

Nonlinear Craig Interpolant Generation Over Unbounded Domains by Separating Semialgebraic Sets

Hao Wu[1], Jie Wang[2], Bican Xia[3], Xiakun Li[3], Naijun Zhan[1,4], and Ting Gan[5(✉)]





[1] SKLCS, Institute of Software, University of CAS, Beijing, China
{znj,wuhao}@ios.ac.cn
[2] Academy of Mathematics and Systems Science, CAS, Beijing, China
wangjie212@amss.ac.cn
[3] School of Mathematical Sciences, Peking University, Beijing, China
xbc@math.pku.edu.cn, 2301110072@stu.pku.edu.cn
[4] School of Computer Science, Peking University, Beijing, China
[5] School of Computer Science, Wuhan University, Wuhan, China
ganting@whu.edu.cn

Abstract. Interpolation-based techniques become popular in recent years, as they can improve the scalability of existing verification techniques due to their inherent modularity and local reasoning capabilities. Synthesizing Craig interpolants is the cornerstone of these techniques. In this paper, we investigate nonlinear Craig interpolant synthesis for two polynomial formulas of the general form, essentially corresponding to the underlying mathematical problem to separate two disjoint semialgebraic sets. By combining the homogenization approach with existing techniques, we prove the existence of a novel class of non-polynomial interpolants called semialgebraic interpolants. These semialgebraic interpolants subsume polynomial interpolants as a special case. To the best of our knowledge, this is the first existence result of this kind. Furthermore, we provide complete sum-of-squares characterizations for both polynomial and semialgebraic interpolants, which can be efficiently solved as semidefinite programs. Examples are provided to demonstrate the effectiveness and efficiency of our approach.

Keywords: Craig interpolation · Separating semialgebraic sets · Homogenization · Sum-of-squares · Semidefinite programming

H. Wu and J. Wang—The first two authors contributed equally to this work and should be considered co-first authors. This work has been partially funded by the National Key R&D Program of China under grant No. 2022YFA1005101 and 2022YFA1005102, by the NSFC under grant No. 62192732, 62032024, and 12201618, by the CAS Project for Young Scientists in Basic Research under grant No. YSBR-040, by the Key R&D Program of Hubei Province (2023BAB170), and by the Fundamental Research Funds for the Central Universities.

© The Author(s) 2025
A. Platzer et al. (Eds.): FM 2024, LNCS 14933, pp. 92–110, 2025.
https://doi.org/10.1007/978-3-031-71162-6_5

1 Introduction

Background. Craig interpolant is a fundamental concept in formal verification and automated theorem proving. It was introduced by William Craig in the 1950 s as a tool for reasoning about logical formulas and their satisfiability. Craig interpolation techniques possess excellent modularity and local reasoning capabilities, making them effective tools for enhancing the scalability of formal verification methods, like theorem proving [23,35], model-checking [31], abstract interpretation [14,32], program verification [19,26] and so on.

Efficient generation of Craig interpolants is crucial in interpolation-based techniques, and therefore has garnered increasing attention. Formally, a formula I is called a Craig interpolant for two mutually exclusive formulae ϕ and ψ in a background theory $\mathcal{T}$, if it is defined on the common symbols of ϕ and ψ, implied by ϕ in the theory $\mathcal{T}$, and inconsistent with ψ in the theory $\mathcal{T}$. Due to the diversity of background theories and their integration, researchers have been dedicated to developing efficient interpolation synthesis algorithms. Currently, numerous effective algorithms for automatic synthesis of interpolants have been proposed for various fragments of first-order logic, e.g., linear arithmetic [14], logic with arrays [15,33], logic with sets [20], equality logic with uninterpreted functions (EUF) [5,32], etc., and their combinations [22,38,43]. Moreover, D'Silva et al. [9] explored how to compare the strength of various interpolants.

However, interpolant generation for nonlinear arithmetic and its combination with the aforementioned theories is still in infancy, although nonlinear polynomial inequalities are quite common in software involving number theoretic functions as well as hybrid systems [44,45]. In addition, when the formulas ϕ and ψ are defined by polynomial inequalities, generating an interpolant is essentially equivalent to the mathematical problem of separating two disjoint semialgebraic sets, which has a long history and is a challenging mathematical problem [1].

In [7], Dai et al. attempted to generate interpolants for conjunctions of mutually contradictory nonlinear polynomial inequalities without unshared variables. They proposed an algorithm based on Stengle's Positivstellensatz [41], which guarantees the existence of a witness and can be computed using semidefinite programming (SDP). While their algorithm is generally incomplete, it becomes complete when all variables are bounded, known as the Archimedean condition (see in Sect. 2.1).

In [10], Gan et al. introduced an algorithm for generating interpolants specifically for quadratic polynomial inequalities. Their approach is based on the insight that analyzing the solution space of concave quadratic polynomial inequalities can be achieved by linearizing them, using a generalization of Motzkin's transposition theorem. Additionally, they discussed generating interpolants for a combination of the theory of quadratic concave polynomial inequalities and EUF using a hierarchical calculus proposed in [39] and employed in [38].

In [11], Gan et al. further extended the problem from the case of quadratic concave inequalities to the more general Archimedean case. To accomplish this, they utilized Putinar's Positivstellensatz and proposed a Craig interpolation generation method based on SDP. This method allows to generate interpolants in

a broader class of situations involving nonlinear polynomial inequalities. However, the Archimedean condition still imposes a limitation on the method, as it requires bounded domains.

In [4], Chen et al. proposed a counterexample-guided framework based on support vector machines for synthesizing nonlinear interpolants. Later in [27], Lin et al. combined this framework and deep learning for synthesizing neural interpolants. In [18], Jovanović and Dutertre also designed a counterexample-guided framework based on cylindrical algebraic decomposition (CAD) for synthesizing interpolants as boolean combinations of constraints. However, these approaches rely on quantifier elimination to ensure completeness and convergence, which terribly affects their efficiency due to its doubly exponential time complexity [8].

For theories including non-polynomial expressions, the general idea is to abstract non-polynomial expressions into polynomial or linear expressions. In [13], Gao and Zufferey presented an approach for extracting interpolants for nonlinear formulas that may contain transcendental functions and differential equations. They accomplished this by transforming proof traces from a δ-decision procedure [12] based on interval constraint propagation (ICP) [3]. Like the Archimedean condition, δ-decidability also imposes the restriction that variables are bounded (in a hyper-rectangle). A similar idea was also reported in [24]. In [6,40], Srikanth et al. and Cimatti et al. proposed approaches to abstract nonlinear formulas into the theory of linear arithmetic with uninterpreted functions.

Contributions. In this paper, we consider how to synthesize an interpolant function $h(\mathbf{x})$ for two polynomial formulas $\phi(\mathbf{x}, \mathbf{y})$ and $\psi(\mathbf{x}, \mathbf{z})$ such that $\phi(\mathbf{x}, \mathbf{y}) \models h(\mathbf{x}) > 0$ and $\psi(\mathbf{x}, \mathbf{z}) \models h(\mathbf{x}) < 0$ without assuming the Archimedean condition, i.e., the variables in ϕ and ψ can have an unbounded range of values. Here, uncommon variables of ϕ and ψ are allowed, and the description of formulas may involve any polynomial of any degree. Hence the problem is more general than the ones discussed in [7,10,11], and is also more difficult as polynomial interpolants may not exist [1]. To address this problem, we first utilize homogenization techniques to elevate the descriptions of ϕ and ψ to the homogeneous space. In this homogeneous space, we can impose the constraint that the variables lie on a unit sphere, thus reviving the Archimedean condition. Combining this idea with the work in [11], we can prove the existence of a semialgebraic function $h(\mathbf{x}) = h_1(\mathbf{x}) + h_2(\mathbf{x})\sqrt{\|\mathbf{x}\|^2 + 1}$ such that $h(\mathbf{x}) > 0$ serves as an interpolant, where h_1, h_2 are polynomials (h becomes a polynomial when $h_2 = 0$). Furthermore, we provide sum-of-squares (SOS) programming procedures for finding such semialgebraic interpolants as well as polynomial interpolants. Under certain assumptions, we prove that the SOS procedures are sound and complete.

Organization. The rest of the paper is organized as follows. Preliminaries are introduced in Sect. 2. Section 3 proves the existence of an interpolant for two mutually contradictory polynomial formulas. Section 4 derives an SOS characterization for the interpolant. Section 5 presents an SDP-based method for computation and provides examples. Finally, Sect. 6, we conclude this paper and

discuss some future works. Omitted proofs and portraits for examples can be found in the extended version [42].

