c^{(i, i+1)}$  must be attained on a face corresponding to some  $\text{RES}(\lambda)$ , where  $\lambda \in \text{CompMax}(n, d)$ . Since the weighted Vandermonde variety  $V(a) \cap \mathcal{W}_c^{(i, i+1)}$  is contractible, every point in the connected component of  $x$  is connected to the minimizer, which lies in  $S_c^{\text{RES}(\lambda)}$ . This completes the proof.  $\square$

We will use Lemma 6.9 to decide if a given point  $x \in \mathcal{W}_c$  is connected to a wall  $\mathcal{W}_c^{(i, i+1)}$ . Indeed, the following algorithm is a clear consequence.

**Lemma 6.10.** The `Connected_Wall` algorithm correctly decides if the connected component of  $x \in S_c$  is path-connected to the wall  $\mathcal{W}_c^{(i, i+1)}$ . The worst-case complexity is bounded by

$$O\left(\binom{n - \lceil d/2 \rceil - 1}{\lfloor d/2 \rfloor - 1} \cdot d^{4d+1} \log(d) \cdot \left((sd)^n n^{d^2} + s^{d+1} d^{O(d^2)}\right)\right).$$

---

**Algorithm 5** `Connected_Wall( $F, x, i$ )`

---

**Input:**  $F = (f_1, \dots, f_s) \in \mathbb{R}[\mathbf{X}]^{\mathfrak{S}_n}$  is a set of symmetric polynomials of degree at most  $d$ , defining a semi-algebraic set  $S$ ; a point  $x \in \mathcal{W}_c$ ; and  $i \in \mathbb{N}$ .

**Output:** TRUE if  $x$  can be connected in  $S_c$  to the wall  $\mathcal{W}_c^{(i, i+1)}$ , otherwise return FALSE.

- (1) compute a list  $L$  of alternate odd compositions of  $n$  into  $d$  parts.
- (2) **for** each  $\lambda$  in  $L$ , compute the modified polynomial set:

$$F^{[\text{RES}_i(\lambda)]} := (f_1^{[\text{RES}_i(\lambda)]}, \dots, f_s^{[\text{RES}_i(\lambda)]}).$$

- (3)  $L^{[\text{RES}_i(\lambda)]} \leftarrow \text{Normal\_Connect\_Points}(F^{[\text{RES}_i(\lambda)]})$  a list of points in  $S_c^{[\text{RES}_i(\lambda)]}$ .
  - (4) **for** each  $z \in L^{[\text{RES}_i(\lambda)]}$ :
    - (a) **if** `Connectivity_Symmetric_Canonical( $F^{[\text{RES}_i(\lambda)]}, (x, z)$ )` is TRUE, return TRUE.
  - (5) return FALSE.
- 

*Proof.* It follows directly from Lemma 6.9 that we can restrict to the different  $d$ -dimensional faces  $\mathcal{W}_c^{[\text{RES}_i(\lambda)]}$ . If one of these contains a point  $z$  connected in  $S_c$  to  $x$  we have established a connection to  $\mathcal{W}_c^{(i, i+1)}$ . However, in the case when none of the connected components in one  $\mathcal{W}_c^{[\text{RES}_i(\lambda)]}$  can be connected in  $S_c$  to  $x$  we have established that there is no path connection to  $\mathcal{W}_c^{(i, i+1)}$ . Since every one of the calls of `Normal_Connect_Points` has as input a  $d$ -dimensional set, we can deduce that the driving factor comes from the number of elements in `CompMax` and get the announced bound.  $\square$

Finally, this gives rise to the overall algorithm.

---

**Algorithm 6** `Connectivity_Symmetric( $F, (x, y)$ )`

---

**Input:**  $F = (f_1, \dots, f_s) \in \mathbb{R}[\mathbf{X}]^{\mathfrak{S}_n}$  is a set of symmetric polynomials of degree at most  $d$ , defining a semi-algebraic set  $S$ . Two points  $x, y \in \mathbb{R}^n$  are given, with  $x \in \mathcal{W}_c$  (the canonical Weyl chamber).

**Output:** TRUE if  $x$  and  $y$  are connected in  $S$ , otherwise return FALSE.

- (1) compute  $(y', T) \leftarrow \text{Minimal\_Adjacent\_Transpositions}(y)$ .

