EFFICIENT COMPUTATION OF A SEMI-ALGEBRAIC BASIS OF THE FIRST HOMOLOGY GROUP OF A SEMI-ALGEBRAIC SET

SAUGATA BASU AND SARAH PERCIVAL

Abstract. Let R be a real closed field and C the algebraic closure of R. We give an algorithm for computing a semi-algebraic basis for the first homology group, $\mathrm{H}_1(S, \mathbb{F})$, with coefficients in a field $\mathbb{F}$, of any given semi-algebraic set $S \subset \mathrm{R}^k$ defined by a closed formula. The complexity of the algorithm is bounded singly exponentially. More precisely, if the given quantifier-free formula involves s polynomials whose degrees are bounded by d, the complexity of the algorithm is bounded by $(sd)^{k^{O(1)}}$. This algorithm generalizes well known algorithms having singly exponential complexity for computing a semi-algebraic basis of the zero-th homology group of semi-algebraic sets, which is equivalent to the problem of computing a set of points meeting every semi-algebraically connected component of the given semi-algebraic set at a unique point. It is not known how to compute such a basis for the higher homology groups with singly exponential complexity.

As an intermediate step in our algorithm we construct a semi-algebraic subset Γ of the given semi-algebraic set S, such that $\mathrm{H}_q(S, \Gamma) = 0$ for $q = 0, 1$. We relate this construction to a basic theorem in complex algebraic geometry stating that for any affine variety X of dimension n, there exists Zariski closed subsets

$$Z^{(n-1)} \supset \cdots \supset Z^{(1)} \supset Z^{(0)}$$

with $\dim_{\mathrm{C}} Z^{(i)} \leq i$, and $\mathrm{H}_q(X, Z^{(i)}) = 0$ for $0 \leq q \leq i$. We conjecture a quantitative version of this result in the semi-algebraic category, with X and $Z^{(i)}$ replaced by closed semi-algebraic sets. We make initial progress on this conjecture by proving the existence of $Z^{(0)}$ and $Z^{(1)}$ with complexity bounded singly exponentially (previously, such an algorithm was known only for constructing $Z^{(0)}$).

Contents

Date: **January 28, 2025**.

2020 *Mathematics Subject Classification.* Primary 14F25; Secondary 68W30.

Key words and phrases. semi-algebraic sets, singly exponential, homology groups, basis.

Basu was partially supported by NSF grants CCF-1618918, DMS-1620271 and CCF-1910441.

1. Introduction

We fix a real closed field R, and denote by $\mathrm{D} \subset \mathrm{R}$ a fixed ordered domain. We will denote by $\mathrm{C} = \mathrm{R}[i]$ the algebraic closure of R. For example, one can take $\mathrm{R} = \mathbb{R}$, and $\mathrm{D} = \mathbb{Z}$.

Semi-algebraic subsets of R^k, $k \geq 0$, are subsets of R^k defined by quantifier-free first-order formulas with atoms of the form

$$P = 0, P > 0, P \in \mathrm{R}[X_1, \ldots, X_k].$$

Algorithmic semi-algebraic geometry deals with computing geometric and topological invariants of semi-algebraic subsets of R^k, and is a very well-developed topic. Examples of algorithmic problems in semi-algebraic geometry that have been investigated include effective quantifier-elimination, the decision problem of the first order theory of the reals, computing topological invariants, such as the dimension, number of semi-algebraically connected components, the Euler-Poincaré characteristic (appropriately defined), and more generally the Betti numbers of a given semi-algebraic set.

1.1. **Background and history.**

The problem of computing the Betti numbers (i.e. the ranks of homology groups) of semi-algebraic sets has a long history and is an active area of current research. Since algorithms having doubly exponential complexity for computing the Betti numbers follow from effective triangulation algorithms for semi-algebraic sets, the emphasis has been on obtaining algorithms with singly exponential complexity (see Section 1.3 below). Singly exponential algorithms for computing the zero-th Betti number (i.e. the number of connected components) of semi-algebraic sets via construction of roadmaps was obtained by several authors and the complexity of the algorithms successively improved over the years [17, 21, 22, 24, 4].

An algorithm with singly exponential complexity is known for computing the first Betti number of semi-algebraic sets and is given in [11], and then extended to the first ℓ (for any fixed ℓ) Betti numbers in [2]. The Euler-Poincaré characteristic, which is the alternating sum of the Betti numbers, is easier to compute, and a singly exponential algorithm for computing it is known [1, 5]. While many advances have been made in recent years [2, 11], the best algorithm for computing *all* the Betti numbers of any given semi-algebraic set $S \subset \mathrm{R}^k$ still has doubly exponential (in k) complexity, even in the case where the degrees of the defining polynomials are assumed to be bounded by a constant (≥ 2) [30] (here we are talking about exact algorithms, see Section 1.1.1 below for a different model). The existence of algorithms with *singly exponential complexity* for computing all the Betti numbers of a given semi-algebraic set is considered to be a major open question in algorithmic semi-algebraic geometry (see the survey [8]).

Unlike the singly exponential complexity algorithms for computing the zero-th Betti numbers, the algorithms for computing the higher Betti numbers do not produce a semi-algebraic basis. Obtaining such a basis efficiently and of small complexity is of interest in geometric applications. In classical algebraic geometry over algebraically closed fields, representing homology classes by *algebraic* cycles is a well studied problem with deep connections to Hodge theory. The existence of a *semi-algebraic* basis with singly exponential complexity for the higher homology groups of a given semi-algebraic set is not known (other than in the zero-th homology case discussed above). In this paper we partially remedy this deficiency by proving the existence of such a basis in the case of the first homology group (cf. Theorem 2).

We note here that in the category of finite simplicial complexes, obtaining optimal representatives (as cycles) of homology classes is a well-studied problem. Early work on efficient algorithms for obtaining a shortest set of loops generating the first homology group of a two dimensional oriented manifold (given as a simplicial complex) appears in [20] (see for example, [19, 27] for recent work on this topic). However, these algorithms are primarily combinatorial in nature and the main difficulty in the semi-algebraic version of the problem is precisely that a triangulation of the given semi-algebraic set is not available (at least not known to be computable within the complexity we are aiming for).

1.1.1. *Exact vs numeric.* We remark here that by the word "algorithm" in the previous paragraphs we are referring only to algorithms that work correctly for all inputs and whose complexity is uniformly bounded, i.e. bounded in terms of the degrees and the number of input polynomials and independent of the actual coefficients of the polynomials (so in particular they always terminate). In contrast to this exact/symbolic model which is valid over arbitrary real closed fields, in numerical analysis it is common to consider algorithms whose complexity do depend on the coefficients (via a condition number). Such algorithms work only over the field of real numbers and might not terminate on ill-conditioned inputs (i.e. if the condition number is infinite). In this latter model, algorithms with singly exponential complexity for computing all the Betti numbers of semi-algebraic sets have been developed [14, 15, 16]. As noted above, these algorithms will fail to produce any result on certain inputs. Also, they do not produce semi-algebraic bases for the homology groups. In this paper we will be concerned only with exact algorithms that work for all possible inputs.

1.2. **Model of computation and definition of complexity.**

There are several models of computation that one can consider while dealing with semi-algebraic sets (and also several notions of what constitutes an algorithm). If the real closed field $\mathrm{R} = \mathbb{R}$, and $\mathrm{D} = \mathbb{Z}$, one can consider these algorithmic problems in the classical Turing model and measure the bit complexity of the algorithms. In this paper, we will follow the book [6] and take a more general approach valid over arbitrary real closed fields. In the particular case, when $\mathrm{D} = \mathbb{Z}$, our method will yield bit-complexity bounds. The precise notion of complexity that we use is defined in Definition 1 below.

1.2.1. *Definition of complexity.*

We will use the following notion of "complexity" in this paper. We follow the same definition as used in the book [6].

Definition 1 (Complexity of algorithms)**.** In our algorithms we will usually take as input quantifier-free first order formulas whose terms are polynomials with coefficients belonging to an ordered domain D contained in a real closed field R. By *complexity of an algorithm* we will mean the number of arithmetic operations and comparisons in the domain D. If $\mathrm{D} = \mathbb{R}$, then the complexity of our algorithm will agree with the Blum-Shub-Smale notion of real number complexity [13] [1].

It is also useful for what follows to introduce the following mathematical definition of "complexity" of formulas and semi-algebraic sets.

[1] In case $\mathrm{D} = \mathbb{Z}$, it is possible to deduce the bit-complexity of our algorithms in terms of the bit-sizes of the coefficients of the input polynomials, and this will agree with the classical (Turing) notion of complexity. We do not state the bit complexity separately in our algorithms, but note that it is always bounded by a polynomial in the bit-size of the input times the complexity upper bound stated in the paper. We do not count the cost of doing linear algebra over the field $\mathbb{F}$. However, if $\mathbb{F} = \mathbb{Q}$, then the bit complexity of doing operations over $\mathbb{F}$ will be bounded by the complexity upper bound stated in the paper.

1.2.2. *$\mathcal{P}$-formulas, $\mathcal{P}$-semi-algebraic sets, realizations.*

Notation 1 ($\mathcal{P}$-formulas, $\mathcal{P}$-closed formulas and their realizations)**.** For any finite set of polynomials $\mathcal{P} \subset \mathrm{R}[X_1, \ldots, X_k]$, we call a quantifier-free first order formula Φ with atoms $P = 0, P < 0, P > 0, P \in \mathcal{P}$, to be a *$\mathcal{P}$-formula*. Given any semi-algebraic subset $Z \subset \mathrm{R}^k$, we call the *realization of* Φ *in* Z, namely the semi-algebraic set

$$\mathcal{R}(\Phi, Z) \quad := \quad \{\mathbf{x} \in Z \mid \Phi(\mathbf{x})\}$$

a *$\mathcal{P}$-semi-algebraic subset of* Z. If $Z = \mathrm{R}^k$, we often denote the realization of Φ in R^k by $\mathcal{R}(\Phi)$.

We say that a quantifier-free formula Φ is *closed* if it is a formula in disjunctive normal form with no negations, and with atoms of the form $P \geq 0, P \leq 0$, where $P \in \mathrm{R}[X_1, \ldots, X_k]$. If the set of polynomials appearing in a closed formula is contained in a finite set $\mathcal{P}$, we will call such a formula a *$\mathcal{P}$-closed formula*, and we call the realization, $\mathcal{R}(\Phi)$, a *$\mathcal{P}$-closed semi-algebraic set.*

Definition 2 (Complexity of semi-algebraic sets)**.** For $\mathcal{P} \subset \mathrm{D}[X_1, \ldots, X_k]$, and $\mathcal{P}$-formula (resp. $\mathcal{P}$-closed formula) Φ, we say that the complexity of Φ is bounded by C, where $C = \mathrm{card}(\mathcal{P}) \cdot d$, where $d = \max_{P \in \mathcal{P}} \deg(P)$. If $S = \mathcal{R}(\Phi)$, then we will say that the complexity of S is bounded by $\mathrm{card}(\mathcal{P}) \cdot d$.

For the rest of the paper we fix a field $\mathbb{F}$.

Notation 2. For any closed semi-algebraic set X, we will denote by $\mathrm{H}_i(X) = \mathrm{H}_i(X, \mathbb{F})$ the i-th homology group of X with coefficients in $\mathbb{F}$ (we refer the reader to [6, Chapter 6] for definition of homology groups of semi-algebraic subsets of R^k, where R is an arbitrary real closed field).

Notation 3. Given $\mathbf{x} \in \mathrm{R}^k, r > 0$, we will denote by $B_k(\mathbf{x}, r) \subset \mathrm{R}^k$ the (open) euclidean ball of radius r centered at $\mathbf{x}$, and by $\mathbf{S}^{k-1}(\mathbf{x}, r) \subset \mathrm{R}^k$, the sphere of radius r centered at $\mathbf{x}$. Note that these are semi-algebraic subsets of R^k.

1.3. **Singly vs doubly exponential.** The problem of computing topological invariants (such as the number of semi-algebraically connected components) of semi-algebraic sets in general is a hard problem (known to be PSPACE-hard in the Turing model).

From the point of view of upper bounds on the complexity, these problems can be solved by combinatorial means if we have in hand a triangulation of the given semi-algebraic set. A semi-algebraic triangulation of a closed and bounded semi-algebraic set $S \subset \mathrm{R}^k$, consists of a finite simplicial complex K, and a semi-algebraic homeomorphism $h : |K| \to S$ (where $| \cdot |$ denotes the *geometric realization* functor). Moreover, the homology group $\mathrm{H}_*(S)$ is isomorphic to the simplicial homology groups $\mathrm{H}_*(K)$ which can be computed using standard linear algebra with complexity polynomial in the size of K.

Closed and bounded semi-algebraic sets admit semi-algebraic triangulations, and more pertinently such triangulations can be effectively computed. However, the algorithms with the best complexity for computing such triangulations have *doubly exponential* complexity (doubly exponential in k). More precisely, if $S \subset \mathrm{R}^k$ is defined by a quantifier-free formula involving s polynomials of degrees at most d, the best algorithm for computing a semi-algebraic triangulation of S is bounded by $(sd)^{2^{O(k)}}$.

It is a common belief in algorithmic semi-algebraic geometry that topological invariants satisfying a certain bound (say singly exponential) should in fact be computable by algorithms with complexity reflecting the mathematical bound. So invariants which are bounded singly exponential should in fact be computable by algorithms with singly exponentially bounded complexity. The intuition behind this belief is that the natural way to compute a topological invariant (such as the the zero-th Betti number of a semi-algebraic set) is often by computing a semi-algebraic representative or witnessing set (for example, a semi-algebraic basis of $\mathrm{H}_0(S)$ in the case of the zero-th homology – see Problem 2 below), and the cardinality of this witnessing set is the invariant to be computed. One expects the complexity of the algorithm for computing the witnessing set should reflect the complexity of this set – which includes the cardinality but also its "algebraic complexity" as well. The Betti numbers (ranks of homology groups) of semi-algebraic sets admit singly exponential upper bounds [28, 31, 25]. From this point of view one expects that there should exist algorithms for computing the Betti numbers of semi-algebraic sets with complexity bounded singly exponentially. Indeed, algorithms for computing the zero-th Betti number (i.e. the number of semi-algebraically connected components [2]) of semi-algebraic sets have been investigated in depth, and nearly optimal algorithms are known for this problem.

1.4. **Computing a basis for $\mathrm{H}_0(S)$.**

As mentioned above, all algorithms for computing the zero-th Betti number (i.e. the number of semi-algebraically connected components) of a given semi-algebraic set actually solve the following more general problem.

Problem 1. Given a quantifier-free formula Φ defining a semi-algebraic subset $S \subset \mathrm{R}^k$, compute a finite subset $\Gamma \subset S$, such that for each connected component C of S, $\mathrm{card}(\Gamma \cap C) = 1$ (i.e. Γ contains a unique representative from each connected component of S).

Remark 1. Note that one needs to allow points in Γ whose coordinates are algebraic over the ring generated by the coefficients of the polynomials

[2] The reason behind insisting on "semi-algebraically" connected instead of just connected (in the Euclidean topology) is that over an arbitrary real closed field these two notions are distinct. On the other hand if $\mathrm{R} = \mathbb{R}$, then being semi-algebraically connected is equivalent to being connected (see for example [6, Theorem 5.22]).

appearing in the formula Φ. At this point we ignore the question of representation of such points (cf. Definition 4 below) other than commenting that any algorithm for solving this problem needs to address this issue.

Problem 1 has been studied in depth and we now have very close to optimal algorithms for solving it. The solution is in two steps.

Step 1. The first and easier step is solving a weaker problem of computing a finite subset $\Gamma \subset S$ of "sample points" with the property that for each connected component C of S, $\mathrm{card}(\Gamma \cap C) \geq 1$. (The above property is equivalent to the property that the zero-th homology, $\mathrm{H}_0(S, \Gamma)$, of the pair (S, Γ) is trivial.) There are now very efficient algorithms for solving this problem (see for example, [23, 3], [6, Algorithm 13.3 (Sample points on a variety)]). In fact, such algorithms form the basic building block for efficient algorithms for solving the quantifier-elimination problem in the theory of the reals.

Step 2. The second step is more complicated and involves solving the problem of deciding whether two given points in a semi-algebraic set S belong to the same semi-algebraically connected component of S efficiently. This problem has a long history. The key idea is that of a roadmap of a semi-algebraic set (cf. Definition 3 for a precise definition).

If we have a roadmap Γ of S which contains $\mathbf{x}, \mathbf{y} \in S$, then it is easy to decide whether $\mathbf{x}$ and $\mathbf{y}$ belong to the same semi-algebraically connected component of Γ, which tells us if they are in the same semi-algebraically connected component of S itself because of the defining property of a roadmap. There exists singly exponential complexity algorithms for construction of roadmaps of semi-algebraic sets [17, 21, 22, 24, 4], and hence for the problem of deciding whether two given points belong to the same semi-algebraically connected component of a semi-algebraic set. Once we can decide if two points of the set of sample points Γ belong to the same semi-algebraically connected component of S, we can then select exactly one point in every semi-algebraically connected component of S and thus obtain a singly-exponential algorithm for solving Problem 1.