2 Preliminaries

We first fix some basic notations. Let $\mathbb{N}$ and $\mathbb{R}$ be the sets of integers and real numbers, respectively. By convention, we use boldface letters to denote vectors. Fixing a vector of indeterminates $\mathbf{x} \coloneqq (x_1, \ldots, x_r)$, let $\mathbb{R}[\mathbf{x}]$ denote the polynomial ring in variables $\mathbf{x}$ over real numbers. We use $\Sigma[\mathbf{x}] \coloneqq \{\sum_{i=1}^{m} p_i^2 \mid p_i \in \mathbb{R}[\mathbf{x}], m \in \mathbb{N}\}$ to denote the set of SOS polynomials in variables $\mathbf{x}$. A basic semialgebraic set $S \subseteq \mathbb{R}^r$ is of the form $\{\mathbf{x} \in \mathbb{R}^r \mid p_1(\mathbf{x}) \triangleright 0, \ldots, p_m(\mathbf{x}) \triangleright 0\}$, where $p_i(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ and $\triangleright \in \{\geq, >\}$ (each of the inequalities can be either strict or non-strict). A basic semialgebraic set is said to be *closed* if it is defined by non-strict inequalities. Semialgebraic sets are formed as unions of basic semialgebraic sets. i.e., $T = \bigcup_{i=1}^{n} S_i$ is a semialgebraic set, where each S_i is a basic semialgebraic set. For any (semialgebraic) set $S \subseteq \mathbb{R}^r$, let cl(S) denote the closure of S. Let $\bot$ and $\top$ stand for **false** and **true**, respectively. For a vector $\mathbf{x} \in \mathbb{R}^r$, let $\|\mathbf{x}\| \coloneqq \sqrt{\sum_{i=1}^{r} x_i^2}$ denote the standard Euclidean norm.

In the following, we give a brief introduction on important notions used throughout the rest of this paper and then describe the problem of interest.

2.1 Quadratic Module

Definition 1 (Quadratic Module [30]). *A subset $\mathcal{M}$ of $\mathbb{R}[\mathbf{x}]$ is called a* quadratic module *if it contains 1 and is closed under addition and multiplication with squares, i.e.,*

$$1 \in \mathcal{M}, \mathcal{M} + \mathcal{M} \subseteq \mathcal{M}, \text{ and } p^2\mathcal{M} \subseteq \mathcal{M} \text{ for all } p \in \mathbb{R}[\mathbf{x}].$$

Definition 2. *Let $\overline{p} \coloneqq \{p_1, \ldots, p_m\}$ be a finite subset of $\mathbb{R}[\mathbf{x}]$. The quadratic module $\mathcal{M}_{\mathbf{x}}(\overline{p})$, or simply $\mathcal{M}(\overline{p})$, generated by $\overline{p}$ is the smallest quadratic module containing all p_i, i.e.,*

$$\mathcal{M}_{\mathbf{x}}(\overline{p}) \coloneqq \{\sigma_0 + \sum_{i=1}^{m} \sigma_i p_i \mid \sigma_0, \sigma_i \in \Sigma[\mathbf{x}]\}.$$

Let S be a closed basic semialgebraic set described by $\overline{p} \geq \mathbf{0}$, i.e.,

$$S \coloneqq \{\mathbf{x} \in \mathbb{R}^r \mid p_1(\mathbf{x}) \geq 0, \ldots, p_m(\mathbf{x}) \geq 0\}. \tag{1}$$

Since SOS polynomials are non-negative, it is easy to verify that the quadratic module $\mathcal{M}(\overline{p})$ is a subset of polynomials that are nonnegative on S. In fact, under the so-called the Archimedean condition, the quadratic module $\mathcal{M}(\overline{p})$ contains all polynomials that are strictly positive over S. Both the condition and the statement are formalized as follows.

Definition 3 (Archimedean [30]). *Let $\mathcal{M}$ be a quadratic module of $\mathbb{R}[\mathbf{x}]$. $\mathcal{M}$ is said to be* Archimedean *if there exists some $a > 0$ such that $a - \|\mathbf{x}\|^2 \in \mathcal{M}$. Furthermore, if $\mathcal{M}(\overline{p})$ is Archimedean, we say that the semialgebraic set S as defined in Eq. (1) is of the* Archimedean form.

Theorem 1 (Putinar's Positivstellensatz [36]). *Let $\overline{p} \coloneqq \{p_1, \ldots, p_m\}$ and S be defined in Eq. (1). Assume that the quadratic module $\mathcal{M}(\overline{p})$ is Archimedean. If $f(\mathbf{x}) > 0$ over S, then $f \in \mathcal{M}(\overline{p})$.*

The above theorem serves as a key result in real algebraic geometry, as it provides a simple characterization of polynomials that are locally positive on closed basic semialgebraic sets. Because of this, Theorem 1 is widely used in the field of polynomial optimization, referring to [25,30] for an in-depth treatment of this topic.

Though powerful, Theorem 1 relies on the Archimedean condition. Note that the inclusion $a - \|\mathbf{x}\|^2 \in \mathcal{M}$ implies that $a - \|\mathbf{x}\|^2 \geq 0$ over S, deducing that S is contained in a ball with radius $\sqrt{a}$. As a result, in case that the set S is unbounded, Theorem 1 is not directly applicable.

2.2 Homogenization

Let $\mathbf{x} = (x_1, \ldots, x_r) \in \mathbb{R}^r$ be an r-tuple of variables and x_0 a fresh variable. Suppose that $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ is a polynomial of degree d_f. We denote by $\tilde{f}(x_0, \mathbf{x}) \in \mathbb{R}[x_0, \mathbf{x}]$ the homogenization of $f(\mathbf{x})$ which is obtained by substituting $\frac{x_1}{x_0}$ for x_1, ..., $\frac{x_r}{x_0}$ for x_r in $f(\mathbf{x})$ and then multiplying with $x_0^{d_f}$, that is,

$$\tilde{f}(x_0, \mathbf{x}) \coloneqq x_0^{d_f} f(\frac{x_1}{x_0}, \ldots, \frac{x_r}{x_0}). \tag{2}$$

For example, if $f(\mathbf{x}) = x_1^3 + 2x_1x_2 + 3x_2 + 4$, then $\tilde{f}(x_0, \mathbf{x}) = x_1^3 + 2x_0x_1x_2 + 3x_0^2x_2 + 4x_0^3$. In what follows, we always use the variable x_0 as the homogenizing variable.

Let S be defined as in Eq. (1). We define the following set related to S by homogenizing polynomials in the description of S:

$$\tilde{S}^h \coloneqq \{(x_0, \mathbf{x}) \in \mathbb{R}^{r+1} \mid \tilde{p}_1(x_0, \mathbf{x}) \geq 0, \ldots, \tilde{p}_m(x_0, \mathbf{x}) \geq 0, x_0 > 0, x_0^2 + \|\mathbf{x}\|^2 = 1\}. \tag{3}$$

Obviously, the following property holds.

Property 1. *Let S be as in Eq. (1) and $\tilde{S}^h$ be defined as above. Then, $\mathbf{x} \in S$ if and only if*

$$\left(\frac{1}{\sqrt{1 + \|\mathbf{x}\|^2}}, \frac{x_1}{\sqrt{1 + \|\mathbf{x}\|^2}}, \ldots, \frac{x_r}{\sqrt{1 + \|\mathbf{x}\|^2}}\right) \in \tilde{S}^h.$$

Moreover, $(x_0, \mathbf{x}) \in \tilde{S}^h$ if and only if $(\frac{x_1}{\sqrt{1-\|\mathbf{x}\|^2}}, \ldots, \frac{x_r}{\sqrt{1-\|\mathbf{x}\|^2}}) \in S$.

Property 1 shows that there exists a one-to-one correspondence between points in $S \in \mathbb{R}^n$ and those in $\tilde{S}^h \in \mathbb{R}^{n+1}$.

We also define the set $\tilde{S}$ by replacing $x_0 > 0$ in Eq. (3) with $x_0 \geq 0$:

$$\tilde{S} \coloneqq \{(x_0, \mathbf{x}) \in \mathbb{R}^{r+1} \mid \tilde{p}_1(x_0, \mathbf{x}) \geq 0, \ldots, \tilde{p}_m(x_0, \mathbf{x}) \geq 0, x_0 \geq 0, x_0^2 + \|\mathbf{x}\|^2 = 1\}. \tag{4}$$

To capture the relation between $\tilde{S}^h$ and $\tilde{S}$, we introduce the following definition and a related useful lemma.

Definition 4. *A closed basic semialgebraic set S is closed at ∞ if* $\mathrm{cl}(\tilde{\mathrm{S}}^{\mathrm{h}}) = \tilde{\mathrm{S}}$.

Lemma 1 ([17]). *Let $f \in \mathbb{R}[\mathbf{x}]$ and S be a closed basic semialgebraic set. Then $f \geq 0$ on S if and only if $\tilde{f} \geq 0$ on* $\mathrm{cl}(\tilde{\mathrm{S}}^{\mathrm{h}})$. *Moreover, assuming that S is closed at ∞, then $f \geq 0$ on S if and only if $\tilde{f} \geq 0$ on $\tilde{S}$.*

Let us define

$$S^{(\infty)} \coloneqq \{\mathbf{x} \in \mathbb{R}^r \mid p_1^{(\infty)}(\mathbf{x}) \geq 0, \ldots, p_m^{(\infty)}(\mathbf{x}) \geq 0, \|\mathbf{x}\|^2 = 1\}, \tag{5}$$

where $p^{(\infty)}(\mathbf{x})$ denotes the highest degree homogeneous part of a polynomial $p(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$, e.g., if $p = x_1^2 + 2x_1x_2 + 3x_2^2 + 4x_1 + 5x_2$, then $p^{(\infty)} = x_1^2 + 2x_1x_2 + 3x_2^2$.