$\triangleright y'$ : a vector in  $\mathbb{R}^n$

$\triangleright T$ : a sequence of adjacent transpositions

- (2) **if** `Connectivity_Symmetric_Canonical( $F, (x, y')$ )` is FALSE, then return FALSE.
  - (3) compute a list  $L$  of alternate odd compositions of  $n$  into  $d$  parts.
  - (4) **for** each  $(i, i+1) \in T$  **do**:
    - (a) **if** `Connected_Wall( $x, i$ )` is FALSE, return FALSE.
  - (5) return TRUE.
- 

**Theorem 6.11.** Algorithm 6 correctly decides if two points  $x$  and  $y$  are connected within a symmetric semi-algebraic set  $S$ . If  $s$  is the number of symmetric polynomials in the description of  $S$  and if their maximal degree is fixed to be  $d$ , the worst-case complexity of the `Connectivity_Symmetric` algorithm is given by:

$$O\left(s^d n^{d^2} d^{O(d^2)}\right).$$

*Proof.* The correctness of the algorithm builds on Theorem 6.5. If in the second step `Connectivity_Symmetric_Canonical( $F, (x, y')$ )` returns FALSE, then  $x$  and  $y'$  lie in different connected components of  $S_c$ , so condition (1) fails and  $x$  and  $y$  are not connected. Otherwise, if it returns TRUE, Theorem 6.5 tells us that we additionally need to check whether the permutation  $\sigma$  that maps  $y$  to  $y'$  belongs to the wall subgroup  $G(\mathcal{C})$ , where  $\mathcal{C}$  is the connected component of  $S_c$  containing  $x$ . Since  $\sigma \in G(\mathcal{C})$  if and only if each adjacent transposition in the decomposition  $T$  of  $\sigma$  is a generator of  $G(\mathcal{C})$ , it suffices to check, for each  $(i, i+1) \in T$ , whether the connected component of  $x$  in  $S_c$  meets the wall  $W_c^{(i, i+1)}$ . Following Lemma 6.10, Algorithm `Connected_Wall` correctly decides this for each relevant wall. This guarantees that Algorithm 6 returns TRUE if and only if  $x$  and  $y$  are connected in  $S$ .

For the complexity, we obtain

$$O\left(n^2 + n^{d^2} + n^2 \cdot \binom{n - \lceil d/2 \rceil - 1}{\lfloor d/2 \rfloor - 1} \cdot d^{4d+1} \log(d) \cdot \left((sd)^d n^{d^2} + s^{d+1} d^{O(d^2)}\right)\right),$$

where the  $n^2$  terms account for sorting and the loop over at most  $n-1$  transpositions. Since  $d$  and  $s$  are treated as fixed constants, the binomial coefficient is  $O(n^{\lfloor d/2 \rfloor - 1})$ , and all factors not involving  $n$  are absorbed into the constant. The dominant term in  $n$  is  $n^{d^2}$ , yielding the announced bound.  $\square$

#### 6.4. A degree-three example in four variables

To close we exemplify Algorithm 6 and the ideas used on a simple symmetric semi-algebraic set in  $\mathbb{R}^4$ . Let

$$p_k = x_1^k + x_2^k + x_3^k + x_4^k, \quad k = 1, 2, 3,$$

and let  $W_c = \{x_1 \leq x_2 \leq x_3 \leq x_4\}$  be the canonical Weyl chamber. Consider the set

$$S = \{x \in \mathbb{R}^4 \mid p_1(x) = 0, p_2(x) \leq 12, p_3(x) \geq 6\}.$$

We use the four points

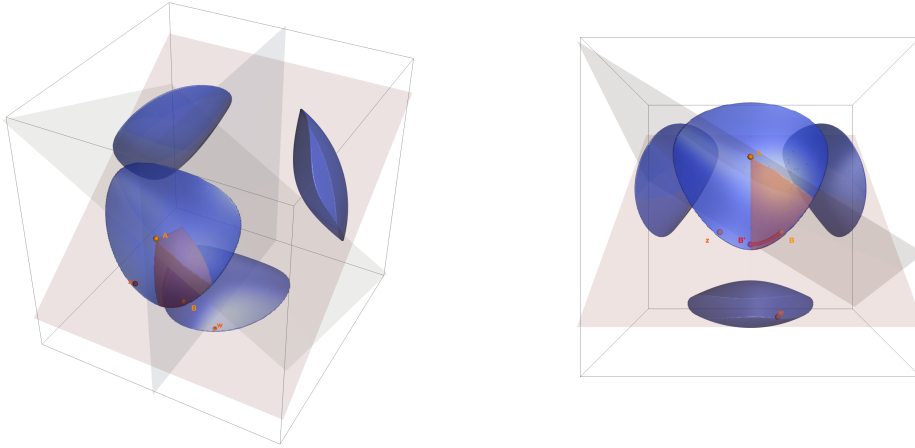
$$A = (-1, -1, -1, 3), \quad B = \left(-2, -1, \frac{1}{2}, \frac{5}{2}\right), \quad z = \left(-2, \frac{1}{2}, -1, \frac{5}{2}\right), \quad w = \left(-2, -1, \frac{5}{2}, \frac{1}{2}\right).$$