1.4.1. *Interpretation in terms of a homology basis.* Note that Problem 1 can be reformulated in terms of homology as follows.

Problem 2. Given a quantifier-free formula Φ defining a semi-algebraic subset $S \subset \mathrm{R}^k$, compute a semi-algebraic basis of $\mathrm{H}_0(S)$. More precisely, compute a semi-algebraic subset $\Gamma = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset S$, such that the subspaces $[\mathbf{x}_1], \ldots, [\mathbf{x}_N] \subset \mathrm{H}_0(S)$ are linearly independent and span $\mathrm{H}_0(S)$. Here $[\mathbf{x}_i]$ is the image of $\mathrm{H}_0(\{\mathbf{x}_i\})(\cong \mathbb{F})$ in $\mathrm{H}_0(S)$ under the linear map induced by the inclusion $\{\mathbf{x}_i\} \hookrightarrow S$.

The discussion in the beginning of this subsection yields the following theorem.

Theorem 1. [17, 21, 22, 24, 4] *There exists an algorithm that takes as input a finite set*

$$\mathcal{P} \subset \mathrm{D}[X_1, \ldots, X_k],$$

and a $\mathcal{P}$-formula Φ whose complexity is bounded by C, and outputs a semi-algebraic basis, $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, of $\mathrm{H}_0(\mathcal{R}(\Phi))$. The complexity of each $\mathbf{x}_i$ (as a semi-algebraic subset of R^k defined over D), as well as the complexity of the algorithm, are both bounded by $C^{k^{O(1)}}$.

2. New Results

We now state the new results proved in the paper.

2.1. **Generalization to the first homology group.** The main goal of this paper is to prove an analog of Theorem 1 with the zero-th homology group replaced by the first homology group. The following theorem is the main result of the paper.

Theorem 2. *There exists an algorithm that takes as input a finite set*

$$\mathcal{P} \subset \mathrm{D}[X_1, \ldots, X_k],$$

and a $\mathcal{P}$-closed formula Φ whose complexity is bounded by C, and outputs a finite set $\mathcal{Q} \subset \mathrm{D}[X_1, \ldots, X_k]$, as well as a finite tuple $(\Psi_j)_{j \in J}$, in which each Ψ_j is a $\mathcal{Q}$-formula, such that the realizations $\Gamma_j = \mathcal{R}(\Psi_j)$ have the following properties:

1. *For each $j \in J$, $\Gamma_j \subset S$, where $S = \mathcal{R}(\Phi)$, and Γ_j is semi-algebraically homeomorphic to $\mathbf{S}^1$ $(= \mathbf{S}^1(\mathbf{0}, 1))$;*
2. *the inclusion map $\Gamma_j \hookrightarrow S$ induces an injective map $\mathbb{F} \cong \mathrm{H}_1(\Gamma_j) \to \mathrm{H}_1(S)$, whose image we denote by $[\Gamma_j]$;*
3. *the tuple $([\Gamma_j])_{j \in J}$ forms a basis of $\mathrm{H}_1(S)$.*

The complexity of each $\Gamma_j, j \in J$ (as a semi-algebraic subset of R^k defined over D) is bounded by $C^{O(k^2)}$, and the complexity of this algorithm is bounded by $C^{k^{O(1)}}$.

Remark 2. Observe that each element $[\Gamma_j]$ of the basis of $\mathrm{H}_1(S)$ in Theorem 2 is represented by the one-dimensional semi-algebraic subset $\Gamma_j \subset S$. In the algorithm that we describe later for computing such a basis (Algorithm 7), we use another algorithm (Algorithm 6) with singly exponential complexity for computing a simplicial complex Δ which is homologically 1-equivalent to S (see Definition 9). One can compute an explicit basis of the first homology group of Δ, with elements of the basis represented by 1-cycles of Δ. However, note that such a basis will not produce semi-algebraic curves in S representing a basis of $\mathrm{H}_1(S)$. This is because there is no natural map between $|\Delta|$ and S (see Definition 9).

Remark 3. We also note that the exponent $O(1)$ in the complexity of the algorithm comes from the complexity of Algorithm 6 applied in the special case of computing a simplicial replacement of a diagram of semi-algebraic sets and inclusion maps which is homologically 1-equivalent to the input. In general, this algorithm computes a simplicial replacement which is homologically ℓ-equivalent to the given diagram for some fixed $\ell \geq 0$, and the exponent in that case is $O(\ell)$ (we use it in the special case with $\ell = 1$).

2.2. **Connections with the "basic lemma" in complex algebraic geometry.** As mentioned previously, designing an efficient algorithm (i.e. with singly exponential complexity) for computing semi-algebraic triangulations of semi-algebraic sets is one of the most important open problems in algorithmic semi-algebraic geometry. In the absence of such an algorithm, several low dimensional work-arounds have been designed to compute important topological invariants without having to compute a full triangulation. For example, there are algorithms with singly exponential complexity for computing the first ℓ Betti numbers (for any fixed $\ell \geq 0$) which do not use triangulations [2]. More recently, in [9], the authors give an algorithm with singly exponential complexity for computing a simplicial complex which is homologically ℓ-equivalent (see Definition 9 below) to a given closed semi-algebraic set (for any fixed $\ell \geq 0$).

Over the complex numbers, algebraic varieties cannot be triangulated (in the usual topological sense) using algebraic sets and maps. The following result (see for example [26] where it is called the "Basic Lemma") serves as a substitute and can be considered a weak analog in complex algebraic geometry of the property that real semi-algebraic sets can be triangulated using semi-algebraic maps.

Lemma 1 (Basic Lemma - first form [26])**.** *Let K be a subfield of* C*. Let W be Zariski closed in an affine variety X defined over K. Assume* $\dim_{\mathrm{C}} W < \dim_{\mathrm{C}} X$*. Then, there is a Zariski closed Z in X so that* $\dim_{\mathrm{C}} Z < \dim_{\mathrm{C}} X$ *with $W \subset Z$, and*

$$\mathrm{H}_q(X, Z) = 0,$$

whenever $q \neq \dim_{\mathrm{C}} X$*.*

Remark 4. Note also that by applying Lemma 1 repeatedly to an affine variety X of dimension n, one obtains Zariski closed subsets of X,

$$Z^{(n-1)} \supset \cdots Z^{(i)} \supset \cdots \supset Z^{(1)} \supset Z^{(0)}$$

with $\dim Z^{(i)} \leq i$, and $\mathrm{H}_q(X, Z^{(i)}) = 0$ for $0 \leq q \leq i$. (This follows from applying the basic lemma with $W = \emptyset$, first to X to obtain $Z^{(n-1)}$, and then to $Z^{(n-1)}$ to obtain $Z^{(n-2)}$ and so on, and the homology long exact sequence of the various triples $(X, Z^{(i)}, Z^{(i-1)})$.)

One could ask for a semi-algebraic version of the basic lemma, where W, X, Z are closed semi-algebraic subsets (or even Zariski closed subsets) of R^k, where R is a real closed field. The proof of the basic lemma (actually of a stronger version) given in [26] is valid over real closed fields. However, the proof is not effective in the sense that no bound on the complexity of Z is given (in terms of the complexities of X and W). However, since the proof depends on iterated projections, the complexity of the construction is likely to be doubly exponential.

Notice that in the semi-algebraic case (taking $W = \emptyset$), one can take $Z = Z^{(i)} = h(|\mathrm{sk}_i(K)|)$, where $h : |K| \to X$ is a semi-algebraic triangulation of X, and $\mathrm{sk}_q(K)$ denotes the q-dimensional skeleton of K. The sequence of semi-algebraic subsets $Z^{(\dim X-1)} \supset \cdots \supset Z^{(0)}$ would then satisfy the properties of Remark 4. However, it is clear that the best complexity one can obtain in this way is doubly exponential. The existence of the sequence $Z^{(i)}$ over algebraically closed fields (cf. Remark 4) inspires the following the question.

Is it possible to prove in the semi-algebraic case the existence of a similar sequence satisfying the homological property in Remark 4, but where the the subsets $Z^{(i)}$ do not correspond to skeleta of some triangulation – and hence could be potentially of smaller complexity ?

In fact, it makes sense to ask for the existence of the sequence $Z^{(i)}$ whose complexity is graded in terms of i i.e. $Z^{(0)}$ has the smallest complexity, followed by $Z^{(1)}$, and so on. We formulate below a quantitative conjecture which is a version of the basic lemma in the semi-algebraic case as follows.

Conjecture 1. *Let $X \subset \mathrm{R}^k$ be a semi-algebraic set defined by a closed formula of complexity bounded by C, and let $\dim X = n$. Then there exists closed semi-algebraic subsets of X,*

$$Z^{(n-1)} \supset \cdots \supset Z^{(1)} \supset Z^{(0)}$$

with $\dim Z^{(i)} \leq i$, and $\mathrm{H}_q(X, Z^{(i)}) = 0$ for $0 \leq q \leq i$, such that for each $i, 0 \leq i \leq n-1$, the complexity of $Z^{(i)}$ is bounded by $C^{O(k^{i+1})}$.

Moreover, there exists an algorithm for computing closed formulas describing $Z^{(0)}, \ldots, Z^{(\ell)}, 0 \leq \ell \leq n-1$, whose complexity is bounded by $C^{k^{O(\ell)}}$.

Remark 5. Conjecture 1 is especially interesting because of the following observation. The standard algorithms for triangulating semi-algebraic sets using cylindrical algebraic decomposition (see for example [6, Chapter 5]) can be modified so that their complexities are bounded doubly exponentially only in the dimension of the given semi-algebraic set rather than that of the ambient space. Hence if Conjecture 1 is true, then it will provide an alternative approach (compared to [2, 9]) towards the problem of computing the first ℓ Betti numbers of any given semi-algebraic set with singly exponential complexity for each fixed ℓ. In this paper, we take an initial step towards verifying the conjecture (see Theorem 3). We believe that the same inductive approach used in the proof of Proposition 3 can be generalized to handle the full conjecture.

We prove the following.

Theorem 3. *With the same notation as in Conjecture 1, there exists closed semi-algebraic subsets of X,*

$$Z^{(1)} \supset Z^{(0)}$$

with $\dim Z^{(i)} \leq i$, *and* $\mathrm{H}_q(X, Z^{(i)}) = 0$ *for* $0 \leq q \leq i, i = 0, 1$, *such that the complexities of* $Z^{(0)}, Z^{(1)}$ *are bounded by* $C^{O(k^2)}$.

Moreover, there exists an algorithm for computing closed formulas describing $Z^{(0)}, Z^{(1)}$, *whose complexity is bounded by* $C^{k^{O(1)}}$.

2.3. **Comparison with roadmaps.** One of our intermediate constructions, namely the semi-algebraic subset $Z^{(1)}$ in Theorem 3 (also the set Γ constructed in Algorithm 4) is reminiscent of roadmaps of semi-algebraic sets (mentioned earlier), and their construction is somewhat similar to the (classical) construction of roadmaps [3]. We describe here the key difference between our construction and the classical construction of roadmaps in Section 4.1 where we give an outline of Algorithm 4. In this section, we recall the defining property of roadmaps of semi-algebraic sets and indicate why a roadmap is not sufficient for the purposes of the current paper.

The following definition is taken from [6, Chapter 15]. Let $S \subset \mathrm{R}^k$ be a semi-algebraic set. We denote by $\pi_1 : \mathrm{R}^k \to \mathrm{R}$ the projection to the first coordinate and denote for $x \in \mathrm{R}$, $S_x = S \cap \pi_1^{-1}(x)$ (cf. Notation 6 below).

Definition 3. A semi-algebraic subset $\Gamma \subset S$ is called a roadmap of S if it satisfies the following properties:

RM$_0$. $\dim \Gamma \leq 1$;

RM$_1$. for every semi-algebraically connected component C of S, $C \cap \Gamma$ is semi-algebraically connected;

RM$_2$. for every $x \in \mathrm{R}$ and for every semi-algebraically connected component D of S_x, $D \cap \Gamma \neq \emptyset$.

Remark 6. We note that the design of efficient algorithms for construction of roadmaps of semi-algebraic sets has a long history [17, 21, 22, 24, 4]. The algorithm with the best complexity can be found in [4]. The complexity of this algorithm is bounded by $s^{k'+1} d^{O(k^2)}$, where s is the number of polynomials used to define the given set, d a bound on their degrees, k the dimension of the ambient space, and k' is the dimension of a real variety containing the given set [4]. More recently, the dependence on k in the exponent has been further improved using new methods [7, 12], but for the moment these new algorithms work only for algebraic (rather than semi-algebraic) sets.

Notice that if S is a closed semi-algebraic set and $\Gamma \subset S$ is a roadmap of S, then the homomorphism $i_{*,1} : \mathrm{H}_1(\Gamma) \to \mathrm{H}_1(S)$ need not be surjective. A simple example of this phenomenon is provided by a torus $T \subset \mathrm{R}^3$ as depicted in Figure 1. The classical construction of roadmap (see for example [6, Chapter 15]) produces the one dimensional semi-algebraic subset Γ

[3] More modern algorithms such as those described in [7, 12] use different techniques.

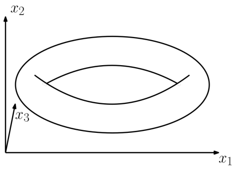

FIGURE 1. figure $T \subset \mathrm{R}^3$

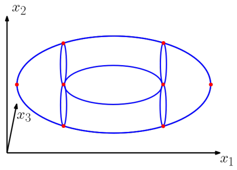

FIGURE 2. figure $\Gamma \subset T$

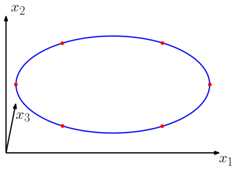

FIGURE 3. figure $\Gamma' \subset T$

depicted in Figure 2. In this example, the set Γ does satisfy the property that $\mathrm{H}_q(T, \Gamma) = 0, q = 0, 1$, though this is not ensured in general by the classical roadmap algorithm. However, notice that if we take the subset $\Gamma' \subset \Gamma$ consisting only of the larger horizontal circle as depicted in the Figure 3, then Γ' does satisfy the property of being a roadmap of T (cf. Definition 3), but $\mathrm{H}_1(T, \Gamma') \neq 0$ (in fact, $\dim \mathrm{H}_1(T, \Gamma') = 1$).

The rest of the paper is devoted to proving Theorems 2 and 3. In Section 3, we discuss some preliminaries including definitions of representations of real algebraic numbers, points, and curve segments that we use in our algorithms. In Section 4, we prove Theorems 2 and 3.

3. Preliminaries

In this section we introduce some notation and describe how we algebraically represent points and curve segments. We first introduce some notation.

3.1. Some notation.

Notation 4 (Sign conditions)**.** For a finite set $\mathcal{P} \subset \mathrm{R}[X_1, \ldots, X_k]$ we will call any element of $\{0, 1, -1\}^{\mathcal{P}}$ a *sign condition on* $\mathcal{P}$. For a semi-algebraic subset $Z \subset \mathrm{R}^k$, we will denote by

$$\mathcal{R}(\sigma) = \{\mathbf{x} \in Z \mid \mathbf{sign}(P(\mathbf{x})) = \sigma(P), P \in \mathcal{P}\}.$$

Given a sign condition $\sigma \in \{0, 1, -1\}^{\mathcal{P}}$, we will denote by $\overline{\sigma}$ the formula defined by

$$\overline{\sigma} = \bigwedge_{\sigma(P)=0} (P = 0) \wedge \bigwedge_{\sigma(P)=1} (P \geq 0) \wedge \bigwedge_{\sigma(P)=-1} (P \leq 0),$$

and call $\overline{\sigma}$ the *weak sign condition associated to* σ.

Notation 5 (Closure)**.** For any semi-algebraic subset $S \subset \mathrm{R}^k$, we will denote by $\mathrm{clos}(S)$ the closure of S (in the euclidean topology). It is a consequence of the fact that the first-order theory of real closed fields admits quantifier-elimination that $\mathrm{clos}(S)$ is again a semi-algebraic subset of R^k.

Notation 6 (Projections to coordinate subspaces)**.** We will denote by $\pi_i : \mathrm{R}^k \to \mathrm{R}$, the projection on to the i-th coordinate. More generally, for a subset $J \subset \{1,\ldots,k\}$, we denote by $\pi_J : \mathrm{R}^k \to \mathrm{R}^J$, the projection on the coordinates indexed by J. In particular, $\pi_{[1,i]} : \mathrm{R}^k \to \mathrm{R}^{[1,i]}$ will denote the projection onto the first i coordinates.