Property 2. *Let $\tilde{S}^h$, $\tilde{S}$ and $S^{(\infty)}$ be defined as above. If $S^{(\infty)}$ is empty, then $\tilde{S}^h = \tilde{S}$.*

2.3 Problem Description

Given two formulas ϕ and ψ in a first-order theory $\mathcal{T}$ s.t. $\phi \models \psi$, Craig showed that there always exists an *interpolant* I over the common symbols of ϕ and ψ s.t. $\phi \models I$ and $I \models \psi$. In the context of verification, we slightly abuse the terminology following [32]: A *reverse interpolant* (as coined in [22]) I over the common symbols of ϕ and ψ is defined as follows.

Definition 5 (Interpolant). *Given two formulas ϕ and ψ in a theory $\mathcal{T}$ s.t. $\phi \wedge \psi \models_{\mathcal{T}} \bot$, a formula I is an* interpolant *of ϕ and ψ if (1) $\phi \models_{\mathcal{T}} I$; (2) $I \wedge \psi \models_{\mathcal{T}} \bot$; and (3) I only contains common symbols and free variables shared by ϕ and ψ.*

The interpolant synthesis problem of interest in this paper is formulated as follows.

Problem 1. Let $\phi(\mathbf{x}, \mathbf{y})$ and $\psi(\mathbf{x}, \mathbf{z})$ be two polynomial formulas of the form

$$\phi(\mathbf{x}, \mathbf{y}) \coloneqq \bigvee_{k=1}^{K_\phi} \bigwedge_{i=1}^{m_k} f_{k,i}(\mathbf{x}, \mathbf{y}) \geq 0, \tag{6}$$

$$\psi(\mathbf{x}, \mathbf{z}) \coloneqq \bigvee_{k'=1}^{K_\psi} \bigwedge_{j=1}^{n_{k'}} g_{k',j}(\mathbf{x}, \mathbf{z}) \geq 0, \tag{7}$$

where $\mathbf{x} \in \mathbb{R}^{r_1}$, $\mathbf{y} \in \mathbb{R}^{r_2}$, $\mathbf{z} \in \mathbb{R}^{r_3}$ are variable vectors, $r_1, r_2, r_3 \in \mathbb{N}$, and $f_{k,i}, g_{k',j}$ are polynomials in the corresponding variables. We aim to find a function $h(\mathbf{x})$ such that $h(\mathbf{x}) > 0$ is an interpolant for ϕ and ψ, i.e.,

$$\phi(\mathbf{x}, \mathbf{y}) \models h(\mathbf{x}) > 0 \text{ and } \psi(\mathbf{x}, \mathbf{z}) \models h(\mathbf{x}) < 0.$$

Here $h(\mathbf{x})$ is called an interpolant function. Specifically, we are interested in two scenarios where

1. **Polynomial interpolants**: the function $h(\mathbf{x})$ is a polynomial in $\mathbb{R}[\mathbf{x}]$;
2. **Semialgebraic interpolants**[1]: the function $h(\mathbf{x})$ can be expressed as

$$h(\mathbf{x}) = h_1(\mathbf{x}) + \sqrt{\|\mathbf{x}\|^2 + 1} \cdot h_2(\mathbf{x}), \tag{8}$$

with $h_1(\mathbf{x}), h_2(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$.

Obviously, the second case degenerates to the first case when $h_2(\mathbf{x}) = 0$.

Remark 1. Like in [11,12], we require ϕ and ψ to be defined by *non-strict* polynomial inequalities, mainly for two reasons: (1) Theoretically, our approach relies on Theorem 1, which necessitates a closed underlying basic semialgebraic set. (2) Numerically, we employ numerical solvers incapable of distinguishing $\geq$ from $>$. In the coming sections, we will see the significance of both closedness and closedness at ∞ for the existence of interpolants.

3 Existence of Interpolant

In this section, we prove the existence of a semialgebraic interpolant function $h(\mathbf{x})$ of the form Eq. (8), under certain conditions on ϕ and ψ. In Sect. 3.1, we begin by focusing on the scenario where both ϕ and ψ exclusively involve the variable $\mathbf{x}$. Subsequently, in Sect. 3.2, we expand our scope to the case where unshared variables, $\mathbf{y}$ and $\mathbf{z}$, emerge.

3.1 Interpolant Between $\phi(\mathbf{x})$ and $\psi(\mathbf{x})$

In this part, we prove the existence of a semialgebraic interpolant function of the form in Eq. (8) that separates the two closed semialgebraic sets in $\mathbb{R}^r$ corresponding to $\phi(\mathbf{x})$ and $\psi(\mathbf{x})$. The basic idea goes as follows: First, we consider the problem of finding a semialgebraic function $h(\mathbf{x})$ such that $h(\mathbf{x}) = 0$ separates two closed basic semialgebraic sets S_1 and S_2 in $\mathbb{R}^r$. Using the homogenization technique, we prove that there exists a polynomial $g \in \mathbb{R}[x_0, \mathbf{x}]$ with $g(x_0, \mathbf{x}) = 0$ separating $\tilde{S_1}$ and $\tilde{S_2}$, and the existence of $h(\mathbf{x})$ is directly induced by that of g (see Proposition 2). After that, we extend the result to the case where S_1 becomes a closed semialgebraic set (see Lemma 3) and when both S_1 and S_2 are closed semialgebraic sets (see Theorem 2).

We begin by recapping an existing result from [11].

[1] A function $f(\mathbf{x})$ is called semialgebraic if its graph $\{(\mathbf{x}, f(\mathbf{x})) \mid \mathbf{x} \in \mathbb{R}^r\}$ is a semialgebraic set. The graph of $h(\mathbf{x})$ is $\{\mathbf{x} \in \mathbb{R}^r \mid \exists w.\ h(\mathbf{x}) = h_1(\mathbf{x}) + w \cdot h_2(\mathbf{x}) \wedge w^2 = 1 + \|\mathbf{x}\|^2 \wedge w \geq 0\}$.

Proposition 1 ([11, Lemma 2]). *Let* $S_1 = \{\mathbf{x} \in \mathbb{R}^r \mid p_1(\mathbf{x}) \geq 0, \ldots, p_m(\mathbf{x}) \geq 0\}$, $S_2 = \{\mathbf{x} \in \mathbb{R}^r \mid q_1(\mathbf{x}) \geq 0, \ldots, q_n(\mathbf{x}) \geq 0\}$ *be two closed basic semialgebraic sets of the Archimedean form. Assuming that* $S_1 \cap S_2 = \emptyset$, *then there exists a polynomial* $h(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ *such that*

$$\forall \mathbf{x} \in S_1.\ h(\mathbf{x}) > 0 \text{ and } \forall \mathbf{x} \in S_2.\ -h(\mathbf{x}) > 0. \tag{9}$$

It is important to emphasize that the proof of Proposition 1 relies on Theorem 1 and hence is limited to the case where the sets S_1 and S_2 are of the Archimedean form. In the following Proposition 2, we show how to remove this restriction.

Proposition 2. *Let* $S_1 = \{\mathbf{x} \in \mathbb{R}^r \mid p_1(\mathbf{x}) \geq 0, \ldots, p_m(\mathbf{x}) \geq 0\}$, $S_2 = \{\mathbf{x} \in \mathbb{R}^r \mid q_1(\mathbf{x}) \geq 0, \ldots, q_n(\mathbf{x}) \geq 0\}$ *be closed basic semialgebraic sets. Assuming that* $\tilde{S}_1 \cap \tilde{S}_2 = \emptyset$, *then there exists a semialgebraic function* $h(\mathbf{x})$ *of the form in Eq. (8) such that*

$$\forall \mathbf{x} \in S_1.\ h(\mathbf{x}) > 0 \text{ and } \forall \mathbf{x} \in S_2.\ -h(\mathbf{x}) > 0. \tag{10}$$

Proof. By the definition of $\tilde{S}$ in Eq. (4), we know that $\tilde{S}_1$ and $\tilde{S}_2$ are two basic semialgebraic sets of the Archimedean form (as $1 - x_0^2 - \|\mathbf{x}\|^2$ belongs to the corresponding quadratic modules). Since $\tilde{S}_1 \cap \tilde{S}_2 = \emptyset$, by invoking Proposition 1, there exists a polynomial $g \in \mathbb{R}[x_0, \mathbf{x}]$ such that

$$\forall (x_0, \mathbf{x}) \in \tilde{S}_1.\ g(x_0, \mathbf{x}) > 0 \text{ and } \forall (x_0, \mathbf{x}) \in \tilde{S}_2.\ -g(x_0, \mathbf{x}) > 0. \tag{11}$$

Note that for any $\mathbf{x} \in S_1$ (resp. S_2), by Property 1 we have $(\frac{1}{\sqrt{\|\mathbf{x}\|^2+1}}, \frac{\mathbf{x}}{\sqrt{\|\mathbf{x}\|^2+1}}) \in \tilde{S}_1$ (resp. $\tilde{S}_2$). Let

$$h(\mathbf{x}) := (\sqrt{\|\mathbf{x}\|^2 + 1})^{\deg(g)} g(\frac{1}{\sqrt{\|\mathbf{x}\|^2 + 1}}, \frac{\mathbf{x}}{\sqrt{\|\mathbf{x}\|^2 + 1}}). \tag{12}$$

Since $(\sqrt{\|\mathbf{x}\|^2 + 1})^{\deg(g)} \geq 1$, combining with Eq. (11), we have that $h(\mathbf{x})$ satisfies Eq. (10).