One verifies that  $A$  and  $B$  belong to  $S \cap W_c$ : for  $A$  one has  $p_1 = 0, p_2 = 12, p_3 = 24$ , and for  $B$  one has  $p_1 = 0, p_2 = \frac{23}{2}, p_3 = \frac{27}{4}$ . Furthermore,  $z$  and  $w$  are permutations of  $B$  lying outside  $W_c$ , each with canonical representative  $B$ . Note that  $A$  lies on the walls  $\{x_1 = x_2\}$  and  $\{x_2 = x_3\}$  of  $W_c$ , while  $B$  lies in the interior of  $W_c$  (all four coordinates are distinct).

**Orbit boundary and wall analysis.** In order to simplify the presentation, we make some preliminary observations, all of which can also be seen in Figure 1.

The set  $\text{CompMax}(4, 3)$  consists of the single composition  $(1, 2, 1)$ , corresponding to the face  $W_c^{(1, 2, 1)} = \{x_1 \leq x_2 = x_3 \leq x_4\}$ . Let  $S_c = S \cap W_c$ . By Theorem 5.7, the orbit boundary  $S_{4,3} = S \cap W_c^{(1, 2, 1)}$  is a retract of  $S_c$ , so the connected components of  $S_c$  are in bijection with those of  $S_{4,3}$ .

Following Definition 6.3, we determine the wall subgroup  $G(S_c) \subseteq \mathfrak{S}_4$ . The point  $A = (-1, -1, -1, 3)$  lies in  $S_c$  and satisfies  $x_1 = x_2 = x_3 = -1$ , so  $A$  belongs to the walls  $W_c^{(1, 2)} = \{x_1 = x_2\}$  and  $W_c^{(2, 3)} = \{x_2 = x_3\}$ . In particular,  $S_c \cap W_c^{(1, 2)}$  and  $S_c \cap W_c^{(2, 3)}$  are both nonempty, so  $s_1, s_2 \in G(S_c)$ .



(a) View showing the four lobes of  $S$ . The permutation walls (rose-tinted planes) cut out the orbit space in the Weyl chamber  $W_c$  which contains  $A$  and  $B$ ;  $z$  lies in an adjacent chamber,  $w$  in a separated lobe.

(b) Top view. The red curve is the Vandermonde fiber  $V(0, \frac{23}{2}, \frac{27}{4})$  through  $B$ , with  $B'$  marking its retraction to the orbit boundary  $W_c^{(1,2,1)}$  (Algorithm 3). The lower wall  $\{x_3 = x_4\}$  separates  $w$  from  $S \cap W_c$ .

Fig. 1. The set  $S = \{p_1 = 0, p_2 \leq 12, p_3 \geq 6\} \subset \mathbb{R}^4$ , projected onto the hyperplane  $\{p_1 = 0\} \cong \mathbb{R}^3$ . The three permutation walls  $\{x_i = x_{i+1}\}$  are shown as semi-transparent planes.

For the wall  $W_c^{(3,4)} = \{x_3 = x_4\}$ , we show that  $S_c \cap W_c^{(3,4)} = \emptyset$ . Indeed, on  $\{x_3 = x_4 = v\} \cap W_c \cap \{p_1 = 0\}$  one has  $x_1 + x_2 = -2v$  with  $x_1 \leq x_2 \leq v$ . This forces  $v \geq 0$  (if  $v < 0$  then  $x_1 \leq x_2 \leq v < 0$ , so  $x_1 + x_2 < 0$ , contradicting  $x_1 + x_2 = -2v > 0$ ). Using  $p_1 = 0$  and  $x_1 + x_2 = -2v$ , we compute  $x_1^3 + x_2^3 = (x_1 + x_2)^3 - 3x_1x_2(x_1 + x_2) = -8v^3 + 6vx_1x_2$  and  $x_1x_2 = x_1(-2v - x_1) = -2vx_1 - x_1^2$ , so