For $1 \leq i \leq k$, $S \subset \mathrm{R}^k$, $J \subset \{1,\ldots,k\}$, and $Z \subset \mathrm{R}^J$, we will denote by $S_Z = \pi_J^{-1}(Z) \cap S$. If $Z = \{\mathbf{z}\} \subset \mathrm{R}^J$, then we will write $S_{\mathbf{z}}$ instead of $S_{\{\mathbf{z}\}}$.

3.2. **Representations of points and curves.** While the algorithms that we describe have certain geometric underpinnings, it is important to remember that the points and curve segments that we compute need to be represented algebraically, and hence we need to specify the precise representations that we use.

Moreover, we often fix a set of coordinates (say $(X_1,\ldots,X_i)$), to $\mathbf{t} = (t_1,\ldots,t_i)$ and call an algorithm recursively in the fiber $\{\mathbf{t}\} \times \mathrm{R}^{k-i}$. This necessitates the introduction of triangular Thom encodings (which fixes points), and all our representations of points and curve segments are introduced relative to such triangular Thom encodings.

3.2.1. *Thom encoding.* The following definitions are adapted from [6].

We begin with the representations of elements of R (which are algebraic over D) as roots of polynomials in $\mathrm{D}[X]$ with a given Thom encoding (cf. Definition 4 below).

Definition 4 (Thom encoding, associated element of R)**.** For $P \in \mathrm{R}[X]$ we will denote by

$$\mathrm{Der}(P) = \left(P, P', \ldots, P^{(\deg(P))}\right)$$

the list of derivatives of P.

We will call a pair $\tau = (P, \sigma)$ with $\sigma \in \{0,1,-1\}^{\mathrm{Der}(P)}$, the *Thom encoding of* $x \in \mathrm{R}$ [4], if $\sigma(P) = 0$ and $\sigma(P^{(i)}) = \mathbf{sign}(P^{(i)}(x))$ for $0 \leq i \leq \deg(P)$.

We will sometimes abuse notation and sometime call σ *the Thom encoding of the root x of P*. We will denote x by $\mathrm{ass}(\tau)$, and call $\mathrm{ass}(\tau)$ *the element of* R *associated to τ*.

We will call $\deg(P)$ to be *the degree of the Thom encoding τ*, and denote it by $\deg(\tau)$.

As remarked before we will often need to fix a block of variables $(X_1,\ldots,X_i)$ to $\mathbf{t} = (t_1,\ldots,t_i) \in \mathrm{R}^i$, and perform arithmetic operations in the ring $\mathrm{D}[\mathbf{t}]$. For this purpose we introduce the notion of a triangular Thom encoding whose associated point is an element of R^i.

Definition 5 (Triangular Thom encoding and associated point)**.** A *triangular Thom encoding* $\mathcal{T} = (\mathbf{F}, \boldsymbol{\sigma})$ of size i is a tuple (triangular system) of polynomials,

$$\mathbf{F} = (f_1,\ldots,f_i)$$

[4] It is a consequence of the well-known Thom's lemma, that the Thom encoding uniquely characterizes a root in R of a polynomial in $\mathrm{D}[X]$ (see for example, [6, Proposition 2.27]).

where $f_j \in \mathrm{R}[X_1, \ldots, X_j], 1 \leq j \leq i$, and a tuple of Thom encodings $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_i)$, with $\sigma_j \in \{0, 1, -1\}^{\mathrm{Der}_{X_j}(f_j)}$, such that for each $j, 1 \leq j \leq i$, there exists $t_j \in \mathrm{R}$, such that t_j is a root of the polynomial $f_j(t_1, \ldots, t_{j-1}, T_j)$ with Thom encoding σ_j. We call $(t_1, \ldots, t_i) \in \mathrm{R}^i$ the *point associated to* $\mathcal{T}$ and denote $\mathrm{ass}(\mathcal{T}) = (t_1, \ldots, t_i)$.

Given a triangular Thom encoding

$$\mathcal{T}^+ = ((f_1, \ldots, f_{i+1}), (\sigma_1, \ldots, \sigma_{i+1})),$$

with $\mathrm{ass}(\mathcal{T}^+) = (t_1, \ldots, t_{i+1})$, we will sometimes call the pair $\tau = (f_{i+1}, \sigma_{i+1})$ a *Thom encoding over the triangular Thom encoding*

$$\mathcal{T} = ((f_1, \ldots, f_i), (\sigma_1, \ldots, \sigma_i)).$$

In this case we will denote t_{i+1} by $\mathrm{ass}(\tau)$ (generalizing Definition 4).

We will call $\max_{1 \leq j \leq i} \deg(F_j)$ the degree of the triangular Thom encoding $\mathcal{T}$, and denote it by $\deg(\mathcal{T})$.

If $\tau = (f_{i+1}, \sigma_{i+1})$ is a Thom encoding over a triangular Thom encoding

$$\mathcal{T} = ((f_1, \ldots, f_i), (\sigma_1, \ldots, \sigma_i)),$$

we will call $\deg_{T_{i+1}}(f_{i+1})$, the degree of τ, and denote it by $\deg(\tau)$.

Finally, given a triangular Thom encoding $\mathcal{T} = (\mathbf{F}, \boldsymbol{\sigma})$ of size i, we denote by $\theta(\mathcal{T})$, the formula

$$\bigwedge_{1 \leq j \leq i} \quad \bigwedge_{0 \leq h \leq \deg_{X_j}(f_j)} \left(\mathbf{sign}(f_j^{(h)}) = \sigma_j(f_j^{(h)}) \right).$$

Notation 7. Given a triangular Thom encoding $\mathcal{T} = (\mathbf{F}, \boldsymbol{\sigma})$ of size i (following the same notation as in Definition 5 above), we will denote by $\overline{\boldsymbol{\sigma}}$, the closed formula obtained from $\boldsymbol{\sigma}$ by replacing each sign condition on the derivatives by the corresponding weak inequality (i.e. replacing $f_j^{(h)} > 0$ by $f_j^{(h)} \geq 0$ and $f_j^{(h)} < 0$ by $f_j^{(h)} \leq 0$).

It is a consequence of Thom's lemma [6] that:

Lemma 2. *Given a triangular Thom encoding* $\mathcal{T} = (\mathbf{F}, \boldsymbol{\sigma})$ *of size* i,

$$\mathrm{ass}(\mathcal{T}) = \mathcal{R}(\overline{\boldsymbol{\sigma}}, \mathrm{R}^i).$$

Proof. Follows from [6, Proposition 5.39] (Generalized Thom's Lemma). □

3.2.2. *Real univariate representations.* We will represent points in R^k using real univariate representations [6, pp. 465] defined below. As explained previously, we need to define this notion with a block of variables fixed by a triangular Thom encoding. In this case the first i coordinates of the point are fixed by a triangular Thom encoding, and the real univariate representation specifies the remaining $k - i$ coordinates.

Definition 6 (Real univariate representations over a triangular Thom encoding and associated point)**.** Let $\mathcal{T} = (\mathbf{F}, \boldsymbol{\sigma})$ be a triangular Thom encoding of size $i, 0 \leq i < k$.

A *real univariate representation* u *in* R^k *over* $\mathcal{T}$ is a pair (F, σ) where $F = (f, g_0, g_{i+1}, \ldots, g_k) \in \mathrm{R}[X_1, \ldots, X_i, T]$, with $f(\mathrm{ass}(\mathcal{T}), T), g_0(\mathrm{ass}(\mathcal{T}), T)$ co-prime, and σ a Thom encoding of a real root of $f(\mathrm{ass}(\mathcal{T}), T) \in \mathrm{R}[T]$.

We denote by $\mathrm{ass}(u) \in \mathrm{R}^k$ the point

$$\left(\mathrm{ass}(\mathcal{T}), \frac{g_{i+1}(\mathrm{ass}(\mathcal{T}), \mathrm{ass}(\tau))}{g_0(\mathrm{ass}(\mathcal{T}), \mathrm{ass}(\tau))}, \ldots, \frac{g_k(\mathrm{ass}(\mathcal{T}), \mathrm{ass}(\tau))}{g_0(\mathrm{ass}(\mathcal{T}), \mathrm{ass}(\tau))}\right) \in \mathrm{R}^k, \tag{3.1}$$

where $\tau = (f, \sigma)$ (notice that τ is a Thom encoding over $\mathcal{T}$), and call $\mathrm{ass}(u)$ *the point associated to* u.

We will call the pair (D_1, D_2), denoted $\deg(u)$, the degree of u, where $D_1 = \deg(\mathcal{T})$ and D_2 is the maximum of $\deg_T(f), \deg_T(g_0), \ldots, \deg_T(g_k)$.

3.2.3. *Curve segments.* We describe semi-algebraic curve segments by real univariate representations parametrized by one of the coordinates. As before the following definition assumes a triangular Thom encoding fixing the first i coordinates.

Definition 7 (Curve segment representation over a triangular Thom encoding)**.** Let $\mathcal{T} = (\mathbf{F}, \boldsymbol{\sigma})$ be a triangular Thom encoding of size $i, 0 \leq i \leq k-1$. A *curve segment representation* γ *above* $\mathcal{T}$ consists of:

(a) Thom encodings, $\tau_1 = \tau_1(\gamma), \tau_2 = \tau_2(\gamma)$, over $\mathcal{T}$, with $\mathrm{ass}(\tau_1) < \mathrm{ass}(\tau_2)$;

(b) a pair $(u, \rho) = (u(\gamma), \rho(\gamma))$, where

$$u = (f, g_0, g_{i+2}, \ldots, g_k) \in \mathrm{R}[X_1, \ldots, X_{i+1}, T]^{k-i+1},$$

and

$$\rho \in \{0, 1, -1\}^{\mathrm{Der}_T(f)},$$

such that for every $x_{i+1} \in (\mathrm{ass}(\tau_1), \mathrm{ass}(\tau_2))$ there exists a real root $t(x_{i+1})$ of $f(\mathrm{ass}(\mathcal{T}), x_{i+1}, T)$ with Thom encoding ρ and

$$f(\mathrm{ass}(\mathcal{T}), x_{i+1}, T), g_0(\mathrm{ass}(\mathcal{T}), x_{i+1}, T)$$

co-prime.

We will call (τ_1, τ_2) the *Thom encoding of the interval of definition of* γ.

The semi-algebraic function h which maps $x_{i+1} \in (\mathrm{ass}(\tau_1), \mathrm{ass}(\tau_2))$ to the point of R^k defined by

$$h(x_{i+1}) =$$
$$\left(\mathrm{ass}(\mathcal{T}), x_{i+1}, \frac{g_{i+2}(\mathrm{ass}(\mathcal{T}), x_{i+1}, t(x_{i+1}))}{g_0(\mathrm{ass}(\mathcal{T}), x_{i+1}, t(x_{i+1}))}, \ldots, \frac{g_k(\mathrm{ass}(\mathcal{T}), x_{i+1}, t(x_{i+1}))}{g_0(\mathrm{ass}(\mathcal{T}), x_{i+1}, t(x_{i+1}))}\right),$$

is a continuous injective semi-algebraic function, and we will denote its image by $\mathrm{ass}(\gamma)$.

We will also call $\lim_{x_{i+1} \to \mathrm{ass}(\tau_1)+} h(x_{i+1})$ (resp. $\lim_{x_{i+1} \to \mathrm{ass}(\tau_2)-} h(x_{i+1})$) (if it is defined) the *left end-point* (resp. *right end-point*) of γ.

We will call the pair (D_1, D_2), denoted $\deg(\gamma)$, the degree of γ, where

$$D_1 = \deg(\mathcal{T}),$$

and D_2 is the maximum of $\deg(\tau_1), \deg(\tau_2)$, and

$$\deg_{X_{i+1},T}(f), \deg_{X_{i+1},T}(g_0), \deg_{X_{i+1},T}(g_{i+2}), \ldots, \deg_{X_{i+1},T}(g_k).$$

Remark 7. Note that in Definition 7, $\mathrm{ass}(\gamma)$ is a semi-algebraic set of dimension one, and if it is bounded over R, then its left and right endpoints are well-defined.

Also note that if $i = k - 1$, and τ_1, τ_2 are two Thom encodings over $\mathcal{T}$ with $\mathrm{ass}(\tau_1) < \mathrm{ass}(\tau_2)$, then if γ is the curve segment representation over $\mathcal{T}$ defined by,

$$\begin{array}{rcl} \tau_1(\gamma) & = & \tau_1, \\ \tau_2(\gamma) & = & \tau_2, \\ u(\gamma) & = & ((T, 1), (0, 1)), \end{array}$$

then $\mathrm{ass}(\gamma) = \{\mathrm{ass}(\mathcal{T})\} \times (\mathrm{ass}(\tau_1(\gamma)), \mathrm{ass}(\tau_2(\gamma)))$. Thus, an open interval whose end points are given by Thom encodings can be represented by the curve segment representation given above and we will later use this fact without mention.

4. Proofs of Theorems 2 and 3

4.1. **Outline of the proofs of Theorems 2 and 3.** As in the case of the zero-th homology in Section 1.4 we solve the problem in two steps (cf. Steps 1 and 2 in the solution of Problem 1).

Step 1. In the first step, we develop an algorithm (see Algorithm 4 below) that takes as input a $\mathcal{P}$-closed formula Φ, and produces as output a description of a semi-algebraic subset $\Gamma \subset S = \mathcal{R}(\Phi)$, having dimension ≤ 1, and such that the homomorphism $i_{*,1} : \mathrm{H}_1(\Gamma) \to \mathrm{H}_1(S)$ induced by the inclusion $i : \Gamma \hookrightarrow S$ is surjective, and the homomorphism $i_{*,0} : \mathrm{H}_0(\Gamma) \to \mathrm{H}_0(S)$ is an isomorphism. This is equivalent to $\mathrm{H}_q(S, \Gamma) = 0$ for $q = 0, 1$.

This is the analog of Step 1 in the outline for the solution of Problem 1 above. The main idea behind the construction of Γ comes from the classical construction of a roadmap of a semi-algebraic set S that goes as follows (see also the description in [6, Chapter 15].

Step 1(a) We first compute curves parametrized by the first coordinate, such that the set of curves so constructed meets, for every $x \in \mathrm{R}$, each semi-algebraically connected component of S_x.

Step 1(b) The algorithm is then called recursively on a certain finite set of "slices", namely on sets of the form S_x, where $x \in \mathrm{R}$ varies over a certain finite set of *distinguished values*.

The set of distinguished values includes values at which the connectivity of the set $S_{\leq x}$ changes. Algorithm 4 follows a similar paradigm. The main *new feature* is that the set of distinguished values at which the algorithm makes recursive calls satisfies a stronger property than in the case of roadmap algorithms (cf. Proposition 2 below). This stronger property of the distinguished values allows us to prove inductively (using a little homological algebra via the Five Lemma [18, Proposition 1.1]), the *surjectivity* of the map $i_{*,1} : \mathrm{H}_1(\Gamma) \to \mathrm{H}_1(S)$. The proof of the *isomorphism* $i_{*,0} : \mathrm{H}_0(\Gamma) \to \mathrm{H}_0(S)$ is the same as in the classical construction of a roadmap of a semi-algebraic set. It follows from the exact homology sequence of the pair (S, Γ), that $\mathrm{H}_q(S, \Gamma) = 0$, for $0 \leq q \leq 1$. The semi-algebraic set Γ is initially described as a union of points and curve segments (cf. Definitions 6 and 7)). Converting this description into an equivalent *closed* formula of complexity bounded singly exponentially (cf. Algorithm 5 (Conversion of curve segment representations to closed formulas)) yields a proof of Theorem 3.

Step 2. In this step we use the fact that the semi-algebraic set Γ is semi-algebraically homeomorphic to the geometric realization $|G|$ of a finite graph G having singly exponential size, and it is a relatively easy combinatorial task to choose a basis of simple cycles, $\Gamma_1, \ldots, \Gamma_N$, for the cycle space of G. The images $[\Gamma_1], \ldots, [\Gamma_N]$ (here $[\Gamma_i]$ denotes the image of $\mathrm{H}_1(|\Gamma_i|)$ in $\mathrm{H}_1(S)$ under the homomorphism induced by the inclusion $|\Gamma_i| \hookrightarrow S$), span $\mathrm{H}_1(S)$ but are not necessarily linearly independent. We need to select a minimal spanning subset from amongst the $[\Gamma_1], \ldots, [\Gamma_N]$. For this purpose we use an algorithm for replacing a given semi-algebraic set and a tuple of subsets by a simplicial complex and a tuple of corresponding subcomplexes, which are homologically ℓ-equivalent (cf. Definition 9) for any fixed ℓ, and which has singly exponentially bounded complexity [9] (cf. Algorithm 6 below). We use this algorithm in the case $\ell = 1$. This is analogous to the usage of a roadmap algorithm for overcoming the corresponding obstacle in the case of the zero-th homology (cf. Step 2 in the outline for the solution of Problem 1).