To see that $h(\mathbf{x})$ admits the form in Eq. (8), we expand the right-hand side of Eq. (12) and simplify the terms with power of $\sqrt{\|\mathbf{x}\|^2 + 1}$ greater than or equal to 2. After simplification, we collect the terms with and without $\sqrt{\|\mathbf{x}\|^2 + 1}$ into two groups so that $h(\mathbf{x})$ can be expressed as $h_1(\mathbf{x}) + \sqrt{\|\mathbf{x}\|^2 + 1} \cdot h_2(\mathbf{x})$ for $h_1(\mathbf{x}), h_2(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$. □

In order to check whether the condition $\tilde{S}_1 \cap \tilde{S}_2 = \emptyset$ in Proposition 2 holds, one can use the following lemma.

Lemma 2. *Given two closed basic semialgebraic set* S_1 *and* S_2, *if* $S_1 \cap S_2 = \emptyset$ *and* $S_1^{(\infty)} \cap S_2^{(\infty)} = \emptyset$, *then* $\tilde{S}_1 \cap \tilde{S}_2 = \emptyset$.

Now, we extend the result in Proposition 2 to the case when S_1 and S_2 are two closed semialgebraic sets. A closed semialgebraic set, say T, is a union of some closed basic semialgebraic sets, i.e., $T = \cup_{i=1}^{a} S_i$ with

$$S_i = \{\mathbf{x} \in \mathbb{R}^r \mid p_{i1}(\mathbf{x}) \geq 0, \ldots, p_{im_i}(\mathbf{x}) \geq 0\}, \quad i = 1, \ldots, a,$$

where $p_{ik}(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$, $m_i \in \mathbb{N}$, $k = 1, ..., m_i$, $i = 1, \ldots, a$. Mirroring the definition of $S^{(\infty)}$ and $\tilde{S}$, we define $T^{(\infty)} \coloneqq \bigcup_{i=1}^{a} S_i^{(\infty)}$ and $\tilde{T} \coloneqq \bigcup_{i=1}^{a} \tilde{S}_i$. In the following lemma, we deal with the case when S_1 in Proposition 2 becomes a union of closed basic semialgebraic sets.

Lemma 3. *Let $T_1 = \cup_{i=1}^{a} S_i$ be a closed semialgebraic set with $S_i = \{\mathbf{x} \in \mathbb{R}^r \mid p_{i1}(\mathbf{x}) \geq 0, \ldots, p_{im_i}(\mathbf{x}) \geq 0\}$, and let $T_2 = \{\mathbf{x} \in \mathbb{R}^r \mid q_1(\mathbf{x}) \geq 0, \ldots, q_n(\mathbf{x}) \geq 0\}$ be a closed basic semialgebraic set. Assume that $\tilde{T}_1 \cap \tilde{T}_2 = \emptyset$. Then there exists a polynomial $g \in \mathbb{R}[x_0, \mathbf{x}]$ such that*

$$\forall (x_0, \mathbf{x}) \in \tilde{T}_1.\ g(x_0, \mathbf{x}) > 0 \text{ and } \forall (x_0, \mathbf{x}) \in \tilde{T}_2.\ -g(x_0, \mathbf{x}) > 0. \tag{13}$$

Then, we use Lemma 3 to prove the case where both sets are unions of closed basic semialgebraic sets.

Theorem 2. *Let $T_1 = \cup_{i=1}^{a} S_i$ and $T_2 = \cup_{j=1}^{b} S'_j$ be closed semialgebraic sets, where S_i and S'_j are closed basic semialgebraic sets for $i = 1, \ldots, a, j = 1, \ldots, b$. Assume $\tilde{T}_1 \cap \tilde{T}_2 = \emptyset$. Then there exists a semialgebraic function $h(\mathbf{x})$ of the form in Eq. (8) such that*

$$\forall \mathbf{x} \in T_1.\ h(\mathbf{x}) > 0 \text{ and } \forall \mathbf{x} \in T_2.\ -h(\mathbf{x}) > 0. \tag{14}$$

Similarly to Lemma 2, the condition $\tilde{T}_1 \cap \tilde{T}_2 = \emptyset$ can be verified by checking whether $T_1 \cap T_2 = \emptyset$ and $T_1^{(\infty)} \cap T_2^{(\infty)} = \emptyset$. As a direct inference of Theorem 2, we know that there exists a semialgebraic function $h(\mathbf{x})$ of the form in Eq. (8) such that $h(\mathbf{x}) > 0$ is an interpolant of $\phi(\mathbf{x})$ and $\psi(\mathbf{x})$.

3.2 Interpolant Between $\phi(\mathbf{x}, \mathbf{y})$ and $\psi(\mathbf{x}, \mathbf{z})$

Let $\phi(\mathbf{x}, \mathbf{y})$ and $\psi(\mathbf{x}, \mathbf{z})$ be given in Problem 1. We denote by $T_\phi \subseteq \mathbb{R}^{r_1+r_2}$ and $T_\psi \subseteq \mathbb{R}^{r_1+r_3}$ the semialgebraic sets corresponding to ϕ and ψ, i.e.,

$$T_\phi \coloneqq \bigcup_{k=1}^{K_\phi} S_k, \text{ with } S_k \coloneqq \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{r_1+r_2} \mid \bigwedge_{i=1}^{m_k} f_{k,i}(\mathbf{x}, \mathbf{y}) \geq 0\}, \tag{15}$$

$$T_\psi \coloneqq \bigcup_{k'=1}^{K_\psi} S'_{k'}, \text{ with } S'_{k'} \coloneqq \{(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^{r_1+r_3} \mid \bigwedge_{j=1}^{n_{k'}} g_{k',j}(\mathbf{x}, \mathbf{z}) \geq 0\}. \tag{16}$$

Since an interpolant contains only common symbols of ϕ and ψ, Problem 1 can be reduced to finding a function $h(\mathbf{x})$ such that $h(\mathbf{x}) = 0$ separates the two projection sets $P_{\mathbf{x}}(T_\phi) \coloneqq \{\mathbf{x} \in \mathbb{R}^{r_1} \mid \exists \mathbf{y}.\ (\mathbf{x}, \mathbf{y}) \in T_\phi\}$ and $P_{\mathbf{x}}(T_\psi) \coloneqq \{\mathbf{x} \in \mathbb{R}^{r_1} \mid \exists \mathbf{z}.\ (\mathbf{x}, \mathbf{z}) \in T_\psi\}$. We have the following theorem as a direct consequence of Theorem 2.

Theorem 3. *Let $\phi(\mathbf{x},\mathbf{y})$ and $\psi(\mathbf{x},\mathbf{z})$ be defined in Problem 1, and let $P_{\mathbf{x}}(T_\phi)$ and $P_{\mathbf{x}}(T_\psi)$ be defined above. Let $T_1 = \mathrm{cl}(\mathrm{P}_{\mathbf{x}}(\mathrm{T}_\phi))$ and $T_2 = \mathrm{cl}(\mathrm{P}_{\mathbf{x}}(\mathrm{T}_\psi))$. Assume $\tilde{T}_1 \cap \tilde{T}_2 = \emptyset$. Then there exists a semialgebraic function $h(\mathbf{x})$ of the form in Eq. (8) such that*

$$\forall \mathbf{x} \in P_{\mathbf{x}}(T_\phi).\ h(\mathbf{x}) > 0 \text{ and } \forall \mathbf{x} \in P_{\mathbf{x}}(T_\psi).\ -h(\mathbf{x}) > 0. \tag{17}$$

As a consequence, $h(\mathbf{x}) > 0$ is a semialgebraic interpolant of $\phi(\mathbf{x},\mathbf{y})$ and $\psi(\mathbf{x},\mathbf{z})$.

Remark 2. Note that in Theorem 3, we need to consider the closures $\mathrm{cl}(\mathrm{P}_{\mathbf{x}}(\mathrm{T}_\phi))$ and $\mathrm{cl}(\mathrm{P}_{\mathbf{x}}(\mathrm{T}_\psi))$ rather than $P_{\mathbf{x}}(T_\phi)$ and $P_{\mathbf{x}}(T_\psi)$ themselves. The reason lies in that the projections of closed semialgebraic sets are not necessarily closed. For example, consider $\phi(\mathbf{x},\mathbf{y}) \coloneqq x_1 x_2 - 1 \geq 0 \wedge x_2 \geq 0$ with $\mathbf{x} = x_1$ and $\mathbf{y} = x_2$. Then $P_{\mathbf{x}}(T_\phi) = \{x_1 \mid x_1 > 0\}$ is an open set.