$$p_3 = x_1^3 + x_2^3 + 2v^3 = -6v^3 + 6vx_1x_2 = -6v(v^2 + 2vx_1 + x_1^2) = -6v(v + x_1)^2 \leq 0 < 6,$$

since  $v \geq 0$  and  $(v + x_1)^2 \geq 0$ . Hence no point on this wall satisfies  $p_3 \geq 6$ , so  $s_3 \notin G(S_c)$ , and

$$G(S_c) = \langle s_1, s_2 \rangle \subsetneq \mathfrak{S}_4.$$

We now trace Algorithm 6 on three pairs of points.

**Case 1 (both points in the Weyl chamber).** Let  $x = A = (-1, -1, -1, 3)$  and  $y = B = (-2, -1, \frac{1}{2}, \frac{5}{2})$ . Since both points lie in  $W_c$ , Algorithm 1 (Minimal\_Adjacent\_Transpositions) applied to  $y$  returns  $y' = B$  and  $T = \emptyset$ .

Algorithm 6, Step (2), calls Algorithm 4 (Connectivity\_Symmetric\_Canonical) on  $(x, y') = (A, B)$ . Since  $A$  and  $B$  have different Vandermonde images  $\nu_{4,3}(A) = (0, 12, 24)$  and  $\nu_{4,3}(B) = (0, \frac{23}{2}, \frac{27}{4})$ , Algorithm 4 must do genuine work as follows.

Steps (1)–(4) build the bipartite graph  $G$  encoding the connected components of the orbit boundary  $S_{4,3}$ , which lives on the face  $W_c^{(1,2,1)}$ .

Step (5) computes the Vandermonde images  $a = \nu_{4,3}(A) = (0, 12, 24)$  and  $b = \nu_{4,3}(B) = (0, \frac{23}{2}, \frac{27}{4})$ .

*Step (6)* calls `Min_Canonical` (Algorithm 3) to retract each point to the orbit boundary. For  $A$ , the minimizer of  $p_4$  on  $V(a) \cap W_c$  is  $x' = A$  itself (since  $A$  already lies on the  $(1, 2, 1)$  face with  $x_1 = x_2 = x_3$ ). For  $B$ , which lies in the interior of  $W_c$ , the minimizer of  $p_4$  on  $V(b) \cap W_c$  yields a point  $y'$  on the face  $(1, 2, 1)$ . By Lemma 5.8,  $B$  and  $y'$  are connected (they lie on the same Vandermonde fiber, which is contractible).

*Steps (7)–(9)* check whether  $x'$  and  $y'$  belong to the same connected component of the bipartite graph  $G$ . Since  $S_{4,3}$  is connected, this check succeeds.

Algorithm 4 returns `TRUE`. Since  $T = \emptyset$ , Step (4) of Algorithm 6 is vacuous, and the algorithm returns `TRUE`.

**Case 2 (different Weyl chambers, but connected).** Let  $x = A = (-1, -1, -1, 3)$  and  $y = z = (-2, \frac{1}{2}, -1, \frac{5}{2})$ . The point  $z$  lies outside  $W_c$  (since  $x_2 = \frac{1}{2} > -1 = x_3$ ).

*Step 1.* Algorithm 1 applied to  $z$  swaps positions 2 and 3 (since  $\frac{1}{2} > -1$ ) and returns  $z' = (-2, -1, \frac{1}{2}, \frac{5}{2}) = B$  with  $T = [s_2]$ .

*Step (2).* Algorithm 4 is called on  $(x, z') = (A, B)$ . Since  $A \neq B$ , this is a non-trivial connectivity query: Algorithm 4 retracts both points to the orbit boundary via `Min_Canonical` and checks their graph connectivity in  $S_{4,3}$ , exactly as described in Case 1. It returns `TRUE`.

*Step (4).* The list  $T = [s_2]$  contains one transposition. Algorithm 6 calls Algorithm 5 (`Connected_Wall`) with input  $(F, A, 2)$ , which checks whether  $A$  is path-connected in  $S_c$  to the wall  $W_c^{(2,3)} = \{x_2 = x_3\}$ . Since  $A$  itself lies on this wall (with  $x_2 = x_3 = -1$ ), Algorithm 5 returns `TRUE`.