Remark 8. We note here that while it is true that set Γ and its subsets Γ_i are one-dimensional semi-algebraic sets, in order to identify a basis amongst the $[\Gamma_i]$ of $\mathrm{H}_1(S)$, one needs to call Algorithm 6 with input which includes the formula defining S (i.e. the computation involves the topology of S itself and not just the one dimensional subsets Γ_i). The inclusion map $\mathrm{H}_1(\Gamma) \to \mathrm{H}_1(S)$ while being surjective need not be injective, and hence one cannot hope to compute a basis of $\mathrm{H}_1(S)$ by just computing a basis of $\mathrm{H}_1(\Gamma)$.

Example 1. It might be helpful to consider the following concrete examples. In each of the two examples the semi-algebraic set S is a real algebraic surface in R^3 shown in red. In the first example, S is the unit sphere in R^3 and in

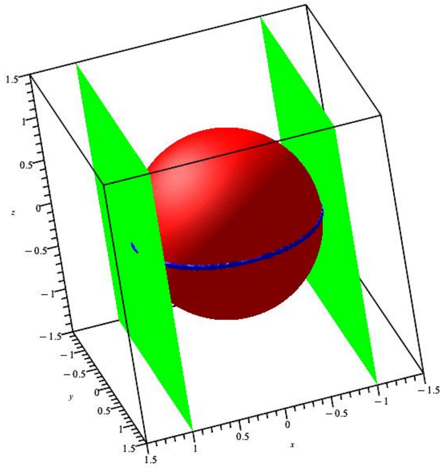

FIGURE 4. Sphere defined by $X_1^2 + X_2^2 + X_3^2 - 1 = 0$

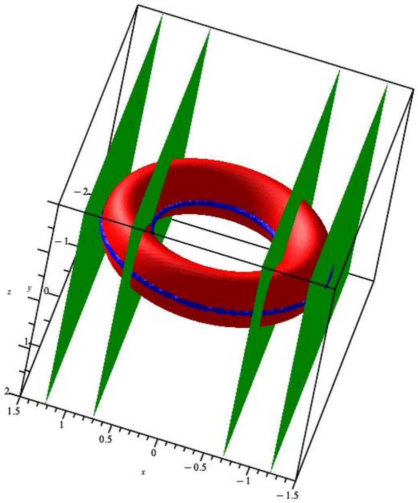

FIGURE 5. Torus defined by $(X_1^2 + X_2^2 - 1)^2 + X_3^2 - 0.25 = 0$

the second example $S \subset \mathrm{R}^3$ be topologically a torus defined by the equation $(X_1^2 + X_2^2 - 1)^2 + X_3^2 - 0.25$.

The semi-algebraic curves computed in Step 1a are shown in blue, and the slices where the recursive calls are made in Step 1b shown in green. In these examples the subset Γ is the union of the blue curves with the intersections of the green planes with S.

In the case of the sphere $\mathrm{H}_1(S) = 0$, while $\mathrm{H}_1(\Gamma) \cong \mathbb{F}$. In the case of the torus, $\mathrm{H}_1(S) \cong \mathbb{F}^2$, while $\mathrm{H}_1(\Gamma) \cong \mathbb{F}^8$, and the inclusion map induces a surjection $\mathrm{H}_1(\Gamma) \to \mathrm{H}_1(S)$ in both cases.

We now describe in detail the two steps outlined above.

4.2. **Implementing Step 1: computing surjection.** In this section we describe an algorithm which will accomplish Step 1 of the two-step algorithm sketched out in Section 4.1. Recall that the goal of this step is to obtain an algorithm with singly exponential complexity that computes a description of a semi-algebraic subset Γ having dimension at most one, such that $\mathrm{H}_q(S, \Gamma) = 0, q = 0, 1$.

As mentioned previously, we will follow the same approach of constructing a roadmap of S (see for instance [6, Chapters 15 and 16]), however with one key additional property.

4.2.1. *Morse-type partition.* In the (classical) construction of the roadmap, one makes recursive calls to the roadmap constructing algorithm at certain special fibers where the X_1-coordinate is fixed to certain values (called “distinguished values” in [6]). These distinguished values include the values c of the X_1-coordinate where the connectivity of the fiber S_c can change. This is sufficient for construction of roadmaps, since the main property of the roadmap is related to connectivity (the intersection of the roadmap with any semi-algebraically connected component of S should be semi-algebraically connected). Since in this paper we are mainly concerned with the first homology group, we need our set of distinguished values to satisfy a more stringent property. Fortunately, there exists a singly exponential complexity algorithm for this purpose which we are going to utilize.

The following algorithm (without a triangular Thom encoding in the input) appears in [10, Algorithm 3]. However, as remarked before in our applications, it will be necessary to fix a block of i variables by a triangular Thom encoding $\mathcal{T}$, and perform the computations over the ring $\mathrm{D}[\mathrm{ass}(\mathcal{T})]$. Each arithmetic operation in the ring $\mathrm{D}[\mathrm{ass}(\mathcal{T})]$ costs $D^{O(i)}$ arithmetic operations in D, where $D = \deg(\mathcal{T})$.

Algorithm 1 (Morse partition)

Input:

(a) $r = \frac{a}{b}, a, b \in \mathrm{D}, a, b > 0$;
(b) A triangular Thom encoding $\mathcal{T}$ of size $i, 0 \leq i \leq k$;
(c) A finite set $\mathcal{P} = \{P_1, \ldots, P_s\} \subset \mathrm{D}[X_1, \ldots, X_k]$;
(d) A $\mathcal{P}$-closed formula Φ.

Output:

An ordered tuple $\mathcal{F} = (\tau_1, \ldots, \tau_N)$ of Thom encodings over $\mathcal{T}$, with associated points $t_1 < \cdots < t_N$, with $-r \leq t_1, t_N \leq r$, and such that for each $j, 1 \leq j \leq N-1$, and all $t \in [t_j, t_{j+1})$ the inclusion maps

$$S_{\{\mathrm{ass}(\mathcal{T})\} \times (-\infty, t_j]} \hookrightarrow S_{\{\mathrm{ass}(\mathcal{T})\} \times (\infty, t]}$$

(resp. $S_{\{\mathrm{ass}(\mathcal{T})\} \times [t_{j+1}, \infty)} \hookrightarrow S_{\{\mathrm{ass}(\mathcal{T})\} \times [t, \infty)}$) induce isomorphisms

$$\mathrm{H}_*(S_{\{\mathrm{ass}(\mathcal{T})\} \times (\infty, t_j]}) \to \mathrm{H}_*(S_{\{\mathrm{ass}(\mathcal{T})\} \times (\infty, t]})$$

(resp. $\mathrm{H}_*(S_{\{\mathrm{ass}(\mathcal{T})\} \times [t_{j+1}, \infty)}) \to \mathrm{H}_*(S_{\{\mathrm{ass}(\mathcal{T})\} \times [t, \infty)})$), where

$$S = \mathcal{R}(\Phi) \cap \mathrm{clos}(B_k(\mathbf{0}, r)).$$

Procedure:

```
1: Call [10, Algorithm 3] (up to Step 13 and doing all computations in the ring $\mathrm{D}[\mathrm{ass}(\mathcal{T})]$) twice with inputs, $(-, r, \mathcal{P}, \Phi, X_{i+1})$ and $(-, r, \mathcal{P}, \Phi, -X_{i+1})$, and let $\mathcal{F}_1, \mathcal{F}_2$ be the set of Thom encodings over $\mathcal{T}$ output (Part (a) of the output of [10, Algorithm 3]).
2: Using Algorithm 12.21 (Triangular Comparison of Roots) in [6] order the Thom encodings in $\mathcal{F}_1 \cup \mathcal{F}_2$, and merge the two sets into one ordered tuple $\mathcal{F} = (\tau_1, \ldots, \tau_N)$ such that
       $\mathrm{ass}(\tau_1) < \cdots < \mathrm{ass}(\tau_N).$
3: Output $\mathcal{F}$.
```

Complexity: The complexity of the algorithm is bounded by

$$D^{O(i)}(sd)^{O(k)},$$

where $s = \mathrm{card}(\mathcal{P})$, $d = \max_{P \in \mathcal{P}} \deg(P)$, and $D = \deg(\mathcal{T})$. Moreover, $\deg(\tau_i) \leq d^{O(k)}$ for $1 \leq i \leq N$, and the size of $\mathcal{F}$ is bounded by $(sd)^{O(k)}$.

We will need the following extra property of the output of Algorithm 1.

Proposition 1. *For each $j, 1 \leq j \leq N-1$, and all $t \in [t_j, t_{j+1}]$ the inclusion maps*

$$S_{\{\mathrm{ass}(\mathcal{T})\} \times \{t\}} \hookrightarrow S_{\{\mathrm{ass}(\mathcal{T})\} \times [t_j, t_{j+1}]}$$

induce isomorphisms

$$\mathrm{H}_*(S_{\{\mathrm{ass}(\mathcal{T})\} \times \{t\}}) \to \mathrm{H}_*(S_{\{\mathrm{ass}(\mathcal{T})\} \times [t_j, t_{j+1}]}),$$

where

$$S = \mathcal{R}(\Phi) \cap \mathrm{clos}(B_k(\mathbf{0}, r)).$$

Proof. Let

$$\begin{aligned} A_1 &= S_{\{\mathrm{ass}(\mathcal{T})\}\times(-\infty,t]}, \\ A_2 &= S_{\{\mathrm{ass}(\mathcal{T})\}\times[t,\infty)}, \\ B_1 &= S_{\{\mathrm{ass}(\mathcal{T})\}\times(-\infty,t_{j+1}]}, \\ B_2 &= S_{\{\mathrm{ass}(\mathcal{T})\}\times[t_j,\infty)}. \end{aligned}$$

Then, $A_h \subset B_h, h = 1,2$ and

$$\begin{aligned} A_1 \cap A_2 &= S_{\{\mathrm{ass}(\mathcal{T})\}\times\{t\}}, \\ B_1 \cap B_2 &= S_{\{\mathrm{ass}(\mathcal{T})\}\times[t_j,t_{j+1}]}, \end{aligned}$$

and

$$A_1 \cup A_2 = B_1 \cup B_2 = S_{\mathrm{ass}(\mathcal{T})}.$$

Moreover the properties of the output of Algorithm 1 imply that the homomorphisms $\mathrm{H}_*(A_h) \to \mathrm{H}_*(B_h), h = 1,2$ induced by inclusions are isomorphisms. The Mayer-Vietoris exact sequence in homology (see for example, [6, Theorem 6.35]) then yields the following commutative diagram with exact rows and vertical homomorphisms induced by inclusion (where A_{12} (resp. B_{12}) denotes $A_1 \cap A_2$ (resp. $B_1 \cap B_2$), and A^{12} (resp. B^{12}) denotes $A_1 \cup A_2$ (resp. $B_1 \cup B_2$)):

$$\begin{tikzcd}
\mathrm{H}_{m+1}(A_1) \oplus \mathrm{H}_{m+1}(A_2) \arrow[r] \arrow[d] & \mathrm{H}_{m+1}(A^{12}) \arrow[r] \arrow[d] & \mathrm{H}_m(A_{12}) \arrow[r] \arrow[d] & \mathrm{H}_m(A_1) \oplus \mathrm{H}_m(A_2) \arrow[r] \arrow[d] & \mathrm{H}_m(A^{12}) \arrow[d] \\
\mathrm{H}_{m+1}(B_1) \oplus \mathrm{H}_{m+1}(B_2) \arrow[r] & \mathrm{H}_{m+1}(B^{12}) \arrow[r] & \mathrm{H}_m(B_{12}) \arrow[r] & \mathrm{H}_m(B_1) \oplus \mathrm{H}_m(B_2) \arrow[r] & \mathrm{H}_m(B^{12})
\end{tikzcd}$$

It follows from the fact that the homomorphisms $\mathrm{H}_*(A_h) \to \mathrm{H}_*(B_h), h = 1,2$ induced by inclusions are isomorphisms, and the fact that $A^{12} = B^{12}$, that all the vertical arrows other than the middle one are isomorphisms. Hence, by the Five Lemma (see for example [29, Proposition 2.72]) the middle arrow is also an isomorphism, thus proving the proposition. □

We will also need the following algorithm that takes as input a quantifier-free formula Φ and outputs $r \in \mathrm{D}, r > 0$, such that $\mathcal{R}(\Phi)$ is homologically equivalent to $\mathcal{R}(\Phi) \cap \mathrm{clos}(\mathrm{B}(\mathbf{0}, r))$.

We first introduce some more notation.

Notation 8. For R a real closed field we denote by $\mathrm{R}\langle\varepsilon\rangle$ the real closed field of algebraic Puiseux series in ε with coefficients in R. As a real closed field $\mathrm{R}\langle\varepsilon\rangle$ is uniquely ordered, and this order extends the order on R. It is the unique order in which $\varepsilon > 0$ and $\varepsilon < x$ for every $x \in \mathrm{R}, x > 0$. In particular, the subring $\mathrm{D}[\varepsilon] \subset \mathrm{R}\langle\varepsilon\rangle$ is ordered by: $\sum_{i\geq 0} a_i\varepsilon^i > 0$ if and only if $a_p > 0$ where $p = \min\{i \mid a_i \neq 0\}$.

Notation 9. If R' is a real closed extension of a real closed field R, and $S \subset \mathrm{R}^k$ is a semi-algebraic set defined by a first-order formula with coefficients in R, then we will denote by $\mathrm{ext}(S, \mathrm{R}') \subset \mathrm{R}'^k$ the semi-algebraic subset of R'^k defined by the same formula [5]. It is well-known that $\mathrm{ext}(S, \mathrm{R}')$ does not depend on the choice of the formula defining S [6, Proposition 2.87].

Notation 10. For $P = a_p T^p + \cdots + a_q T^q, p \geq q \in \mathrm{D}[T]$, $a_p a_q \neq 0$, we denote

$$c'(P) = \left((p+1) \cdot \sum_i \frac{a_i^2}{a_q^2}\right)^{-1}.$$

We will use the following lemma.

Lemma 3. *With the same notation as in Notation 10, if $x \neq 0, x \in \mathrm{R}$ is a root of P, then $|x| > c'(P)$.*

Proof. See Lemma 10.7 in [6]. □

Algorithm 2 (Big enough radius)

```
Input:
    (a) A triangular Thom encoding $\mathcal{T} = (\mathbf{F}, \boldsymbol{\sigma})$ of size $i, 0 \leq i \leq k$;
    (b) a finite set $\mathcal{P} \subset \mathrm{D}[X_1, \ldots, X_k]$;
    (c) a $\mathcal{P}$-closed formula $\Phi$ such that $\mathcal{R}(\Phi)$ is bounded.
Output:
    Elements $a, b \in \mathrm{D}[\mathrm{ass}(\mathcal{T})], a, b > 0$, such that the inclusion map
      $\mathcal{R}(\Phi, \mathrm{R}^k)_{\mathrm{ass}(\mathcal{T})} \cap \mathrm{clos}(\{\mathrm{ass}(\mathcal{T})\} \times B_{k-i}(\mathbf{0}, r)) \hookrightarrow \mathcal{R}(\Phi, \mathrm{R}^k)_{\mathrm{ass}(\mathcal{T})}$,
    where $r = \frac{a}{b}$, induces an isomorphism
    $\mathrm{H}_*\left((\mathcal{R}(\Phi) \cap \mathrm{clos}(\{\mathrm{ass}(\mathcal{T})\} \times B_{k-i}(\mathbf{0}, r)))_{\mathrm{ass}(\mathcal{T})}\right) \to \mathrm{H}_*\left(\mathcal{R}(\Phi)_{\mathrm{ass}(\mathcal{T})}\right)$.

Procedure:
  $P_1 \leftarrow Y - (X_{i+1}^2 + \cdots + X_k^2)$.
  $P_2 \leftarrow (\varepsilon^2(X_{i+1}^2 + \cdots + X_k^2) - 1)$.
  $\widetilde{\Phi} \leftarrow \Phi \wedge (P_1 = 0) \wedge (P_2 \leq 0)$.
  $\widetilde{\mathrm{D}} \leftarrow \mathrm{D}[\varepsilon]$ where $D[\varepsilon] \subset \mathrm{R}\langle\varepsilon\rangle$ (cf. Notation 8).
  Call Algorithm 1 (Morse partition) treating $Y$ as the $(i+1)$-st coordinate,
      with computations occurring in the domain $\widetilde{\mathrm{D}}$, and with input $\mathcal{T}, \mathcal{P} \cup \{P_1, P_2\}, \widetilde{\Phi}$.
  $\mathcal{Q} \in \mathrm{D}[\mathrm{ass}(\mathcal{T})][\varepsilon] \leftarrow$ the set of polynomials whose signs are determined
      during the call to Algorithm 1 (Morse partition) in the previous step.
```

[5] Not to be confused with the homological functor $\mathrm{Ext}(\cdot, \cdot)$.

```
7: $c = \frac{b}{a} \leftarrow \min_{Q \in \mathcal{Q}} c'(Q), a, b \in \mathrm{D}[\mathrm{ass}(\mathcal{T})]$ (cf. Notation 10).
8: $r \leftarrow \frac{a}{b}$.
9: Output $a, b$.
```

Complexity: The complexity of the algorithm is bounded by
$$D^{O(i)}(sd)^{O(k)},$$
where $s = \mathrm{card}(\mathcal{P})$, $d = \max_{P \in \mathcal{P}} \deg(P)$, and $D = \deg(\mathcal{T})$.