4 Sum-of-Squares Formulation

In this section, we provide SOS characterizations for polynomial and semialgebraic interpolants. For simplicity, we will focus on the case where ϕ and ψ are conjunctions of polynomial inequalities given by

$$\phi(\mathbf{x},\mathbf{y}) \coloneqq \bigwedge_{i=1}^{m} f_i(\mathbf{x},\mathbf{y}) \geq 0 \text{ and } \psi(\mathbf{x},\mathbf{z}) \coloneqq \bigwedge_{j=1}^{n} g_j(\mathbf{x},\mathbf{z}) \geq 0, \tag{18}$$

where $\mathbf{x} \in \mathbb{R}^{r_1}$, $\mathbf{y} \in \mathbb{R}^{r_2}$, and $\mathbf{z} \in \mathbb{R}^{r_3}$. Extending to the general case is straightforward.

4.1 SOS Characterization for Polynomial Interpolants

In this part, we provide an SOS characterization for polynomial interpolants based on homogenization. We prove that the characterization is sound and weakly complete. Furthermore, we provide a concrete example to show that our new characterization is strictly more expressive than the one in [11].

Theorem 4 (Weak Completeness). *Let ϕ, ψ be defined as in Eq. (18) and let S_ϕ, and S_ψ be the basic semialgebraic sets corresponding to ϕ and ψ. Let $\tilde{f}_{m+1} = x_0$, $\tilde{g}_{n+1} = x_0$, $\tilde{f}_{m+2} = x_0^2 + \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - 1$, and $\tilde{g}_{n+2} = x_0^2 + \|\mathbf{x}\|^2 + \|\mathbf{z}\|^2 - 1$. If $h(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ is a polynomial interpolant function of ϕ and ψ, then the homogenized polynomial $\tilde{h}(x_0,\mathbf{x})$ satisfies, for arbitrarily small $\epsilon > 0$,*

$$\begin{aligned} \tilde{h}(x_0,\mathbf{x}) + \epsilon &= \sigma_0 + \sum_{i=1}^{m+2} \sigma_i \tilde{f}_i(x_0,\mathbf{x},\mathbf{y}), \\ -\tilde{h}(x_0,\mathbf{x}) + \epsilon &= \tau_0 + \sum_{j=1}^{n+2} \tau_j \tilde{g}_j(x_0,\mathbf{x},\mathbf{z}), \end{aligned} \tag{19}$$

for some $\sigma_i \in \Sigma[x_0,\mathbf{x},\mathbf{y}]$, $i = 0,\ldots,m+1$, $\sigma_{m+2} \in \mathbb{R}[x_0,\mathbf{x},\mathbf{y}]$, $\tau_i \in \Sigma[x_0,\mathbf{x},\mathbf{z}]$, $i = 0,\ldots,n+1$, $\tau_{n+2} \in \mathbb{R}[x_0,\mathbf{x},\mathbf{z}]$.

Remark 3. In Eq. (19), we add a small quantity $\epsilon > 0$ to the left-hand sides in order to invoke Theorem 1. The ideal case is $\epsilon = 0$. Fortunately, in most practice circumstances, we can safely set $\epsilon = 0$ when the *finite convergence property* [34, Theorem 1.1] holds. Indeed, the finite convergence property is generically true and is violated only when h and S_ϕ (or S_ψ) are of certain singular forms.

Theorem 5 (Soundness). *Let ϕ and ψ be defined as in Eq. (18). Suppose that $h(\mathbf{x})$ is a polynomial such that its homogenization $\tilde{h}(x_0, \mathbf{x})$ satisfies Eq. (19) with $\epsilon = 0$. Assume $\phi \wedge h(\mathbf{x}) = 0 \models \bot$ and $\psi \wedge h(\mathbf{x}) = 0 \models \bot$. Then $h(\mathbf{x})$ is an interpolant function of ϕ and ψ.*

Now we compare our characterization Eq. (19) with [11, Theorem 5] which states that if S_ϕ and S_ψ are of the Archimedean form, then a polynomial interpolant function $h(\mathbf{x})$ can be expressed as

$$\begin{aligned} h(\mathbf{x}) - 1 &= \sigma_0 + \sum_{i=0}^{m} \sigma_i f_i(\mathbf{x}, \mathbf{y}), \\ -h(\mathbf{x}) - 1 &= \tau_0 + \sum_{j=0}^{n} \tau_j g_j(\mathbf{x}, \mathbf{z}), \end{aligned} \tag{20}$$

for some $\sigma_i \in \Sigma[\mathbf{x}, \mathbf{y}]$, $i = 0, \ldots, m$ and $\tau_j \in \Sigma[\mathbf{x}, \mathbf{z}]$, $j = 0, \ldots, n$. Clearly, since Theorem 4 removes the restriction of the Archimedean condition, our characterization is *strictly* more expressive.

Let $M(x_1, x_2) \coloneqq x_1^4 x_2^2 + x_1^2 x_2^4 - 3x_1^2 x_2^2 + 1$ be the Motzkin polynomial. It is well known that $M(x_1, x_2)$ is nonnegative but is not an SOS.

Proposition 3. *The polynomial $M(x_1, x_2) + 1$ is positive but is not an SOS.*

Example 1. Let $\phi \coloneqq 1 \geq 0 (= \top)$ and $\psi \coloneqq -1 \geq 0 (= \bot)$. By Proposition 3, the polynomial $M(x_1, x_2) + 1$ is an interpolant function of ϕ and ψ but does not admit a representation as in Eq. (20), i.e., the program

$$\begin{aligned} \text{find} \quad & \sigma_0 \in \Sigma[x_1, x_2] \\ \text{s.t.} \quad & x_1^4 x_2^2 + x_1^2 x_2^4 - 3x_1^2 x_2^2 + 2 = \sigma_0 \end{aligned}$$

is not feasible. However, a numerical solution to the following program:

$$\begin{aligned} \text{find} \quad & \sigma_0, \sigma_1 \in \Sigma[x_0, x_1, x_2], \sigma_2 \in \mathbb{R}[x_0, x_1, x_2] \\ \text{s.t.} \quad & x_1^4 x_2^2 + x_1^2 x_2^4 - 3x_1^2 x_2^2 x_0^2 + 2x_0^6 = \sigma_0 + \sigma_1 x_0 + \sigma_2 (1 - x_0^2 - x_1^2 - x_2^2). \end{aligned}$$

can be obtained by employing the Julia package TSSOS [28] and the SDP solver Mosek [2]. Therefore, the polynomial $M(x_1, x_2) + 1$ admits a representation as in Eq. (19) with $\epsilon = 0$.

4.2 SOS Characterization for Semialgebraic Interpolants

Let $h(\mathbf{x})$ be a semialgebraic interpolant function of the form in Eq. (8) and let w be a fresh variable. Though $h(\mathbf{x})$ is not a polynomial, it can be equivalently represented by a polynomial $l(\mathbf{x}, w) = h_1(\mathbf{x}) + w \cdot h_2(\mathbf{x}) \in \mathbb{R}[\mathbf{x}, w]$ with additional polynomial constraints $w^2 = 1 + \|\mathbf{x}\|^2$ and $w \geq 0$. Adopting this idea, we have the following completeness theorem. The soundness result for the semialgebraic case is omitted, as it is essentially the same as Theorem 5.

Theorem 6 (Completeness). *Let ϕ, ψ be defined as in Eq. (18) and let S_ϕ, and S_ψ be the basic semialgebraic sets corresponding to ϕ and ψ. Let $S_1 = \mathrm{cl}(\mathrm{P}_{\mathbf{x}}(\mathrm{S}_\phi))$, $S_2 = \mathrm{cl}(\mathrm{P}_{\mathbf{x}}(\mathrm{S}_\psi))$, $\tilde{f}_{m+1} = \tilde{g}_{n+1} = x_0$, $\tilde{f}_{m+2} = \tilde{g}_{m+2} = w$, $\tilde{f}_{m+3} = x_0^2 + \|\mathbf{x}\|^2 + w^2 + \|\mathbf{y}\|^2 - 1$, and $\tilde{g}_{n+3} = x_0^2 + \|\mathbf{x}\|^2 + w^2 + \|\mathbf{z}\|^2 - 1$, $\tilde{f}_{m+4} = \tilde{g}_{n+4} = x_0^2 + \|\mathbf{x}\|^2 - w^2$. Assume that the following two conditions hold: (1) $\tilde{S}_1 \cap \tilde{S}_2 = \emptyset$; (2) S_ϕ and S_ψ are closed at ∞. Then there exists a semialgebraic interpolant function $h(\mathbf{x})$ of the form in Eq. (8) such that the polynomial $l(\mathbf{x}, w) = h_1(\mathbf{x}) + w \cdot h_2(\mathbf{x}) \in \mathbb{R}[\mathbf{x}, w]$ satisfies, for arbitrarily small $\epsilon > 0$,*

$$\begin{aligned} \tilde{l}(x_0, \mathbf{x}, w) + \epsilon &= \sigma_0 + \sum_{i=1}^{m+4} \sigma_i \tilde{f}_i(x_0, \mathbf{x}, \mathbf{y}), \\ -\tilde{l}(x_0, \mathbf{x}, w) + \epsilon &= \tau_0 + \sum_{j=1}^{n+4} \tau_j \tilde{g}_j(x_0, \mathbf{x}, \mathbf{z}), \end{aligned} \tag{21}$$

for some $\sigma_i \in \Sigma[x_0, \mathbf{x}, \mathbf{y}, w]$, $i = 0, \ldots, m+2$, $\sigma_{m+3}, \sigma_{m+4} \in \mathbb{R}[x_0, \mathbf{x}, \mathbf{y}, w]$, $\tau_i \in \Sigma[x_0, \mathbf{x}, \mathbf{z}, w]$, $i = 0, \ldots, n+2$, $\tau_{n+3}, \tau_{n+4} \in \mathbb{R}[x_0, \mathbf{x}, \mathbf{z}, w]$.