All transpositions in  $T$  pass the wall check, so Algorithm 6 returns `TRUE`: the points  $A$  and  $z$  are connected in  $S$ .

**Case 3 (orbit-connected, but not connected).** Let  $x = A = (-1, -1, -1, 3)$  and  $y = w = (-2, -1, \frac{5}{2}, \frac{1}{2})$ . Again  $w$  lies outside  $W_c$  (since  $x_3 = \frac{5}{2} > \frac{1}{2} = x_4$ ).

*Step 1.* Algorithm 1 applied to  $w$  swaps positions 3 and 4 (since  $\frac{5}{2} > \frac{1}{2}$ ) and returns  $w' = (-2, -1, \frac{1}{2}, \frac{5}{2}) = B$  with  $T = [s_3]$ .

*Step (2).* Algorithm 4 is called on  $(x, w') = (A, B)$  and returns `TRUE`, exactly as in Case 2.

*Step (4).* Now  $T = [s_3]$ . Algorithm 6 calls Algorithm 5 with input  $(F, A, 3)$ , which checks whether  $A$  is path-connected in  $S_c$  to the wall  $W_c^{(3,4)} = \{x_3 = x_4\}$ . Since  $S_c \cap W_c^{(3,4)} = \emptyset$  (proved in the wall analysis above), Algorithm 5 returns `FALSE`, and Algorithm 6 immediately returns `FALSE`: the points  $A$  and  $w$  are not connected in  $S$ .

Geometrically, any continuous path from  $w$  to  $A$  must pass through a point with  $x_3 = x_4$  (by the intermediate value theorem applied to  $t \mapsto x_3(t) - x_4(t)$ , which changes sign). At every such point  $p_3 \leq 0 < 6$ , so no path can stay inside  $S$ .

## 7. Conclusion and outlook

In this article, we have presented the first polynomial-time algorithm for deciding connectivity in a symmetric semi-algebraic set. The key geometric insight underlying our approach is a dimension reduction technique, which is feasible due to the low degree of the defining polynomials. Notably, this degree restriction can be bypassed if a priori knowledge of the decomposition of symmetric polynomials into power sum polynomials is available. Moreover, alternative families of symmetric polynomial generators can also

be useful. For instance, as demonstrated in [37], a special representation in elementary symmetric polynomials leads to analogous results. The methods developed here can be generalized to this setting with only minor modifications. However, at present, we are still constrained by the specific choice of representation. An important open question is whether a sub-exponential algorithm can be obtained even in cases where no “sparse” representation in power sums or elementary symmetric polynomials is available. In such situations, the recent algorithms proposed by [20] provide a promising direction. Furthermore, we are limited to symmetric semi-algebraic sets described by symmetric polynomials. While every symmetric semi-algebraic set can be described using symmetric polynomials, transforming a general description into a symmetric one may lead to an increase in polynomial degree. For a particular class of symmetric semi-algebraic sets not originally defined by symmetric polynomials, Safey El Din and the first author [36] established a result that could also be utilized for dimension reduction. It would be highly interesting to explore, in a broader context, the extent to which a polynomial-time algorithm for connectivity – and more generally, for topological queries – remains feasible in the setting of symmetric sets that are not explicitly given by symmetric polynomials. Finally, much of our theory extends naturally directly to Coxeter groups. In fact, Friedl, Sanyal, and the first author have shown in [21] that a slightly weaker analog of Timofte’s theorem holds for all finite reflection groups. Establishing a general framework for our polynomial-time algorithm in the context of all Coxeter groups remains an intriguing avenue for future research.

*Acknowledgements.* The authors thank Saugata Basu and Nicolai Vorobjov for their valuable comments on an earlier version of this work, which contributed to both the improvement of the results and the clarity of the presentation.

C. Riener and R. Schabert were supported by the Tromsø Research Foundation grant 17MATCR. A part of this work was completed when T. X. Vu was working at the Department of Mathematics and Statistics, UiT The Arctic University of Norway, Tromsø, Norway, and at the Institute for Algebra, Johannes Kepler University, Linz, Austria. During that time, she was partially supported by the ANR-FWF grant 10.55776/I6130.