Proof of Correctness of Algorithm 2. Note that the formula $\widetilde{\Phi}$ defines a semi-algebraic subset
$$\widetilde{S} \subset \mathrm{R}\langle\varepsilon\rangle^i \times \mathrm{clos}(B_{k-i}(\mathbf{0}, \frac{1}{\varepsilon})).$$
It follows from the conic structure theorem at infinity of semi-algebraic sets (see for example [6, Proposition 5.49]) and the Tarski-Seidenberg transfer principle (see for example [6, Theorem 2.80]) that $\mathrm{ext}(S, \mathrm{R}\langle\varepsilon\rangle)_{\mathrm{ass}(\mathcal{T})}$ is semi-algebraically homeomorphic to $\widetilde{S}_{\mathrm{ass}(\mathcal{T})}$, and hence

$$\mathrm{H}_*\left(\mathrm{ext}(S, \mathrm{R}\langle\varepsilon\rangle)_{\mathrm{ass}(\mathcal{T})}\right) \cong \mathrm{H}_*\left(\widetilde{S}_{\mathrm{ass}(\mathcal{T})}\right). \tag{4.1}$$

It follows from the way r is computed in the algorithm in Line 7 and Lemma 3, that $c = \frac{1}{r}$ is strictly positive, and smaller than all strictly positive roots in R of the polynomials in $\mathcal{Q}$ (defined in Line 6 of the algorithm). It now follows from the ordering of the ring $\mathrm{D}[\mathrm{ass}(\mathcal{T})][\varepsilon]$ (cf. Notation 8) that all the branchings in the algorithm, each of which depend on the determination of the sign of an element in $\mathrm{D}[\mathrm{ass}(\mathcal{T})][\varepsilon]$, remain the same if c is substituted for ε.

It now follows from the correctness of Algorithm 1 (Morse partition), that the inclusion
$$\widetilde{S}_{\{\mathrm{ass}(\mathcal{T})\} \times (-\infty, r]} \hookrightarrow \widetilde{S}_{\{\mathrm{ass}(\mathcal{T})\} \times (-\infty, \frac{1}{\varepsilon}]} = \widetilde{S}_{\mathrm{ass}(\mathcal{T})}$$
(with r as computed in the algorithm) induces an isomorphism

$$\mathrm{H}_*\left(\widetilde{S}_{\{\mathrm{ass}(\mathcal{T})\} \times (-\infty, r]}\right) \to \mathrm{H}_*\left(\widetilde{S}_{\mathrm{ass}(\mathcal{T})}\right). \tag{4.2}$$

Moreover, for any $r' > 0$, $\widetilde{S}_{\{\mathrm{ass}(\mathcal{T})\} \times (-\infty, r']}$ is semi-algebraically homeomorphic to $\mathrm{ext}(S, \mathrm{R}\langle\varepsilon\rangle) \cap \{\mathrm{ass}(\mathcal{T})\} \times \mathrm{clos}(B_{k-i}(\mathbf{0}, r'))$, and hence

$$\mathrm{H}_*\left(\widetilde{S}_{\{\mathrm{ass}(\mathcal{T})\} \times (-\infty, r]}\right) \cong \mathrm{H}_*\left(\mathrm{ext}(S, \mathrm{R}\langle\varepsilon\rangle) \cap \{\mathrm{ass}(\mathcal{T})\} \times \mathrm{clos}(B_{k-i}(\mathbf{0}, r))\right). \tag{4.3}$$

Finally, for any closed semi-algebraic set $X \subset \mathrm{R}^k$,

$$\mathrm{H}_*(X) \cong \mathrm{H}_*(\mathrm{ext}(X, \mathrm{R}\langle\varepsilon\rangle)). \tag{4.4}$$

The isomorphisms (4.1), (4.2), (4.3), and (4.4) imply that
$$\mathrm{H}_*\left(S_{\mathrm{ass}(\mathcal{T})}\right) \cong \mathrm{H}_*(S \cap \{\mathrm{ass}(\mathcal{T})\} \times \mathrm{clos}(B_{k-i}(\mathbf{0}, r))).$$
This proves the correctness of Algorithm 2. □

Complexity of Algorithm 2. The stated complexity follows from the complexity of Algorithm 1 (Morse partition). □

4.2.2. *Constructing curve segments.* Another basic building block of our algorithm is an algorithm that takes as input a closed formula defining a semi-algebraic set, and produces as output a set of Thom encodings with associated elements $t_1 < \cdots < t_N$, and over each open interval (t_j, t_{j+1}), a set of semi-algebraic curves parametrized by the X_1 coordinate, with $t_j < X_1 < t_{j+1}$, described by curve segment representations, and for each j, a set of points (described by real univariate presentations) whose first coordinate equals t_j, such that the set of left and right end points of the curves are contained in the set of points. The main properties that these points and curve segments need to satisfy are listed in Proposition 2, stated after we describe the algorithm. Two remarks are in order.

Remark 9. First, note that this construction of curve segments parametrized by one of the coordinates (in our case X_1), satisfying properties very similar to those listed in Proposition 2, is a first step in the classical construction of roadmaps of semi-algebraic sets [17, 21, 22, 24, 4]. One important extra property that we require is that the partition of the X_1-axis satisfies the property of the output of Algorithm 1 (Morse partition), which is a little stronger than what is needed for such a partition in the classical roadmap constructions. We also ensure that distinct curves that we construct do not intersect (cf. Proposition 2, Part (d)). In the classical roadmap construction, this extra care is unnecessary.

Remark 10. For reasons explained previously, in the description of the algorithm we will assume that the first i coordinates are fixed (by a triangular Thom encoding), and the "first coordinate" in the description given in the beginning of this subsection should be replaced by the "$(i+1)$-st coordinate". This is in fact no different from the case of classical algorithms for computing roadmaps, which also relies on recursive calls in which a first block of i coordinates are fixed in each nested recursive with nesting depth i.

Finally, we note that since the main steps of the following algorithm are quite similar to the corresponding steps in several classical algorithms for constructing roadmaps, we omit a few details, giving pointers to algorithms in [6] to be used for implementing them in an efficient way.

Before formally describing the algorithm we give an outline to make it easy to follow the various steps.

Outline of Algorithm 3.

(a) Algorithm 3 will be used in Algorithm 4 repeatedly after fixing some of the first coordinates. Because of this the input to Algorithm 3 assumes that we have fixed the first i coordinates using a triangular Thom encoding $\mathcal{T}$.

(b) In Lines 1-3 we ensure boundedness of the semi-algebraic set necessary for the correctness of the algorithms called in the following steps.

(c) In Line 4 we use Algorithm 1 to obtain a partition of the X_{i+1}-axis satisfying the conditions of Proposition 1. The end points of the segments of the partitions are described using Thom encodings.

(d) In Line 5 we use Algorithm 16.12 from [6] to compute curve segments (cf. Step 1a) retaining only those that are parametrized by X_{i+1} (the second step is accomplished in Line 6).

(e) Since it is important from the point of computing the first homology groups, we need to ensure that that two different curve segments computed in the prior step can only intersect at their endpoints. We ensure this in Line 7 possibly increasing the cardinality of the set of curve segments in the process. This update is accomplished in Line 8.

(f) In Line 9 the partition of the X_{i+1}-cordinates axis is further refined by adding the X_{i+1}-coordinates of all the end points of the curve segments computed so far.

(g) In Lines 10-11 the set of curve segments is refined by ensuring each curve segment is defined over one of the intervals of the partition of the X_{i+1}-coordinate axis obtained in the previous step.

(h) In Lines 12-18 the various incidences between endpoints and the curves are computed and the output of the algorithm is assembled.

Algorithm 3 (Curve segments)

Input:

1. A triangular Thom encoding $\mathcal{T} = (\mathbf{F}, \boldsymbol{\sigma})$ of size i with $0 \leq i \leq k-1$;
2. a finite set $\mathcal{P} \subset \mathrm{D}[X_1, \ldots, X_k]$;
3. a $\mathcal{P}$-closed formula Φ such that $\mathcal{R}(\Phi)$ is bounded.

Output:

1. A finite tuple $\mathcal{F} = (\tau_1, \ldots, \tau_N)$ of Thom encodings over $\mathcal{T}$, with
$$t_1 = \mathrm{ass}(\tau_1) < \cdots < t_N = \mathrm{ass}(\tau_N);$$
2. for each $j, 1 \leq j \leq N-1$, an indexing set I_j, and a finite tuple $\mathcal{C}_j = (\gamma_h)_{h \in I_j}$ of curve segment representations over $\mathcal{T}$, such that for each $h \in I_j$
$$\tau_1(\gamma_h) = \tau_j, \tau_2(\gamma_h) = \tau_{j+1}$$
(we will let $C_0 = \mathcal{C}_{N+1} = \emptyset$);
3. for each $j, 1 \leq j \leq N$ a finite set $\mathcal{U}_j$ of real univariate representations over $\mathcal{T}$, such that for each $u \in U_j$, the set of points $\{\mathrm{ass}(u) \mid u \in \mathcal{U}_j\}$ includes the set of end-points of the curve segment representations in $C_{j-1} \cup C_j$;
4. mappings $L_j, R_{j-1} : I_j \to \mathcal{U}_j$, such that $\mathrm{ass}(L_j(h))$ is the left end-point of γ_h, and $\mathrm{ass}(R_j(\gamma_h))$ is the right end-point of γ_h.

Procedure:

```
Call Algorithm 2 (Big enough radius) with input $(\mathcal{T}, \mathcal{P}, \Phi)$ and compute $a, b \in \mathrm{D}[\mathrm{ass}(\mathcal{T})]$.
$\mathcal{P} \leftarrow \mathcal{P} \cup \{b^2(X_{i+1}^2 + \cdots + X_k^2) - a^2\}$.
$\Phi \leftarrow \Phi \wedge b^2(X_{i+1}^2 + \cdots + X_k^2) - a^2 \leq 0)$.
Call Algorithm 1 (Morse partition) with $(\mathcal{T}, r, \mathcal{P}, \Phi)$ as input and obtain a finite set $\mathcal{F}$ of Thom encodings over $\mathcal{T}$ as output.
Call Algorithm 16.12 (Bounded Roadmap) in [6] with input $\mathcal{P}, \Phi$ and the radius $r$, performing all computations over the ring $\mathrm{D}[\mathrm{ass}(\mathcal{T})]$.
Retain from the output of the previous step, the set $\mathcal{C}$ of curve segment representations over $\mathcal{T}$ parametrized by $X_{i+1}$, and the set $\mathcal{U}$ of real univariate representations over $\mathcal{T}$.
For each pair $\gamma, \gamma' \in \mathcal{C}$, compute a description of $\mathrm{ass}(\gamma) \cap \mathrm{ass}(\gamma')$ using Algorithm 14.6 (Parametrized Sign Determination) in [6], and refine the set $\mathcal{C}$ to have the property that $\mathrm{ass}(\gamma) \cap \mathrm{ass}(\gamma') = \emptyset$, for all $\gamma, \gamma' \in \mathcal{C}$, $\gamma \neq \gamma'$.
Augment the set $\mathcal{U}$ to also contain the set of real univariate representations over $\mathcal{T}$ whose associated points are the end points of the curve segments in $\mathcal{C}$.
Compute Thom encodings over $\mathcal{T}$ whose associated values are the $X_{i+1}$-coordinates of the asssociated points of $\mathcal{U}$, and add these to $\mathcal{F}$.
Using Algorithm 12.21 (Triangular Comparison of Roots) in [6] order the Thom encodings in $\mathcal{F}$, and let the associated values be $t_1 = \mathrm{ass}(\tau_1), \ldots, t_N = \mathrm{ass}(\tau_N)$.
    $$\mathcal{F} \leftarrow (\tau_1, \ldots, \tau_N).$$
Further refine $\mathcal{C}$, such that for each $\gamma \in \mathcal{C}$, there exists $j, 1 \leq j < N$, such that
    $$\tau_1(\gamma) = \tau_j, \tau_2(\gamma) = \tau_2.$$
Augment the set $\mathcal{U}$ to include the left and the right end points of each $\gamma \in \mathcal{C}$.
for each $j, 1 \leq j \leq N - 1$ do
    Let $I_j$ denote a set indexing the set of curve segment representations $\gamma \in \mathcal{C}$, such that $\tau_1(\gamma) = \tau_j, \tau_2(\gamma) = \tau_{j+1}$. For $h \in I_j$, we denote the corresponding curve segment representation in $\mathcal{C}$ by $\gamma_h$.
    $\mathcal{C}_j \leftarrow (\gamma_h)_{h \in I_j}$.
    $\mathcal{U}_j \leftarrow \{u \in \mathcal{U} \mid \pi_{i+1}(\mathrm{ass}(u)) = \mathrm{ass}(\tau_j)\}$.
    Compute the maps $L_j : I_j \to \mathcal{U}_j, R_j : I_j \to \mathcal{U}_{j+1}$, such that $\mathrm{ass}(L_j(h))$ is the left end-point of $\gamma_h$, and $\mathrm{ass}(R_j(h))$ is the right endpoint of $\gamma_h$.
end for
Output $\big(\mathcal{F}, (I_j, \mathcal{C}_j, \mathcal{U}_j, L_j, R_j)_{j \in [1,N]}\big)$.
```

Complexity: The complexity of the algorithm is bounded by $D^{O(i)}(sd)^{O((k-i)^2)}$, where $s = \mathrm{card}(\mathcal{P}), d = \max_{P \in \mathcal{P}} \deg(P)$, and $D = \deg(\mathcal{T})$.

The degrees of the curve segment representations in the various $\mathcal{C}_j$, and the degrees of the real univariate representations in $\mathcal{U}_j$ are both bounded by $(D, d^{O(k-i)})$. Finally the sum of the cardinalities

$$\sum_{j=1}^{N} (\mathrm{card}(\mathcal{C}_j) + \mathrm{card}(\mathcal{U}_j))$$

is bounded by $(sd)^{O(k-i)}$.

Proposition 2. *The output of Algorithm 3 (Curve segments) satisfies the following:*

(a) For each j, $1 \le j \le N-1$, and all $t \in [\mathrm{ass}(\tau_j), \mathrm{ass}(\tau_{j+1}))$ the inclusion maps

$$S_{\{\mathrm{ass}(\mathcal{T})\} \times (-\infty, \mathrm{ass}(\tau_j)]} \hookrightarrow S_{\{\mathrm{ass}(\mathcal{T})\} \times (-\infty, t]},$$

$$S_{\{\mathrm{ass}(\mathcal{T})\} \times \{t\}} \hookrightarrow S_{\{\mathrm{ass}(\mathcal{T})\} \times [\mathrm{ass}(\tau_j), \mathrm{ass}(\tau_{j+1})},$$

are homological equivalences;

(b) for each $h \in I_j$, $\mathrm{ass}(\gamma_h) \subset S = \mathcal{R}(\Phi)$;

(c) for each $t \in (\mathrm{ass}(\tau_j), \mathrm{ass}(\tau_{j+1}))$ and each semi-algebraically connected component C of $S_{\mathbf{y}}$, where $\mathbf{y} = (\mathrm{ass}(\mathcal{T}), t) \in \mathrm{R}^{i+1}$, there exists $h \in I_j$ such that $\mathrm{ass}(\gamma_h)_{\mathbf{y}} \in C$;

(d) if $h_1, h_2 \in I_j$ with $h_1 \neq h_2$, then $\mathrm{ass}(\gamma_{h_1}) \cap \mathrm{ass}(\gamma_{h_2}) = \emptyset$.

Proof. Part (a) follows from the property of the output of Algorithm 1 (Morse partition) called in Line 4 (cf. Proposition 1).

Parts (b) and (c) follow from the fact that the output of Algorithm 16.26 (General Roadmap) in [6] which is called in Line 5 describes a roadmap of the semi-algebraic set $\mathcal{R}(\Phi)$.

Finally, Part (d) is ensured in Line 7. □

Complexity analysis of Algorithm 3. The stated complexity follows from the complexity bounds of the various algorithms used in the algorithm, keeping in mind that each arithmetic operation in the ring $\mathrm{D}[\mathrm{ass}(\mathcal{T})]$ costs $D^{O(i)}$ arithmetic operations in the ring D. □

Remark 11. Note that in the step described in Line 5 it is sufficient to compute curve segment representations γ over $\mathcal{T}$, such that for each $\mathbf{y} = (\mathrm{ass}(\mathcal{T}, t) \in \mathrm{R}^{i+1}$, and each semi-algebraically connected component C of $S_{\mathbf{y}}$, there is a curve segment representation γ, such that $\mathrm{ass}(\gamma)_{\mathbf{y}} \cap C \neq \emptyset$. Calling Algorithm 16.12 (Bounded Roadmap) in [6] for this purpose is convenient but an overkill. The recursive calls in constructing a full roadmap leads to the quadratic dependence in the exponent - however, we do not need the parts of the roadmap in the various slices, but retain only the curve segments parametrized by X_{i+1}.