We want to emphasize that Theorem 6 is a stronger result than Theorem 4, in the sense that Theorem 6 guarantees the existence of a semialgebraic interpolant (as per Theorem 3), which is not the case for polynomial interpolants in Theorem 4.

5 Synthesizing Interpolant via SOS Programming

In this section, we propose an SOS programming procedure to synthesize polynomial and semialgebraic interpolants. Concrete examples are provided to demonstrate the effectiveness and efficiency of our method. For all examples, existing approaches [7,10,12] are not applicable due to their restrictions on formulas, and the method in [11] also fails to produce interpolants of specified degrees.

Synthesizing Polynomial Interpolants: Let T_ϕ, T_ψ, S_k, and $S_{k'}$ be defined as in Eq. (15) and Eq. (16). By treating S_k and S'_k respectively as S_ϕ and S_ψ in Theorem 4, the problem of synthesizing a polynomial interpolant for ϕ and ψ is reduced to solving the following SOS program:

$$\begin{cases} \text{find} & h(\mathbf{x}) \\ \text{s.t.} & \tilde{h}(x_0,\mathbf{x}) = \sigma_{k,0} + \sum_{i=1}^{m_k+2} \sigma_{k,i}\tilde{f}_{k,i} \quad \text{for } k = 1,\ldots,K_\phi, \\ & -\tilde{h}(x_0,\mathbf{x}) = \tau_{k',0} + \sum_{j=1}^{n_{k'}+2} \tau_{k',j}\tilde{g}_{k',j} \quad \text{for } k' = 1,\ldots,K_\psi, \\ & \sigma_{k,0},\ldots,\sigma_{k,m+1} \in \Sigma[x_0,\mathbf{x},\mathbf{y}], \sigma_{k,m+2} \in \mathbb{R}[x_0,\mathbf{x},\mathbf{y}], \\ & \qquad \text{for } k = 1,\ldots,K_\phi, \\ & \tau_{k',0},\ldots,\tau_{k',n+1} \in \Sigma[x_0,\mathbf{x},\mathbf{z}], \tau_{k',n+2} \in \mathbb{R}[x_0,\mathbf{x},\mathbf{y}], \\ & \qquad \text{for } k = 1,\ldots,K_\psi, \end{cases} \tag{22}$$

where $\tilde{f}_{k,m+1} = \tilde{g}_{k',n+1} = x_0$, $\tilde{f}_{k,m+2} = x_0^2 + \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - 1$, $\tilde{g}_{k',n+2} = x_0^2 + \|\mathbf{x}\|^2 + \|\mathbf{z}\|^2 - 1$ for $k = 1,\ldots,K_\phi$ and $k' = 1,\ldots,K_\psi$.

As Theorem 5 suggests, a solution $h(\mathbf{x})$ to the above program only ensures that $\phi \models h(\mathbf{x}) \geq 0$ and $\psi \models -h(\mathbf{x}) \leq 0$. Nevertheless, since numerical solvers are unable to distinguish $\geq$ from $>$, the equalities are usually not attainable for a numerical solution[2]. Therefore, we can view the SOS program Eq. (22) as a sound approach for computing $h(\mathbf{x})$, while completeness follows from verifying the conditions discussed in Remark 3.

In practice, we solve the program Eq. (22) by solving a sequence of SDP relaxations which are obtained by restricting the highest degree of involved polynomials. Concretely speaking, suppose that we would like to find a polynomial interpolant function $h(\mathbf{x})$ of degree d, we set the template of $h(\mathbf{x})$ to be $h(\mathbf{x}) = \sum_{|\boldsymbol{\alpha}|\leq d} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}}$, where $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_{r_1}) \in \mathbb{N}^{r_1}$, $|\boldsymbol{\alpha}| = \alpha_1 + \cdots \alpha_{r_1}$, and $c_{\boldsymbol{\alpha}} \in \mathbb{R}$ are coefficients to be determined. Then, the homogenization of $h(\mathbf{x})$ is $\tilde{h}(x_0,\mathbf{x}) = \sum_{|\boldsymbol{\alpha}|\leq d} c_{\boldsymbol{\alpha}} x_0^{d-|\boldsymbol{\alpha}|}\mathbf{x}^{\boldsymbol{\alpha}}$.

Given a relaxation order $s \in \mathbb{N}$ with $2s \geq d$, we set the degrees of the remaining unknown polynomials σ_i, τ_j appropriately to ensure that the maximum degree of polynomials involved in Eq. (22) equals $2s$. We refer to the resulting program as the s-th relaxation of Eq. (22), which can be translated into an SDP and can be numerically solved in polynomial time. If the s-th relaxation is solvable, it yields a solution $h(\mathbf{x})$ that serves as a polynomial interpolant function of ϕ and ψ. If it is not solvable, we then increase the relaxation order s to obtain a tighter relaxation, or alternatively, we can increase the degree d of $h(\mathbf{x})$ to search for interpolants of higher degree.

In the following, all experiments were conducted on a Mac lap-top with Apple M2 chip and 8GB memory. We use the Julia package TSSOS [28] to formulate SOS programs and rely on the SDP solver Mosek [2] to solve them. All numerical results are symbolically verified using Mathematica to be real interpolants.

[2] For example, SDP solvers based on interior-point methods typically return strictly feasible solutions.

Example 2 (adapted from [4]). Let $\mathbf{x} = (x, y)$ and $\mathbf{y} = \mathbf{z} = \emptyset$, i.e., there is no uncommon variables. We define the following polynomials:

$$\begin{aligned}
f_1 &= 11 - x^4 + 0.1y^4, & f_2 &= y^3,\\
f_3 &= 0.9025 - (x-1)^4 - y^4, & f_4 &= (x-1)^4 + y^4 - 0.09,\\
f_5 &= (x+1)^4 + y^4 - 1.1025, & f_6 &= 0.04 - (x+1)^4 - y^4,\\
g_1 &= 11 - x^4 + 0.1y^4, & g_2 &= -y^3,\\
g_3 &= 0.9025 - (x+1)^4 - y^4, & g_4 &= (x+1)^4 + y^4 - 0.09,\\
g_5 &= (x-1)^4 + y^4 - 1.1025, & g_6 &= 0.04 - (x-1)^4 - y^4.
\end{aligned}$$

Let ϕ and ψ be defined by

$$\begin{aligned}
\phi &\coloneqq (f_1 \geq 0 \wedge f_2 \geq 0 \wedge f_4 \geq 0 \wedge f_5 \geq 0) \vee (f_3 \geq 0 \wedge f_4 \geq 0 \wedge f_5 \geq 0) \vee (f_6 \geq 0),\\
\psi &\coloneqq (g_1 \geq 0 \wedge g_2 \geq 0 \wedge g_4 \geq 0 \wedge g_5 \geq 0) \vee (g_3 \geq 0 \wedge g_4 \geq 0 \wedge g_5 \geq 0) \vee (g_6 \geq 0).
\end{aligned}$$

Set the degree of the polynomial interpolation function $h(x, y)$ to 7. It takes 0.16 s to solve the 4-th relaxation Eq. (22), yielding the solution

$$h(x, y) = -0.00153942y + 0.03053692x + \cdots + 0.06109453x^6y + 0.01643640x^7,$$

where the coefficients have been scaled so that the largest absolute value is 1.