## References

- [1] J. Acevedo, G. Blekherman, S. Debus, and C. Riener. The wonderful geometry of the Vandermonde map. *Foundations of Computational Mathematics*, pages 1–47, 2025.
- [2] V. I. Arnold. Hyperbolic polynomials and Vandermonde mappings. *Funktsional’nyi Analiz i ego Prilozheniya*, 20(2):52–53, 1986.
- [3] Karol B. On the imbedding of systems of compacta in simplicial complexes. *Fundamenta Mathematicae*, 35:217–234, 1948.
- [4] S. Basu, R. Pollack, and M.-F. Roy. Computing roadmaps of semi-algebraic sets on a variety. *Journal of the American Mathematical Society*, 13(1):55–82, 2000.
- [5] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry*. Algorithms and Computation in Mathematics. Springer-Verlag, second edition edition, 2006.

- [6] S. Basu and C. Riener. Efficient algorithms for computing the Euler-Poincaré characteristic of symmetric semi-algebraic sets. In *Ordered Algebraic Structures and Related Topics: International Conference on Ordered Algebraic Structures and Related Topics, October 12–16, 2015, Centre International de Rencontres Mathématiques (CIRM), Luminy, France*, volume 697, pages 53–81. American Mathematical Soc. Providence, Rhode Island, 2017.
- [7] S. Basu and C. Riener. On the equivariant Betti numbers of symmetric definable sets: vanishing, bounds and algorithms. *Selecta Mathematica*, 24:3241–3281, 2018.
- [8] S. Basu and C. Riener. Vandermonde varieties, mirrored spaces, and the cohomology of symmetric semi-algebraic sets. *Foundations of Computational Mathematics*, 22(5):1395–1462, 2022.
- [9] S. Basu and M.-F. Roy. Divide and conquer roadmap for algebraic sets. *Discrete & Computational Geometry*, 52:278–343, 2014.
- [10] S. Basu, M.-F. Roy, M. Safey El Din, and É. Schost. A baby step–giant step roadmap algorithm for general algebraic sets. *Foundations of Computational Mathematics*, 14:1117–1172, 2014.
- [11] Anders Björner. Topological methods. In Ronald L. Graham, Martin Grötschel, and László Lovász, editors, *Handbook of Combinatorics*, pages 1819–1872. North-Holland, Amsterdam, 1995.
- [12] M. Bläser and G. Jindal. On the complexity of symmetric polynomials. In *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, pages 47–1. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019.
- [13] L. Bröcker. On symmetric semialgebraic sets and orbit spaces. *Banach Center Publications*, 44(1):37–50, 1998.
- [14] J. Canny. *The complexity of robot motion planning*. MIT Press, 1987.
- [15] J. Canny. Computing roadmaps of general semi-algebraic sets. *The Computer Journal*, 36(5):504–514, 1993.
- [16] J. Canny, D. Yu Grigor’ev, and N. N. Vorobjov. Finding connected components of a semialgebraic set in subexponential time. *Applicable Algebra in Engineering, Communication and Computing*, 2(4):217–238, 1992.
- [17] J. Capco, M. Safey El Din, and J. Schicho. Robots, computer algebra and eight connected components. In *Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation*, pages 62–69, 2020.
- [18] J. Capco, M. Safey El Din, and J. Schicho. Positive dimensional parametric polynomial systems, connectivity queries and applications in robotics. *Journal of Symbolic Computation*, 115:320–345, 2023.
- [19] Michael W. Davis. *The Geometry and Topology of Coxeter Groups. (LMS-32)*. Princeton University Press, Princeton, 2008.
- [20] J.-C. Faugère, G. Labahn, M. Safey El Din, É. Schost, and T. X. Vu. Computing critical points for invariant algebraic systems. *Journal of Symbolic Computation*, 116:365–399, 2023.
- [21] T. Friedl, C. Riener, and R. Sanyal. Reflection groups, reflection arrangements, and invariant real varieties. *Proceedings of the American Mathematical Society*, 146(3):1031–1045, 2018.
- [22] P. Gaudry, É. Schost, and N. M. Thiéry. Evaluation properties of symmetric polynomials. *International Journal of Algebra and Computation*, 16(03):505–523, 2006.