One could alternatively achieve the same goal by other algorithms having smaller complexity (namely, $D^{O(i)}(sd)^{O(k-i)}$ instead of $D^{O(i)}(sd)^{O((k-i)^2)}$). For example, it is possible to modify Algorithm 14.1 (Block Elimination) in [6] using as parameter the coordinate X_{i+1} for this purpose. Doing so would reduce the total complexity of the algorithm to $D^{O(i)}(sd)^{O(k-i)}$ instead of $D^{O(i)}(sd)^{O((k-i)^2)}$ as stated in the complexity bound. However, describing the modifications needed to Algorithm 14.1 (Block Elimination) in [6] would complicate the exposition and moreover will not change the asymptotic complexity of Algorithm 4 (Computing one-dimensional subset), which is the only place where Algorithm 3 is used. So we chose not to make this modification.

We are now in a position to describe our algorithm which will complete Step 1 of our proof of Theorem 2.

We first give an outline before describing the algorithm formally.

Outline of Algorithm 4.

(a) Because of the recursive structure of the algorithm where we make recursive calls after fixing some of the first coordinates, the input assumes that we have fixed the first i coordinates using a triangular Thom encoding $\mathcal{T}$. The "first coordinate" referred to in Step 1a refers to the $(i+1)$-st coordinate in Algorithm 4.

(b) The output consists of (over the same triangular Thom encoding $\mathcal{T}$) descriptions of curve segments and their end points and is self-explanatory.

(c) Lines 1 - 22 deal with the case of ambient dimension one. This is the base case of the recursion.

(d) In Line 23 the algorithm computes the curve segments. This corresponds to Step 1a in the outline given in Section 4.1.

(e) In Line 27 the algorithm makes the recursive calls. This corresponds to Step 1b in the outline given in Section 4.1.

(f) In Lines 29-33 the various curve segments and points are collected together for the output.

Algorithm 4 (Computing one-dimensional subset)

Input:

1. A triangular Thom encoding $\mathcal{T} = (\mathbf{F}, \boldsymbol{\sigma})$ of size i;
2. a finite set $\mathcal{P} \subset \mathrm{D}[X_1, \ldots, X_k]$;
3. a $\mathcal{P}$-closed formula Φ such that $\mathcal{R}(\Phi)$ is bounded;
4. a finite set $\mathcal{M}$ of real univariate representations over $\mathcal{T}$, whose set of associated points, M, is contained in $\mathcal{R}(\Phi)$.

Output:

1. a finite set $\mathcal{U}$ of real univariate representations over $\mathcal{T}$;
2. a finite indexing set I and a finite tuple $(\gamma_j)_{j \in I}$ where each γ_j a curve segment representation over $\mathcal{T}$;
3. mappings $L, R : I \to \mathcal{U}$, defined by $\mathrm{ass}(L(j))$ is the left endpoint of γ_j, and $\mathrm{ass}(R(j))$ is the right endpoint of γ_j.

Procedure:

```
1:  if $k - i = 1$ then
2:      for each $u = \{(f, g_0, g_k), \sigma\} \in \mathcal{M}$ do
3:          Let $R_u \in \mathrm{R}[X_k]$ be the Sylvester resultant (see for example [6, pp. 106]) with respect to the variable $T$ of the polynomials $f, X_k g_0 - g_k$.
4:          Use Algorithm 10.11 (Sign Determination) [6, pp. 390] to compute a Thom encoding $\tau_u = (R_u, \sigma_u)$ over $\mathcal{T}$, such that $\mathrm{ass}(\tau_u) = \pi_k(\mathrm{ass}(u))$.
5:          $\mathcal{P} \leftarrow \mathcal{P} \cup \{R_u\}$.
6:      end for
7:      Use Algorithm 12.21 (Triangular Comparison of Roots) in [6, pp. 496] repeatedly with inputs $\mathcal{T}$ and pairs of polynomials in $\mathcal{P}$, and order the real roots of the polynomials $P(\mathrm{ass}(\mathcal{T}), X_k), P \in \mathcal{P}$, and hence obtain a partition of $\mathrm{R}$ into points and open intervals, and identify those points and open intervals which are contained in $\mathcal{R}(\Phi, \mathrm{R}^k)_{\mathrm{ass}(\mathcal{T})}$.
8:      $\mathcal{U} \leftarrow \emptyset$.
9:      $I \leftarrow \emptyset$.
10:     $j \leftarrow 0$.
11:     for each Thom encoding $(P, \sigma)$ over $\mathcal{T}$ obtained in Line 4 whose associated point is in $S = \mathcal{R}(\Phi)$ do
12:         $\mathcal{U} \leftarrow \mathcal{U} \cup \{((P, X_k, 1), \sigma)\}$.
13:     end for
14:     for each open interval with end-points described by the Thom encodings $\tau_1 = ((P_1, X_k, 1), \sigma_1), \tau_2 = ((P_2, X_k, 1), \sigma_2) \in \mathcal{U}$ with $\mathrm{ass}(\tau_1) < \mathrm{ass}(\tau_2)$ such that $(\mathrm{ass}(\tau_1), \mathrm{ass}(\tau_2)) \subset \pi_k(\mathcal{R}(\Phi, \mathrm{R}^k))$ do
15:         $I \leftarrow I \cup \{j\}$.
16:         $j \leftarrow j + 1$.
17:         $\gamma_j \leftarrow \gamma$, where $\gamma$ is the curve segment representation over $\mathcal{T}$ defined by:
                $\tau_1(\gamma) = \tau_1,$
                $\tau_2(\gamma) = \tau_2,$
                $u(\gamma) = ((T, 1), (0, 1)).$
18:         $L(j) \leftarrow \tau_1(\gamma_j)$.
19:         $R(j) \leftarrow \tau_2(\gamma_j)$.
20:     end for
```

```
    Output $\mathcal{U}, (\gamma_j)_{j \in I}$, and the mappings $L, R : I \to \mathcal{U}$.
end if
Use Algorithm 3 (Curve segments) with input $(\mathcal{T}, \mathcal{P}, \Phi)$ to compute:
  (a) A finite tuple $\mathcal{F} = (\tau_1, \ldots, \tau_N)$ of Thom encodings over $\mathcal{T}$, with
      $\mathrm{ass}(\tau_1) < \cdots < \mathrm{ass}(\tau_N)$;
  (b) for each $j, 1 \leq j \leq N-1$, a finite tuple $\mathcal{C}_j$ of curve segment representations
      over $\mathcal{T}$ such that for each $\gamma \in \mathcal{C}_j$, $\tau_1(\gamma) = \tau_j, \tau_2(\gamma) = \tau_{j+1}$);
  (c) for each $j, 1 \leq j \leq N$ a finite set $\mathcal{U}_j$ of real univariate representations
      over $\mathcal{T}$, such that for each $u \in \mathcal{U}_j$, the set of points
      $\{\mathrm{ass}(u) \mid u \in \mathcal{U}_j\}$ is precisely the set of end-points of the curve
      segments in $\mathcal{C}_{j-1} \cup \mathcal{C}_j$ (with the convention that $\mathcal{C}_0 = \emptyset$);
  (d) mappings $L_j, R_{j-1} : \mathcal{C}_j \to \mathcal{U}_j$, such that $\mathrm{ass}(L_j(\gamma))$ is the left end-
      point of $\gamma$, and $\mathrm{ass}(R_j(\gamma))$ is the right end-point of $\gamma$.
for $\tau = (f, \sigma) \in \mathcal{F}$ do
    $\mathcal{M}_\tau \leftarrow \{u \in \mathcal{M} \cup \mathcal{U} \mid \pi_{i+1}(\mathrm{ass}(u)) = \mathrm{ass}(\tau)\}$.
    $\mathcal{T}_\tau \leftarrow ((F, f), (\boldsymbol{\sigma}, \sigma)))$.
    Call Algorithm 4 (Computing one-dimensional subset) recursively
        with input $(\mathcal{T}_\tau, \mathcal{P}, \Phi, \mathcal{M}_\tau)$ and obtain a set of $\mathcal{U}_\tau$ of real univariate
        representations over $\mathcal{T}_\tau$, an indexing set $I_\tau$, a tuple $(\gamma_i)_{i \in I_\tau}$ of
        curve segment representations, and mappings $L_\tau, R_\tau : I_\tau \to \mathcal{U}_\tau$.
        (Note that for each $i \in I_\tau, \mathrm{ass}(\gamma_i) \subset S_{\mathrm{ass}(\tau)}$.)
end for
$I \leftarrow \bigcup_{\tau \in \mathcal{F}} I_\tau$.
$\mathcal{U} \leftarrow \bigcup_{\tau \in \mathcal{F}} \mathcal{U}_\tau$.
$L \leftarrow \bigcup_{1 \leq j \leq N} L_j \cup \bigcup_{\tau \in \mathcal{F}} L_\tau$. (Union of disjoint mappings means the
    disjoint union of their graphs.)
$R \leftarrow \bigcup_{1 \leq j \leq N} R_j \cup \bigcup_{\tau \in \mathcal{F}} R_\tau$.
Output $\mathcal{U}, I, (\gamma_j)_{j \in I}, L, R$.
```

Complexity: Suppose that $\deg(\mathcal{T}) = d^{O(k)}$. The complexity of the algorithm is bounded by $(sd)^{O(k^2)}$, where $s = \mathrm{card}(\mathcal{P}), d = \max_{P \in \mathcal{P}}$. Moreover, $\mathrm{card}(I) = (sd)^{O((k-i)^2)}$, and the degrees of the elements of $\mathcal{U}$ and $\gamma_j, j \in I$ are bounded by $(d^{O(k)}, d^{O(k)})$.

Proposition 3. *The output of Algorithm 4 (Computing one-dimensional subset) has the following properties. Let* $\Gamma = \bigcup_{j \in I} \mathrm{clos}(\mathrm{ass}(\gamma_j))$*, and* $S = \mathcal{R}(\Phi, \mathrm{R}^k)_{\mathrm{ass}(\mathcal{T})}$.

(a) $M \subset \Gamma$*;*

(b) $\Gamma \subset S$*;*

(c) $\dim(\Gamma) \leq 1$*;*

(d) the homomorphism $i_{*,0} : \mathrm{H}_0(\Gamma) \to \mathrm{H}_0(S)$ *induced by the inclusion map* $i : \Gamma \hookrightarrow S$ *is an isomorphism;*

(e) the homomorphism $i_{*,1} : \mathrm{H}_1(\Gamma) \to \mathrm{H}_1(S)$ *induced by the inclusion map* $i : \Gamma \hookrightarrow S$ *is an epimorphism.*

Proof. The property in Part (a) is ensured in Lines 7 and 25. Part (b) follows from the property of the output of Algorithm 3 (Curve segments) (called in Line 23) given in Part (b) of Proposition 2.

Part (c) is clear since Γ is by definition the finite union

$$\bigcup_{j \in I} \mathrm{clos}(\mathrm{ass}(\gamma_j))$$

and $\dim \mathrm{ass}(\gamma_j) = 1$ for each $j \in I$, and taking the closure does not increase the dimension of a semi-algebraic set.

It is a standard exercise (see for example proof of Proposition 15.7 in [6]) to prove that the semi-algebraic set Γ satisfies the properties of being roadmap for $S = \mathcal{R}(\Phi)$ (see [6, Chapter 15]), which implies Part (d).

We now prove Part (e).

The proof is by induction on $k - i$.

Base case: $k - i = 1$. In this case the claim is clear since $\mathrm{H}_1(\Gamma) = \mathrm{H}_1(S) = 0$.

Suppose the claim is true for all smaller values of $k - i$. Notice that Algorithm 4 (Computing one-dimensional subset) is called recursively in Line 27. In these calls the triangular Thom encoding $\mathcal{T}_\tau, \tau \in \mathcal{F}$ in the input is of size $i + 1$, while the number of variables is still k. We have also have,

$$\Gamma_\tau := \mathrm{clos}\left(\bigcup_{j \in I_\tau} \mathrm{ass}(\gamma_j)\right) = \Gamma_{\mathrm{ass}(\tau)}.$$

Thus, using the induction hypothesis for this recursive call (since $k-(i+1) < k - i$), we obtain that the restriction of the inclusion $\Gamma \to S$ to Γ_τ induces a surjection

$$\mathrm{H}_1(\Gamma_\tau) \to \mathrm{H}_1(S_{\mathrm{ass}(\tau)}). \tag{4.5}$$

Denote for $1 \leq j \leq N$, $t_j = \pi_{i+1}(\mathrm{ass}(\tau_j))$.

$$\begin{aligned} a_0 &= t_1, \\ a_j &= \frac{t_j + t_{j+1}}{2}, 1 \leq j < N, \\ a_N &= t_N. \end{aligned}$$

We prove the following claims.

Claim 1. *For each $j, 0 \leq j \leq N$, the inclusion maps induce the following isomorphisms.*

$$\mathrm{H}_*(S_{(-\infty,t_j]}) \to \mathrm{H}_*(S_{(-\infty,a_j]}), \tag{4.6}$$

$$\mathrm{H}_*(S_{t_j}) \to \mathrm{H}_*(S_{[a_{j-1},a_j]}), \tag{4.7}$$

$$\mathrm{H}_*(\Gamma_{(-\infty,t_j]}) \to \mathrm{H}_*(\Gamma_{(-\infty,a_j]}), \tag{4.8}$$

$$\mathrm{H}_*(\Gamma_{t_j}) \to \mathrm{H}_*(\Gamma_{[a_{j-1},a_j]}). \tag{4.9}$$

Proof. Parts (4.6) and (4.7) are consequences of the property of the output of Algorithm 3 (Curve segments) (which is called in Line 23) given in Part (a) of Proposition 2.

Parts (4.8) and (4.9) follow from the fact that there is an easy to define retraction (along the X_{i+1}-coordinate) of $\Gamma_{(-\infty,a_j]}$ to $\Gamma_{(-\infty,t_j]}$ (resp. $\Gamma_{[a_{j-1},a_j]}$ to Γ_{t_j}) making use of the fact that distinct curve segments over the open intervals (t_{j-1},t_j) do not intersect which is ensured by Part (d) of Proposition 2. □

Claim 2. *Let $a,b \in \{a_0,\ldots,a_N\}$, with $a \leq b$. The inclusion map $\Gamma \hookrightarrow S$ induces isomorphisms*

$$\mathrm{H}_0(\Gamma_{(-\infty,a]}) \to \mathrm{H}_0(S_{(\infty,a]}),$$
$$\mathrm{H}_0(\Gamma_{[a,b]}) \to \mathrm{H}_0(S_{[a,b]}),$$

Proof. This follows from the fact that $\Gamma_{(-\infty,a]}$ (resp. $\Gamma_{[a,b]}$) satisfy the roadmap property with respect to the set $S_{(\infty,a]}$ (resp. $S_{[a,b]}$). The proof of this fact is standard and omitted. □

Using the claims proved above, we are now going to prove using induction on j, that the inclusion map $\Gamma \to S$ induces an isomorphism,

$$\mathrm{H}_1(\Gamma_{(\infty,a_j]}) \to \mathrm{H}_1(S_{(\infty,a_j]}).$$

The claim is true for $j=0$ by the global induction hypothesis on i, and hence is also true for $j=1$ using using (4.6) and (4.8).

We prove it for $j>1$ by induction. Suppose the claim holds until $j-1$. Hence we have isomorphism

$$\mathrm{H}_1(\Gamma_{(\infty,a_{j-1}]}) \to \mathrm{H}_1(S_{(\infty,a_{j-1}]})$$

induced by inclusion.

Observe that for any set $X \subset \mathrm{R}^k$,

$$\begin{aligned} X_{(-\infty,a_j]} &= X_{(-\infty,a_{j-1}]} \cup X_{[a_{j-1},a_j]}, \\ X_{a_{j-1}} &= X_{(-\infty,a_{j-1}]} \cap X_{[a_{j-1},a_j]} X_{a_{j-1}}. \end{aligned}$$

Let

$$\begin{aligned} A_1 &= \Gamma_{(-\infty,a_{j-1}]}, \\ A_2 &= \Gamma_{[a_{j-1},a_j]}, \\ B_1 &= S_{(-\infty,a_{j-1}]}, \\ B_2 &= S_{[a_{j-1},a_j]}. \end{aligned}$$

Also, let A_{12} (resp. B_{12}) denote $A_1 \cap A_2$ (resp. $B_1 \cap B_2$), and A^{12} (resp. B^{12}) denote $A_1 \cup A_2$ (resp. $B_1 \cup B_2$).