Synthesizing Semialgebraic Interpolants: Similarly, the synthesis of a semialgebraic interpolant is reduced to solving the following SOS program:

$$\begin{cases}
\text{find} & h_1(\mathbf{x}), h_2(\mathbf{x})\\
\text{s.t.} & l(\mathbf{x}, w) = h_1(\mathbf{x}) + w \cdot h_2(\mathbf{x}),\\
& \tilde{l}(x_0, \mathbf{x}, w) = \sigma_{k,0} + \sum_{i=1}^{m_k+4} \sigma_{k,i}\tilde{f}_{k,i} \quad \text{for } k = 1, \ldots, K_\phi,\\
& -\tilde{l}(x_0, \mathbf{x}, w) = \tau_{k',0} + \sum_{j=1}^{n_{k'}+4} \tau_{k',j}\tilde{g}_{k',j} \quad \text{for } k' = 1, \ldots, K_\psi,\\
& \sigma_{k,0}, \ldots, \sigma_{k,m+2} \in \Sigma[x_0, \mathbf{x}, \mathbf{y}], \sigma_{k,m+3}, \sigma_{k,m+4} \in \mathbb{R}[x_0, \mathbf{x}, \mathbf{y}],\\
& \quad \text{for } k = 1, \ldots, K_\phi,\\
& \tau_{k',0}, \ldots, \tau_{k',n+2} \in \Sigma[x_0, \mathbf{x}, \mathbf{z}], \tau_{k',n+3}, \tau_{k',n+4} \in \mathbb{R}[x_0, \mathbf{x}, \mathbf{y}],\\
& \quad \text{for } k = 1, \ldots, K_\psi,
\end{cases} \tag{23}$$

where $\tilde{f}_{k,m+1} = \tilde{g}_{k',n+1} = x_0$, $\tilde{f}_{k,m+2} = \tilde{g}_{k',n+2} = w$, $\tilde{f}_{k,m+3} = x_0^2 + \|\mathbf{x}\|^2 + w^2 + \|\mathbf{y}\|^2 - 1$, $\tilde{g}_{k',n+3} = x_0^2 + \|\mathbf{x}\|^2 + w^2 + \|\mathbf{z}\|^2 - 1$, $\tilde{f}_{k,m+4} = \tilde{g}_{k',n+4} = x_0^2 + \|\mathbf{x}\|^2 - w^2$, for $k = 1, \ldots, K_\phi$ and $k' = 1, \ldots, K_\psi$.

By Theorem 6, if a feasible solution (h_1, h_2) of Eq. (23) is found, then $h(\mathbf{x}) = h_1(\mathbf{x}) + \sqrt{\|\mathbf{x}\|^2 + 1} \cdot h_2(\mathbf{x})$ is a semialgebraic interpolant function for ϕ and ψ. In practice, w.l.o.g., we can assume that h_1 and h_2 are of the same degree d and solve SDP relaxations of Eq. (23). The soundness result is similar to that of Eq. (22), requiring that $h(\mathbf{x}) = 0$ is not attainable over T_ϕ and T_ψ.

Example 3. Let $\mathbf{x} = (x, y)$, $\mathbf{y} = \mathbf{z} = \emptyset$. We define

$$\phi(x, y) = 8xy - (x^2 - y^3)^2 \geq 0 \wedge x^2 + y^2 - 1 \geq 0,$$
$$\psi(x, y) = -12.5xy - (x^2 + y^2)^2 \geq 0 \wedge x^2 + y^2 - 1 \geq 0.$$

Let the degree of $h_1(\mathbf{x})$ and $h_2(\mathbf{x})$ to be 3, a solution to the 2-th relaxation of Eq. (23) is found in 0.02 s:

$$h_1 = -0.04402209 - 0.00093184y + 0.01446436x + \cdots - 0.03703461x^3,$$
$$h_2 = 0.05644318 - 0.01305178y + 0.02407258x + \cdots + 0.23199837x^2.$$

As a comparison, solving Eq. (22) fails to produce a polynomial interpolant function of degree 3, but succeeds at degree 4.

Example 4. Let $\mathbf{x} = (x, y, z)$, $\mathbf{y} = \emptyset$, and $\mathbf{z} = (r, R)$. We define

$$\begin{aligned} \phi(x, y, z) \coloneqq\ & 1 + 0.1z^4 - x^4 - y^4 \geq 0 \wedge 10z^4 - x^4 - y^4 \geq 0, \\ \psi(x, y, z, r, R) \coloneqq\ & 4R^2(x^2 + y^2) - (x^2 + y^2 + z^2 + R^2 - r^2)^2 \geq 0 \\ & \wedge 6 \geq R \geq 4 \wedge 1 \geq r \geq 0.5, \end{aligned}$$

where $\exists r, \exists R.\ \psi(x, y, z, r, R)$ describes the set of interior points of a 3-dimensional torus with unknown minor radius $r \in [0.5, 1]$ and major radius $R \in [4, 6]$. By solving Eq. (22) and Eq. (23), we obtain a polynomial interpolant

$$h_p(x, y, z) = 1.0 - 0.35507338x^2 - 0.35507338y^2 + 0.45264895z^2,$$

and a semialgebraic interpolant function with

$$h_1(x, y, z) = 0.98004189 - 0.26291972x^2 - 0.26291978y^2 + 0.417581644z^2,$$
$$h_2(x, y, z) = 1.0 - 0.51670759x^2 - 0.51670759y^2 + 0.60569150z^2.$$

As a comparison, [11] fails to produce an interpolant of degree less than 4.

6 Conclusions and Future Work

In this paper, we have addressed the problem of synthesizing Craig interpolants for two general polynomial formulas. By combining the polynomial homogenization techniques with the approach from [11], we have presented a complete SOS characterization of semialgebraic (and polynomial) interpolants. Compared with existing works, our approach removes the restrictions on the form of formulas and is applicable to any polynomial formulas, especially when variables have unbound domains. Moreover, sparsity of polynomial formulas can be exploited to improve the scalability of our approach [16,29].

Our Craig interpolation synthesis technique offers broad applicability in various verification tasks. It can be used as a sub-procedure, for example, in CEGAR-based model checking for identifying counterexamples [32], in bounded model

checking for generating proofs [21], in program verification for squeezing invariants [26], and in SMT for reasoning about nonlinear arithmetic [18]. Compared with existing algorithms, our SDP-based algorithm is efficient and provides a relative completeness guarantee. However, the practical implementation is not a trivial undertaking, as it requires suitable strategies for storing numerical interpolants and taming numerical errors [37]. This remains an ongoing work of our research.

Data Availability Statement. The experimental results of this paper may be reproduced using the artifact on Figshare https://doi.org/10.6084/m9.figshare.26131378, or via GitHub link https://github.com/EcstasyH/Interpolation.

References

1. Acquistapace, F., Andradas, C., Broglia, F.: Separation of semialgebraic sets. J. Am. Math. Soc. **12**(3), 703–728 (1999). https://doi.org/10.1090/S0894-0347-99-00302-1
2. Andersen, E.D., Andersen, K.D.: The Mosek interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In: Frenk, H., Roos, K., Terlaky, T., Zhang, S. (eds.) High Performance Optimization, pp. 197–232. Springer US, Boston, MA (2000). https://doi.org/10.1007/978-1-4757-3216-0_8
3. Benhamou, F., Granvilliers, L.: Continuous and interval constraints. In: Handbook of Constraint Programming, Foundations of Artificial Intelligence, vol. 2, pp. 571–603 (2006). https://doi.org/10.1016/S1574-6526(06)80020-9
4. Chen, M., Wang, J., An, J., Zhan, B., Kapur, D., Zhan, N.: NIL: learning nonlinear interpolants. In: Fontaine, P. (ed.) Automated Deduction – CADE 27: 27th International Conference on Automated Deduction, Natal, Brazil, August 27–30, 2019, Proceedings, pp. 178–196. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-29436-6_11
5. Cimatti, A., Griggio, A., Sebastiani, R.: Efficient interpolation generation in satisfiability modulo theories. In: Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2008. Lecture Notes in Computer Science, vol. 4963, pp. 397–412 (2008). https://doi.org/10.1007/978-3-540-78800-3_30
6. Cimatti, A., Griggio, A., Irfan, A., Roveri, M., Sebastiani, R.: Incremental linearization for satisfiability and verification modulo nonlinear arithmetic and transcendental functions. ACM Trans. Comput. Log. **19**(3), 19:1–19:52 (2018). https://doi.org/10.1145/3230639
7. Dai, L., Xia, B., Zhan, N.: Generating non-linear interpolants by semidefinite programming. In: Sharygina, N., Veith, H. (eds.) Computer Aided Verification - 25th International Conference, CAV 2013. Lecture Notes in Computer Science, vol. 8044, pp. 364–380. Springer (2013). https://doi.org/10.1007/978-3-642-39799-8_25
8. Davenport, J.H., Heintz, J.: Real quantifier elimination is doubly exponential. J. Symb. Comput. **5**(1–2), 29–35 (1988). https://doi.org/10.1016/S0747-7171(88)80004-X
9. D'Silva, V.V., Kroening, D., Purandare, M., Weissenbacher, G.: Interpolant strength. In: Verification, Model Checking, and Abstract Interpretation, 11th International Conference, VMCAI 2010. Lecture Notes in Computer Science, vol. 5944, pp. 129–145. Springer (2010). https://doi.org/10.1007/978-3-642-11319-2_12