- [23] A. B. Givental. Moments of random variables and the equivariant Morse lemma. *Russian Mathematical Surveys*, 42(2):275–276, 1987.
- [24] L. Gournay and J.-J. Risler. Construction of roadmaps in semi-algebraic sets. *Applicable Algebra in Engineering, Communication and Computing*, 4(4):239–252, 1993.
- [25] D. Yu Grigor’ev and N. Vorobjov. Counting connected components of a semialgebraic set in subexponential time. *Computational Complexity*, 2:133–186, 1992.
- [26] J. Heintz, M.-F. Roy, and P. Solernó. Single exponential path finding in semi-algebraic sets, part II: The general case. In *Algebraic Geometry and its Applications: Collections of Papers from Shreeram S. Abhyankar’s 60th Birthday Conference*, pages 449–465. Springer, 1994.
- [27] R. Irají and H. Chitsaz. Nuroa: A numerical roadmap algorithm. In *53rd IEEE Conference on Decision and Control*, pages 5359–5366. IEEE, 2014.
- [28] D. Jungnickel and D. Jungnickel. *Graphs, networks and algorithms*, volume 3. Springer, 2005.
- [29] V. P. Kostov. On the geometric properties of Vandermonde’s mapping and on the problem of moments. *Proceedings of the Royal Society of Edinburgh Section A: Mathematics*, 112(3-4):203–211, 1989.
- [30] G. Labahn, C. Riener, M. Safey El Din, É. Schost, and T. X. Vu. Faster real root decision algorithm for symmetric polynomials. In *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation*, pages 452–460, 2023.
- [31] I. Meguerditchian. A theorem on the escape from the space of hyperbolic polynomials. *Mathematische Zeitschrift*, 211:449–460, 1992.
- [32] R. Prébet, M. Safey El Din, and É. Schost. Computing roadmaps in unbounded smooth real algebraic sets I: connectivity results. *Journal of Symbolic Computation*, 120:102234, 2024.
- [33] R. Prébet, M. Safey El Din, and É. Schost. Computing roadmaps in unbounded smooth real algebraic sets II: algorithm and complexity. *Journal of Symbolic Computation*, page 102532, 2025.
- [34] C. Riener. On the degree and half-degree principle for symmetric polynomials. *Journal of Pure and Applied Algebra*, 216(4):850–856, 2012.
- [35] C. Riener. Symmetric semi-algebraic sets and non-negativity of symmetric polynomials. *Journal of Pure and Applied Algebra*, 220(8):2809–2815, 2016.
- [36] C. Riener and M. Safey El Din. Real root finding for equivariant semi-algebraic systems. In *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*, pages 335–342, 2018.
- [37] C. Riener and R. Schabert. Linear slices of hyperbolic polynomials and positivity of symmetric polynomial functions. *Journal of Pure and Applied Algebra*, 228(5):107552, 2024.
- [38] C. Riener, R. Schabert, and T. X. Vu. Connectivity in symmetric semi-algebraic sets. In *Proceedings of the 2024 International Symposium on Symbolic and Algebraic Computation*, pages 162–169, 2024.
- [39] C. Riener, T. Theobald, L. J. Andrén, and J.-B. Lasserre. Exploiting symmetries in SDP-relaxations for polynomial optimization. *Mathematics of Operations Research*, 38(1):122–141, 2013.
- [40] F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *Applicable Algebra in Engineering, Communication and Computing*, 9(5):433–461, 1999.

- [41] M. Safey El Din and É. Schost. A baby steps/giant steps probabilistic algorithm for computing roadmaps in smooth bounded real hypersurface. *Discrete & Computational Geometry*, 45(1):181–220, 2011.
- [42] M. Safey El Din and É. Schost. A nearly optimal algorithm for deciding connectivity queries in smooth and bounded real algebraic sets. *Journal of the ACM (JACM)*, 63(6):1–37, 2017.
- [43] J. T. Schwartz and M. Sharir. On the “piano movers” problem. II. general techniques for computing topological properties of real algebraic manifolds. *Advances in applied Mathematics*, 4(3):298–351, 1983.
- [44] L. Solomon. A decomposition of the group algebra of a finite Coxeter group. *Journal of Algebra*, 9:220–239, 1968.
- [45] V. Timofte. On the positivity of symmetric polynomial functions. part I: General results. *Journal of Mathematical Analysis and Applications*, 284(1):174–190, 2003.
- [46] T. X Vu. Computing polynomial representation in subrings of multivariate polynomial rings. In *Proceedings of the 2025 International Symposium on Symbolic and Algebraic Computation*, pages 284–292, 2025.