The Mayer-Vietoris exact sequence (see for example [6, Theorem 6.35]) yields the following commutative diagrams with exact rows and vertical arrows induced by various restrictions of the inclusion $\Gamma \hookrightarrow S$.

$$\begin{tikzcd}
\mathrm{H}_1(A_{12}) \arrow[r] \arrow[d, "a"] & \mathrm{H}_1(A_1) \oplus \mathrm{H}_1(A_2) \arrow[r] \arrow[d, "b_1 \oplus b_2"] & \mathrm{H}_1(A^{12}) \arrow[r] \arrow[d, "c"] & \mathrm{H}_0(A_{12}) \arrow[r] \arrow[d, "d"] & \mathrm{H}_0(A_1) \oplus \mathrm{H}_0(A_2) \arrow[d, "e_1 \oplus e_2"] \\
\mathrm{H}_1(B_{12}) \arrow[r] & \mathrm{H}_1(B_1) \oplus \mathrm{H}_1(B_2) \arrow[r] & \mathrm{H}_1(B^{12}) \arrow[r] & \mathrm{H}_0(B_{12}) \arrow[r] & \mathrm{H}_0(B_1) \oplus \mathrm{H}_0(B_2)
\end{tikzcd}$$

By induction hypothesis on j, the map $b_1 : \mathrm{H}_1(A_1) \to \mathrm{H}_1(B_1)$ is surjective. Using (4.5), (4.7), and (4.9) we have that the map $b_2 : \mathrm{H}_1(A_2) \to \mathrm{H}_1(B_2)$ is surjective. Hence the map $b = b_1 \oplus b_2$ is surjective.

Using Proposition 2 (c) we have that map d is surjective.

Finally, using Claim 2, we have that the maps e_1 and e_2 are both isomorphisms, and hence so is e. In particular, e is injective.

It follows from the above and (one-half of) the Five Lemma (see for example [18, Proposition 1.1, Part (2)], that c is a surjection. □

Complexity analysis of Algorithm 4. It follows from the complexity bounds on the algorithms used that all the steps before the recursive call in Line 27 has complexity bounded by $(sd)^{O((k-i)^2)}$ in terms of the number of arithmetic operations in $\mathrm{D}[\mathrm{ass}(\mathcal{T})]$. Hence, the number of arithmetic operations in D for these steps is bounded by $d^{O(ki)}(sd)^{O((k-i)^2)}$.

There are $(sd)^{O(k-i)}$ recursive calls. For each of the recursive calls, the new triangular Thom encoding is of size $i+1$, and its degree is bounded again by $d^{O(k)}$. An easy inductive argument now implies the stated complexity bounds. □

4.3. **Implementing Step 2: proofs of Theorems 2 and 3.** We will now prove Theorem 2 by describing an algorithm (cf. Algorithm 7 below) for computing a semi-algebraic basis of $\mathrm{H}_1(\mathcal{R}(\Phi))$, for any given closed formula Φ. Theorem 2 will then follow from the proof of correctness of this algorithm and its complexity analysis.

4.3.1. *Outline of Algorithm 7.* Our main tool will be Algorithm 4 (Computing one-dimensional subset) that produces a one-dimensional subset Γ of $S = \mathcal{R}(\Phi)$, such that the image of $\mathrm{H}_1(\Gamma)$ in $\mathrm{H}_1(S)$ under the linear map induced by inclusion is surjective. The semi-algebraic set Γ has an underlying structure of a finite graph G, and Γ is semi-algebraically homeomorphic to the geometric realization $|G|$ of G. Using a combinatorial graph-theoretic algorithm it is easy to compute a basis of the cycle space, $Z(G) \cong \mathrm{H}_1(\Gamma)$, consisting of simple cycles (say) $C_1, \ldots, C_N$. Each such simple cycle C_i is a subgraph of G, and its geometric realization $|C_i| \subset |G| \cong \Gamma$ is semi-algebraically homeomorphic to $\mathbf{S}^1$.

We will denote by $[C_i]$ a non-zero element of the image of $\mathrm{H}_1(|C_i|)$ in $\mathrm{H}_1(S)$ if this image is non-zero (and let $[C_i] = 0$ otherwise). However, the set $\{[C_1], \ldots, [C_N]\}$ need not be linearly independent but thanks to the surjectivity of the map $\mathrm{H}_1(\Gamma) \to \mathrm{H}_1(S)$, they span $\mathrm{H}_1(S)$. We identify a minimal subset of $\{[C_1], \ldots, [C_N]\}$ which spans $\mathrm{H}_1(S)$ (i.e. form a basis of

$\mathrm{H}_1(S)$). The corresponding subsets of Γ then constitutes a semi-algebraic basis of $\mathrm{H}_1(S)$. In order to achieve this last step effectively and with singly exponential complexity, we use a recent result proved in [9], giving an algorithm (see Algorithm 6 below for input, output and complexity) with singly exponential complexity for replacing a tuple of given closed and bounded semi-algebraic subsets of R^k, by a simplicial complex Δ which is homologically 1-equivalent to the given tuple of sets (cf. Definition 11). This simplicial complex is of singly exponential bounded size, and the various $|C_i|$'s can be identified with subcomplexes of this complex. So to find a basis from amongst the elements $[C_1], \ldots, [C_N]$ becomes a problem of ordinary linear algebra which can be solved with complexity bounded by a polynomial in the size of the simplicial complex Δ.

4.3.2. *Conversion of curve segment representations to closed formulas.* We will need the following algorithm. It is needed in order to address the following technical issue.

The output of Algorithm 4 (Computing one-dimensional subset) contains amongst other objects, a set of curve segment representations. We will need to convert these descriptions into *closed formulas* describing the closure of the associated curves in order to use Algorithm 6 which only accepts such descriptions in the input.

Note that the algorithmic problem of computing a *closed* formula describing the closure of a given semi-algebraic set described by a quantifier-free (but not necessarily closed-) formula is far from being easy, and no algorithm with singly exponential complexity is known for solving this problem in general [6]. (A doubly exponential algorithm is known, using the notion of a stratifying family [6, Chapter 5]). However, fortunately for us the curve segment representations describing the associated curve have a special structure. In particular, it is clear that given a curve segment representation γ, it is algorithmically quite simple to obtain a description of the the image, $\pi_{\{1,j\}}(\mathrm{ass}(\gamma))$, of the projection of $\mathrm{ass}(\gamma)$, to each of the coordinate subspaces spanned by $(X_1, X_j), 2 \leq j \leq k$. More precisely, suppose that γ is the curve segment representation with

$$u(\gamma) = ((f, g_0, \ldots, g_k), \sigma),$$

where $f, g_i \in \mathrm{D}[X_1, T]$, and $\sigma \in \{0, 1, -1\}^{\mathrm{Der}_T(f)}$. Then, $\mathrm{ass}(\gamma)$ is defined by

$$\mathrm{ass}(\gamma) = \left\{ \left(x_1, \frac{g_2(x_1, t(x_1))}{g_0(x_1, t(x_1))}, \ldots, \frac{g_k(x_1, t(x_1))}{g_0(x_1, t(x_1))} \right) \;\middle|\; \mathrm{ass}(\tau_1(\gamma)) < x_1 < \mathrm{ass}(\tau_2(\gamma)) \right\},$$

where for each $x_1, \mathrm{ass}(\tau_1(\gamma)) < x_1 < \mathrm{ass}(\tau_2(\gamma))$, and $t(x_1)$ is a root of $f(x_1, T)$ with Thom encoding σ.

[6] Note that without the requirement that the output formula be closed it is straightforward to obtain a singly exponential complexity algorithm via quantifier elimination.

Now for each $j, 2 \leq j \leq k$, the projection of $\mathrm{ass}(\gamma)$ to the (X_1, X_j)-plane is described by

$$\pi_{\{1,j\}}(\mathrm{ass}(\gamma)) = \left\{ \left(x_1, \frac{g_j(x_1, t(x_1))}{g_0(x_1, t(x_1))} \right) \;\middle|\; \mathrm{ass}(\tau_1(\gamma)) < x_1 < \mathrm{ass}(\tau_2(\gamma)) \right\}.$$

Using an effective quantifier elimination (eliminating T), one can obtain from the above description a quantifier-free formula with free variables X_1, X_j whose realization is equal to $\pi_{\{1,j\}}(\mathrm{ass}(\gamma))$.

We will use the following claim.

Claim 3. *Suppose that γ is a curve segment representation and* $\mathrm{ass}(\gamma)$ *is bounded, then*

$$\mathrm{clos}(\mathrm{ass}(\gamma)) = \bigcap_{2 \leq j \leq k} \pi^{-1}_{\{1,j\}}(\mathrm{clos}(\pi_{\{1,j\}}(\mathrm{ass}(\gamma)))).$$

Proof of Claim 3. First, suppose that $\mathbf{x} \in \mathrm{clos}(\mathrm{ass}(\gamma))$. Then we have that

$$\mathbf{x} \in \pi^{-1}_{\{1,j\}}(\pi_{\{1,j\}}(\mathrm{ass}(\gamma)))$$

for all j. It is a general fact from topology that for any continuous function f and set A, $f(\mathrm{clos}(A)) \subset \mathrm{clos}(f(A))$. Hence,

$$\pi^{-1}_{\{1,j\}}(\pi_{\{1,j\}}(\mathrm{ass}(\gamma))) \subset \pi^{-1}_{\{1,j\}}(\mathrm{clos}(\pi_{\{1,j\}}(\mathrm{ass}(\gamma))))$$

for all j. Therefore $\mathbf{x} \in \bigcap_{2 \leq j \leq k} \pi^{-1}_{\{1,j\}}(\mathrm{clos}(\pi_{\{1,j\}}(\mathrm{ass}(\gamma))))$.

Now, suppose that

$$\mathbf{x} = (x_1, \ldots, x_k) \in \bigcap_{2 \leq j \leq k} \pi^{-1}_{\{1,j\}}(\mathrm{clos}(\pi_{\{1,j\}}(\mathrm{ass}(\gamma))))$$

and let $\mathbf{x}_{1,j} = \pi_{\{1,j\}}(\mathbf{x})$ for $j = 2, \ldots, k$.

Since, $\mathbf{x}_{1,j} \in \mathrm{clos}(\pi_{\{1,j\}}(\mathrm{ass}(\gamma)))$, using the semi-algebraic curve selection lemma (see for example [6, Theorem 3.19]) there exists $t_{j,0} > 0$, such that there exists a semi-algebraic curve, $\gamma_j = (\gamma_{1,j}, \gamma_{2,j}) : [0, t_{j,0}] : \mathrm{R}^2$, such that $\gamma_{1,j}, \gamma_{2,s}$ are continuous semi-algebraic functions, $\gamma_j(0) = \mathbf{x}_{1,j}$, and $\gamma((0, t_{j,0}]) \subset \pi_{\{1,j\}}(\mathrm{ass}(\gamma))$. Moreover, clearly since $\pi_{\{1,j\}}(\mathrm{ass}(\gamma))$ is a curve parametrized by the X_1 coordinate, $\gamma_{1,j}$ is not a constant function, and without loss of generality (choosing $t_{j,0}$ smaller if necessary) we can assume that $\gamma_{1,j}$ is a strictly increasing function. For $j = 2, \ldots, k$, let $f_j : [x_1, \gamma_{1,j}(t_{j,0})] \to \mathrm{R}$ be defined by

$$f_j(X_1) = \gamma_{2,j}(\gamma^{-1}_{1,j}(X_1)).$$

Taking $x'_1 = \min_{2 \leq j \leq k} \gamma_{1,j}(t_{j,0})$, we obtain a semi-algebraic curve $\widetilde{\gamma} : [x_1, x'_1] \to \mathrm{ass}(\gamma)$, defined by

$$\widetilde{\gamma}(X_1) = (X_1, f_2(X_1), \ldots, f_k(X_1)).$$

It is easy to check that

$$\widetilde{\gamma}(x_1) = \mathbf{x},$$

and

$$\widetilde{\gamma} : (x_1, x'_1] \subset \mathrm{ass}(\gamma),$$

which proves that $\mathbf{x} \in \mathrm{clos}(\mathrm{ass}(\gamma))$. □

Using Claim 3 we reduce the problem of computing a closed description of $\mathrm{clos}(\mathrm{ass}(\gamma))$ to the problem of computing the closures of $\pi_{\{1,j\}}(\mathrm{ass}(\gamma))$, $2 \leq j \leq k$, and each of the latter is a 2-dimensional problems which can be solved within our allowed complexity bound using the doubly exponential algorithm referred to previously.

Finally, as in the case of the other algorithms in this paper we include in the input a triangular Thom encoding $\mathcal{T}$ that fixes the first i-coordinates, and the curve segment representations in the input is over $\mathcal{T}$. The computations in the algorithm takes place in the ring $\mathrm{D}[\mathrm{ass}(\mathcal{T})]$, and in the description given above, the first coordinate is replaced by the $(i+1)$-st coordinate.

Algorithm 5 (Conversion of curve segment representations to closed formulas)

Input:

1. A triangular Thom encoding $\mathcal{T} = (\mathbf{F}, \boldsymbol{\sigma})$ of size $i, 0 \leq i \leq k$;
2. a curve segment representation γ over $\mathcal{T}$.

Output:

1. A finite set of polynomials $\mathcal{Q} \subset \mathrm{D}[\mathrm{ass}(\mathcal{T}][X_{i+1}, \ldots, X_k]$;
2. A $\mathcal{Q}$-closed formula Ψ such that

$$\mathcal{R}(\Psi, \mathrm{R}^k)_{\mathrm{ass}(\mathcal{T})} = \mathrm{clos}(\mathrm{ass}(\gamma)).$$

Procedure:

```
$u \leftarrow u(\gamma) = ((f, g_0, g_{i+2}, \ldots, g_k), \sigma)$.
for $j = i+2, \ldots, k$ do
    Using Algorithm 14.5 (Quantifier Elimination) in [6, pp. 549] with
        the formula
        $(\exists T)(f = 0) \wedge \bigwedge_{1 \leq h \leq \deg_T(f)} (\mathbf{sign}(f^{(h)}) = \sigma(f^{(h)})) \wedge (X_j g_0 - g_j = 0)$,
        as input, and obtain a $\widetilde{\mathcal{Q}}_j$-formula quantifier-free formula $\widetilde{\phi}_j$, for
        some $\widetilde{\mathcal{Q}}_j \subset \mathrm{D}[\mathrm{ass}(\mathcal{T})][X_{i+1}, X_j]$ describing such that $\mathcal{R}(\widetilde{\phi}_j) =$
        $\pi_{[1,i+1] \cup \{j\}}(\mathrm{ass}(\gamma))$.
```

```
4:      Compute a stratifying family of polynomials (see [6, Proposition 5.40]
            for definition), $\mathcal{Q}_j \subset \mathrm{D}[\mathrm{ass}(\mathcal{T})][X_{i+1}, X_j]$ containing $\widetilde{\mathcal{Q}}_j$.
5:      Using Algorithm 13.1 (Computing realizable sign conditions) in [6,
            pp. 511] determine the set $\Sigma_j \subset \{0, 1, -1\}^{\mathcal{Q}_j}$ of realizable sign
            conditions of $\mathcal{Q}_j$.
6:      Determine $\Theta_j \subset \Sigma_j$ such that $\pi_{[1,i+1]\cup\{j\}}(\mathrm{ass}(\gamma)) = \bigcup_{\theta \in \Theta_j} \mathcal{R}(\theta)$.
7:      $\Psi_j \leftarrow \bigvee_{\theta \in \Theta_j} \overline{\theta}$.
8:  end for
9:  $\mathcal{Q} \leftarrow \mathbf{F} \cup \bigcup_{i+2 \le j \le k} \mathcal{Q}_j$.
10: $\Psi \leftarrow \overline{\boldsymbol{\sigma}} \wedge \bigwedge_{i+2 \le j \le k} \Psi_j$ (see Notation 7 for definition of $\overline{\boldsymbol{\sigma}}$).
11: Output $\mathcal{Q}, \Psi$.
```

Complexity: The complexity of the algorithm is bounded by $(k - i)D_1^{O(i)} D_2^{O(1)}$, where $D_1 = \deg(\mathcal{T})$, and $D_2 = \deg(\gamma)$. Moreover, $\mathrm{card}(\mathcal{Q})$ is bounded by $i + (k - i)D_2^{O(1)}$, and the degrees of the polynomials in $\mathcal{Q}$ are bounded by $\max(D_1, D_2^{O(1)})$.