10. Gan, T., Dai, L., Xia, B., Zhan, N., Kapur, D., Chen, M.: Interpolant synthesis for quadratic polynomial inequalities and combination with EUF. In: Automated Reasoning: 8th International Joint Conference, IJCAR 2016, pp. 195–212. Springer (2016). https://doi.org/10.1007/978-3-319-40229-1_14
11. Gan, T., Xia, B., Xue, B., Zhan, N., Dai, L.: Nonlinear Craig interpolant generation. In: Computer Aided Verification - 32nd International Conference, CAV 2020. Lecture Notes in Computer Science, vol. 12224, pp. 415–438. Springer (2020). https://doi.org/10.1007/978-3-030-53288-8_20
12. Gao, S., Kong, S., Clarke, E.M.: Proof generation from delta-decisions. In: 16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2014, pp. 156–163. IEEE Computer Society (2014). https://doi.org/10.1109/SYNASC.2014.29
13. Gao, S., Zufferey, D.: Interpolants in nonlinear theories over the reals. In: Tools and Algorithms for the Construction and Analysis of Systems - 22nd International Conference, TACAS 2016. Lecture Notes in Computer Science, vol. 9636, pp. 625–641. Springer (2016). https://doi.org/10.1007/978-3-662-49674-9_41
14. Henzinger, T.A., Jhala, R., Majumdar, R., McMillan, K.L.: Abstractions from proofs. In: Proceedings of the 31st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2004, pp. 232–244. ACM (2004). https://doi.org/10.1145/964001.964021
15. Hoenicke, J., Schindler, T.: Efficient interpolation for the theory of arrays. In: Automated Reasoning - 9th International Joint Conference, IJCAR 2018. Lecture Notes in Computer Science, vol. 10900, pp. 549–565. Springer (2018). https://doi.org/10.1007/978-3-319-94205-6_36
16. Huang, L., Kang, S., Wang, J., Yang, H.: Sparse polynomial optimization with unbounded sets (2024). https://arxiv.org/abs/2401.15837
17. Huang, L., Nie, J., Yuan, Y.: Homogenization for polynomial optimization with unbounded sets. Math. Program. **200**(1), 105–145 (2023). https://doi.org/10.1007/S10107-022-01878-5
18. Jovanovic, D., Dutertre, B.: Interpolation and model checking for nonlinear arithmetic. In: Computer Aided Verification - 33rd International Conference, CAV 2021. Lecture Notes in Computer Science, vol. 12760, pp. 266–288. Springer (2021). https://doi.org/10.1007/978-3-030-81688-9_13
19. Jung, Y., Lee, W., Wang, B., Yi, K.: Predicate generation for learning-based quantifier-free loop invariant inference. In: Tools and Algorithms for the Construction and Analysis of Systems - 17th International Conference, TACAS 2011. Lecture Notes in Computer Science, vol. 6605, pp. 205–219. Springer (2011). https://doi.org/10.1007/978-3-642-19835-9_17
20. Kapur, D., Majumdar, R., Zarba, C.G.: Interpolation for data structures. In: Proceedings of the 14th ACM SIGSOFT International Symposium on Foundations of Software Engineering, FSE 2006, pp. 105–116. ACM (2006). https://doi.org/10.1145/1181775.1181789
21. Komuravelli, A., Gurfinkel, A., Chaki, S.: SMT-based model checking for recursive programs. In: Computer Aided Verification - 26th International Conference, CAV 2014. Lecture Notes in Computer Science, vol. 8559, pp. 17–34. Springer (2014). https://doi.org/10.1007/978-3-319-08867-9_2
22. Kovács, L., Voronkov, A.: Interpolation and symbol elimination. In: 22nd International Conference on Automated Deduction, CADE'22. Lecture Notes in Computer Science, vol. 5663, pp. 199–213. Springer (2009). https://doi.org/10.1007/978-3-642-02959-2_17

23. Krajícek, J.: Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. J. Symb. Log. **62**(2), 457–486 (1997). https://doi.org/10.2307/2275541
24. Kupferschmid, S., Becker, B.: Craig interpolation in the presence of non-linear constraints. In: Fahrenberg, U., Tripakis, S. (eds.) Formal Modeling and Analysis of Timed Systems - 9th International Conference, FORMATS 2011. Lecture Notes in Computer Science, vol. 6919, pp. 240–255. Springer (2011). https://doi.org/10.1007/978-3-642-24310-3_17
25. Lasserre, J.B.: Moments, positive polynomials and their applications, vol. 1. World Scientific (2009). https://doi.org/10.1142/p665
26. Lin, S., Sun, J., Xiao, H., Sanán, D., Hansen, H.: Fib: Squeezing loop invariants by interpolation between forward/backward predicate transformers. In: Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering, ASE 2017, pp. 793–803. IEEE Computer Society (2017). https://doi.org/10.1109/ASE.2017.8115690
27. Lin, W., Ding, M., Lin, K., Mei, G., Ding, Z.: Formal synthesis of neural Craig interpolant via counterexample guided deep learning. In: 9th International Conference on Dependable Systems and Their Applications, DSA 2022, pp. 116–125. IEEE (2022). https://doi.org/10.1109/DSA56465.2022.00023
28. Magron, V., Wang, J.: TSSOS: a Julia library to exploit sparsity for large-scale polynomial optimization. CoRR **abs/2103.00915** (2021). https://arxiv.org/abs/2103.00915
29. Magron, V., Wang, J.: Sparse Polynomial Optimization - Theory and Practice, Series on Optimization and its Applications, vol. 5. WorldScientific (2023). https://doi.org/10.1142/Q0382
30. Marshall, M.: Positive polynomials and sums of squares. Am. Math. Soc., 146 (2008)
31. McMillan, K.L.: Interpolation and sat-based model checking. In: Computer Aided Verification, 15th International Conference, CAV 2003. Lecture Notes in Computer Science, vol. 2725, pp. 1–13. Springer (2003). https://doi.org/10.1007/978-3-540-45069-6_1
32. McMillan, K.L.: An interpolating theorem prover. Theor. Comput. Sci. **345**(1), 101–121 (2005). https://doi.org/10.1016/J.TCS.2005.07.003
33. McMillan, K.L.: Quantified invariant generation using an interpolating saturation prover. In: Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008. Lecture Notes in Computer Science, vol. 4963, pp. 413–427. Springer (2008). https://doi.org/10.1007/978-3-540-78800-3_31
34. Nie, J.: Optimality conditions and finite convergence of Lasserre's hierarchy. Math. Program. **146**(1–2), 97–121 (2014). https://doi.org/10.1007/S10107-013-0680-X
35. Pudlák, P.: Lower bounds for resolution and cutting plane proofs and monotone computations. J. Symb. Log. **62**(3), 981–998 (1997). https://doi.org/10.2307/2275583
36. Putinar, M.: Positive polynomials on compact semi-algebraic sets. Indiana Univ. Math. J. **42**(3), 969–984 (1993). https://www.jstor.org/stable/24897130
37. Roux, P., Voronin, Y., Sankaranarayanan, S.: Validating numerical semidefinite programming solvers for polynomial invariants. Formal Methods Syst. Design **53**(2), 286–312 (2018). https://doi.org/10.1007/s10703-017-0302-y
38. Rybalchenko, A., Sofronie-Stokkermans, V.: Constraint solving for interpolation. J. Symb. Comput. **45**(11), 1212–1233 (2010). https://doi.org/10.1016/J.JSC.2010.06.005

39. Sofronie-Stokkermans, V.: Interpolation in local theory extensions. Log. Methods Comput. Sci. **4**(4) (2008). https://doi.org/10.2168/LMCS-4(4:1)2008
40. Srikanth, A., Sahin, B., Harris, W.R.: Complexity verification using guided theorem enumeration, pp. 639–652 (2017). https://doi.org/10.1145/3009837.3009864
41. Stengle, G.: A nullstellensatz and a positivstellensatz in semialgebraic geometry. Ann. Math. **207**, 87–97 (1974). https://doi.org/10.1007/BF01362149
42. Wu, H., Wang, J., Xia, B., Li, X., Zhan, N., Gan, T.: Nonlinear Craig interpolant generation over unbounded domains by separating semialgebraic sets (2024). https://arxiv.org/abs/2407.00625
43. Yorsh, G., Musuvathi, M.: A combination method for generating interpolants. In: 20th International Conference on Automated Deduction, CADE'20. Lecture Notes in Computer Science, vol. 3632, pp. 353–368. Springer (2005). https://doi.org/10.1007/11532231_26
44. Zhan, N., Wang, S., Zhao, H.: Formal Verification of Simulink/Stateflow Diagrams. A Deductive Approach. Springer (2017). https://doi.org/10.1007/978-3-319-47016-0
45. Zhao, H., Zhan, N., Kapur, D., Larsen, K.G.: A "hybrid" approach for synthesizing optimal controllers of hybrid systems: a case study of the oil pump industrial example. In: Formal Methods - 18th International Symposium, FM 2012, Lecture Notes in Computer Science, vol. 7436, pp. 471–485. Springer (2012). https://doi.org/10.1007/978-3-642-32759-9_38

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