Proof of Correctness of Algorithm 5. The algorithm reduces the problem to obtaining closed formulas describing the closures of the various $\pi_{i+1,j}(\mathrm{ass}(\gamma))$, $i + 2 \le j \le k$ (cf. Line 2). After obtaining the descriptions of the various $\pi_{i+1,j}(\mathrm{ass}(\gamma))$ using effective quantifier elimination algorithm (Algorithm 14.5 (Quantifier Elimination) in [6, pp. 549] called in Line 3), closed formulas are obtained describing the closure by computing a stratifying family (cf. 4) in each case. The important property of the stratifying families $\mathcal{Q}_j$ is that the closures, $\mathrm{clos}(\pi_{i+1,j}(\mathrm{ass}(\gamma)))$ are unions of realizations of a set of weak sign conditions on $\mathcal{Q}_j$. This is a consequence of the generalized Thom's Lemma (see [6, Proposition 5.39]). Finally, the set of weak sign conditions on $\mathcal{Q}_j$ whose realizations are contained in $\mathrm{clos}(\pi_{i+1,j}(\mathrm{ass}(\gamma)))$ computed using Algorithm 13.1 (Computing realizable sign conditions) in [6, pp. 511] in Line 5. The disjunction of these weak formulas gives a closed formula, Ψ_j describing $\mathrm{clos}(\pi_{i+1,j}(\mathrm{ass}(\gamma)))$ (cf. Line 7). Now Claim 3 together with Lemma 2 (and noting that the conjunction of a finite set of closed formulas is also closed), imply that the conjunction, Ψ, of the formulas $\Psi_j, j = i + 2, \ldots, k$ along with the closed formula $\overline{\boldsymbol{\sigma}}$ (Notation 7), describes $\mathrm{clos}(\mathrm{ass}(\gamma))$ (cf. Line 10). This proves the correctness of the algorithm. □

Complexity analysis of Algorithm 5. As before, each arithmetic operation in $\mathrm{D}[\mathrm{ass}(\mathcal{T})]$ costs $D_1^{O(i)}$ arithmetic operations in D (where $D_1 = \deg(\mathcal{T})$). There are $(k - i)$ two dimensional projections (cf. Line 2). The complexity of each of these two-dimensional sub-problems (measured in terms of number of operations in $\mathrm{D}[\mathrm{ass}(\mathcal{T})]$) is bounded by $D_2^{O(1)}$, where $D_2 = \deg(\gamma)$, and this follows from the complexity bounds on the various algorithms used in the different steps (namely, Algorithm 14.5 (Quantifier Elimination) in [6, pp. 549] in Line 3, algorithm for computing stratifying families in 4, and

Algorithm 13.1 (Computing realizable sign conditions) in [6, pp. 511] in Line 5). Note that these algorithms are used with the number of variables equal to 2, and hence the complexity of each call (measured in terms of arithmetic operations in $\mathrm{D}[\mathrm{ass}(\mathcal{T})]$) are polynomially bounded in D_1. This completes the complexity analysis of Algorithm 3. □

4.3.3. *Efficient algorithm for computing a simplicial replacement.* As explained in the outline above, in order to obtain a semi-algebraic basis we will use an algorithm described in [9]. We reproduce below for the reader's benefit the input, output and complexity of this algorithm. But before stating these we need some preliminary definitions.

Definition 8 (Homological ℓ-equivalence)**.** We say that a map $f : X \to Y$ between two topological spaces is a *homological ℓ-equivalence* if the induced homomorphisms between the homology groups $f_* : \mathrm{H}_i(X) \to \mathrm{H}_i(Y)$ are isomorphisms for $0 \leq i \leq \ell$.

The relation of homological ℓ-equivalence as defined above is not an equivalence relation since it is not symmetric. In order to make it symmetric one needs to "formally invert" ℓ-equivalences.

Definition 9 (Homologically ℓ-equivalent)**.** We will say that X *is homologically ℓ-equivalent to* Y (denoted $X \sim_\ell Y$), if and only if there exists spaces, $X = X_0, X_1, \ldots, X_n = Y$ and homological ℓ-equivalences $f_1, \ldots, f_n$ as shown below:

$$\begin{tikzcd}
& X_1 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & & X_3 \arrow[dl, "f_3"'] \arrow[dr, "f_4"] & \cdots & \cdots & X_{n-1} \arrow[dl, "f_{n-1}"'] \arrow[dr, "f_n"] & & . \\
X_0 & & X_2 & & \cdots & \cdots & & X_n &
\end{tikzcd}$$

It is clear that $\sim_\ell$ is an equivalence relation.

Definition 10 (Diagrams of topological spaces)**.** A *diagram of topological spaces* is a functor, $X : J \to \mathbf{Top}$, from a small category J to $\mathbf{Top}$.

We extend Definition 8 to diagrams of topological spaces. We denote by **Top** the category of topological spaces.

Definition 11 (Homological ℓ-equivalence between diagrams of topological spaces)**.** Let J be a small category, and $X, Y : J \to \mathbf{Top}$ be two functors.

We will say that *a diagram* $X : J \to \mathbf{Top}$ *is homologically ℓ-equivalent to the diagram* $Y : J \to \mathbf{Top}$ (denoted as before by $X \sim_\ell Y$), if and only if there exists diagrams $X = X_0, X_1, \ldots, X_n = Y : J \to \mathbf{Top}$ and homological ℓ-equivalences $f_1, \ldots, f_n$ as shown below:

$$\begin{tikzcd}
& X_1 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & & X_3 \arrow[dl, "f_3"'] \arrow[dr, "f_4"] & \cdots & \cdots & X_{n-1} \arrow[dl, "f_{n-1}"'] \arrow[dr, "f_n"] & & . \\
X_0 & & X_2 & & \cdots & \cdots & & X_n &
\end{tikzcd}$$

It is clear that $\sim_\ell$ is an equivalence relation.

Notation 11 (Diagram of various unions of a finite number of subspaces)**.** Let J be a finite set, A a topological space, and $\mathcal{A} = (A_j)_{j\in J}$ a tuple of subspaces of A indexed by J.

For any subset $J' \subset J$, we denote

$$\mathcal{A}^{J'} \quad = \quad \bigcup_{j'\in J'} A_{j'}.$$

We consider 2^J as a category whose objects are elements of 2^J, and whose only morphisms are given by:

$$\begin{aligned} 2^J(J', J'') &= \emptyset \text{ if } J' \not\subset J'', \\ 2^J(J', J'') &= \{\iota_{J',J''}\} \text{ if } J' \subset J''. \end{aligned}$$

We denote by $\mathbf{Simp}^J(\mathcal{A}) : 2^J \to \mathbf{Top}$ the functor (or the diagram) defined by

$$\mathbf{Simp}^J(\mathcal{A})(J') = \mathcal{A}^{J'}, J' \in 2^J,$$

and $\mathbf{Simp}^J(\mathcal{A})(\iota_{J',J''})$ is the inclusion map $\mathcal{A}^{J'} \hookrightarrow \mathcal{A}^{J''}$.

Notation 12. For $N \in \mathbb{Z}$ we denote by $[n] = \{0, \ldots, N\}$. In particular, $[-1] = \emptyset$.

Armed with the definition of homological equivalence of diagrams as defined above we are finally in a position to state the specifications of the simplicial replacement algorithm described in [9].

Algorithm 6 (Simplicial replacement)

Input:

1. A finite set of polynomials $\mathcal{P} \subset \mathrm{D}[X_1, \ldots, X_k]$;
2. an integer $N \geq 0$, and for each $i \in [N]$, a $\mathcal{P}$-closed formula ϕ_i;
3. $\ell, 0 \leq \ell \leq k$.

Output:

A simplicial complex Δ and for each $I \subset [N]$ a subcomplex $\Delta_I \subset \Delta$ such that there is a diagrammatic homological ℓ-equivalence

$$(I \mapsto \Delta_I)_{I\subset[N]} \overset{h}{\sim}_\ell \mathbf{Simp}^{[N]}(\mathcal{R}(\Phi)),$$

where $\Phi(i) = \phi_i, i \in [N]$.

Complexity: The complexity of the algorithm is bounded by $(sd)^{k^{O(\ell)}}$, where $s = \mathrm{card}(\mathcal{P})$ and $d = \max_{P\in\mathcal{P}} \deg(P)$.

Notation 13. A finite (directed) graph G is a tuple $(V(G), E(G), \mathrm{head}, \mathrm{tail})$, where $V(G), E(G)$ are finite sets and $\mathrm{head}, \mathrm{tail} : E(G) \to V(G)$ are maps. To every finite graph G there is an one-dimensional regular cell complex associated naturally to it. We will denote this cell complex by $|G|$.

4.3.4. *Algorithm for computing basis of* $\mathrm{H}_1(S)$. We are now in a position to describe the algorithm that will accomplish Step 2 of our algorithm for computing a semi-algebraic basis of the first homology of a given closed semi-algebraic set.

We first give an outline of the algorithm followed by a formal description.

Outline of Algorithm 7.

(a) We first replace the given closed semi-algebraic set by one that is closed and bounded and which is homologically equivalent to the given set. This is accomplished in Lines 1-3 using Algorithm 2 (Big enough radius).
(b) We then use Algorithm 4 (Computing one-dimensional subset) to compute a one-dimensional subset Γ (described in terms of curve-segment representations and real univariate representations) having the property that $\mathrm{H}_q(S, \Gamma) = 0$ for $q = 0, 1$, where S is the given semi-algebraic set. This is accomplished in Line 4.
(c) We then use Algorithm 5 (Conversion of curve segment representations to closed formulas) in Line 5 to convert the description of Γ obtained in the last step into a formula.
(d) In Line 8, we extract the underlying structure of a combinatorial graph G from Γ and compute a set of simple cycles which spans the cycle space of G.
(e) Finally in Lines 12-13, we use Algorithm 6 (Simplicial replacement) and Gauss-Jordan elimination to compute a minimal subset of the simple cycles computed in the previous step which span $\mathrm{H}_1(S)$, and these form a semi-algebraic basis of $\mathrm{H}_1(S)$.

Algorithm 7 (Computing homology basis)

Input:
1. a finite set $\mathcal{P} \subset \mathrm{D}[X_1, \ldots, X_k]$;
2. a $\mathcal{P}$-closed formula Φ.

Output:
1. a finite set $\mathcal{Q} \subset \mathrm{D}[X_1, \ldots, X_k]$;
2. a finite tuple $(\Psi_j)_{j \in J}$, in which each Ψ_j is a $\mathcal{Q}$-formula, such that the realizations $\Gamma_j = \mathcal{R}(\Psi_j, \mathrm{R}^k)$ have the following properties:
   (a) For each $j \in J$, $\Gamma_j \subset S$ and Γ_j is semi-algebraically homeomorphic to $\mathbf{S}^1$;
   (b) the inclusion map $\Gamma_j \hookrightarrow S$ induces an injective map $\mathbb{F} \cong \mathrm{H}_1(\Gamma_j) \to \mathrm{H}_1(S)$, whose image we denote by $[\Gamma_j]$;
   (c) the tuple $([\Gamma_j])_{j \in J}$ forms a basis of $\mathrm{H}_1(S)$.

```
Procedure:
Use Algorithm 2 (Big enough radius) with input $(\mathcal{P}, \Phi)$ and let $r = \frac{a}{b} > 0, a, b \in \mathrm{D}$ be the output.
$\mathcal{P} \leftarrow \mathcal{P} \cup \{b^2(X_1^2 + \cdots + X_k^2) - a^2\}$.
$\Phi \leftarrow \Phi \wedge (b^2(X_1^2 + \cdots + X_k^2) - a^2 \leq 0)$.
Call Algorithm 4 (Computing one-dimensional subset) with input $(\mathcal{P}, \Phi)$ to obtain
    (a) a finite set $\mathcal{U}$ of real univariate representations over $\mathcal{T}$;
    (b) a finite indexing set $I$ and a finite tuple $(\gamma_j)_{j \in I}$ where each $\gamma_j$ a curve segment representation over $\mathcal{T}$;
    (c) mappings $L, R : I \to \mathcal{U}$, defined by $\mathrm{ass}(L(j))$ is the left endpoint of $\gamma_j$, and $\mathrm{ass}(R(j))$ is the right endpoint of $\gamma_j$.
Using Algorithm 5 (Conversion of curve segment representations to closed formulas) compute for each $j \in I$ a set of polynomials $\mathcal{Q}_j$ and a $\mathcal{Q}_j$-closed formula $\Theta_j$ such that $\mathcal{R}(\Theta_j) = \mathrm{clos}(\mathrm{ass}(\gamma_j))$.
$\mathcal{Q} \leftarrow \bigcup_{j \in I} \mathcal{Q}_j$.
$G \leftarrow (E = I, V = \mathcal{U}, \mathrm{head} = L, \mathrm{tail} = R)$.
Using a graph traversal algorithm compute a tuple $(C_1, \ldots, C_N)$ where each $C_h = (i_{h,0}, \ldots, i_{h,q_h-1}) \in I^{q_h}$ and
    (a) $\mathrm{tail}(i_{j,h}) = \mathrm{head}(i_{j,h+1 \bmod q_j}), h = 0, \ldots, q_j - 1$.
    (b) $C_1, \ldots, C_N$ are simple cycles of $G$.
    (c) The cycles $C_1, \ldots, C_N$ form a basis of the cycle space of $G$ (which is isomorphic to $\mathrm{H}_1(|G|)$).
for $1 \leq h \leq N$ do
    $\Psi_h \leftarrow \Theta_{i_{h,0}} \vee \cdots \vee \Theta_{i_{h,q_h-1}}$.
end for
Call Algorithm 6 (Simplicial replacement) with input:
    1. $\mathcal{Q}$,
    2. the tuple of $\mathcal{Q}$-closed formulas $\mathbf{\Phi} = (\phi_0, \ldots, \phi_N) = (\Psi_1, \ldots, \Psi_N, \Phi)$,
    3. $\ell = 1$
  to obtain a simplicial complex $\Delta_1(\mathbf{\Phi})$ such that
      $$(J \mapsto |\Delta_1(\mathbf{\Phi}|_J)|)_{J \subset [N]}$$
  is homologically 1-equivalent (cf. Definition 9) to $\mathbf{Simp}^{[N]}(\mathcal{R}(\mathbf{\Phi}))$ (cf. Notation 11).
Using Gauss-Jordan elimination identify a minimal subset $J \subset \{1, \ldots, N\}$ such that the $\mathrm{span}(\{\mathrm{Im}(\mathrm{H}_1(\Delta_1(\mathbf{\Phi}|_{\{h\}})) \to \mathrm{H}_1(\Delta_1(\mathbf{\Phi}))) \mid h \in J\}) = \mathrm{H}_1(\Delta_1(\mathbf{\Phi}))$.
Output $(\Psi_h)_{h \in J}$.
Complexity: The complexity of each $\Psi_h, h \in J$ is bounded by $(sd)^{O(k^2)}$, and the complexity of the algorithm is bounded by $(sd)^{k^{O(1)}}$, where $s = \mathrm{card}(\mathcal{P})$ and $d = \max_{P \in \mathcal{P}} \deg(P)$.
```

Proof of correctness of Algorithm 7. The correctness of Algorithm 7 follows from the correctness of Algorithms 2, 4, 5, and 6. □

Complexity analysis of Algorithm 7. The complexity upper bound is a consequence of the complexity analysis of the Algorithms 2, 4, 5, and 6. □

Proof of Theorem 2. Theorem 2 follows from the correctness and complexity analysis of Algorithm 7. □

Proof of Theorem 3. Theorem 3 follows from the correctness and complexity analysis of Algorithms 4 (Computing one-dimensional subset) and 5 (Conversion of curve segment representations to closed formulas). □

5. Conclusion and open problems

In this paper we have proved the existence of an algorithm with singly exponential complexity for obtaining a semi-algebraic basis of the first homology group of of a given closed semi-algebraic set (generalizing existing algorithm in the case of the zero-th dimensional homology). One obvious open problem is to generalize this result to the higher homology groups. As an intermediate problem we can ask for the solution to Conjecture 1. The techniques developed in the current paper may possibly generalize to prove Conjecture 1, but there are formidable technical problems to overcome and we leave this to future work.

Acknowledgement

We are grateful to the anonymous referees for their comments which helped to improve the paper. This research was supported by the National Science Foundation grant CCF-1910441.

Conflict of interest and data sharing

On behalf of all authors, the corresponding author states that there is no conflict of interest. Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

References

1. S. Basu, *On bounding the Betti numbers and computing the Euler characteristic of semi-algebraic sets*, Discrete Comput. Geom. **22** (1999), no. 1, 1–18. MR 1692627 (2000d:14061) 3
2. S. Basu, *Computing the first few Betti numbers of semi-algebraic sets in single exponential time*, J. Symbolic Comput. **41** (2006), no. 10, 1125–1154. MR 2262087 (2007k:14120) 3, 9, 10
3. S. Basu, R. Pollack, and M.-F. Roy, *On computing a set of points meeting every cell defined by a family of polynomials on a variety*, J. Complexity **13** (1997), no. 1, 28–37. MR 98d:14071 7
4. ______, *Computing roadmaps of semi-algebraic sets on a variety*, J. Amer. Math. Soc. **13** (2000), no. 1, 55–82. MR 1685780 (2000h:14048) 3, 7, 8, 11, 24

5. ______, *Computing the Euler-Poincaré characteristics of sign conditions*, Comput. Complexity **14** (2005), no. 1, 53–71. MR 2134045 (2006a:14095) 3
6. ______, *Algorithms in real algebraic geometry*, Algorithms and Computation in Mathematics, vol. 10, Springer-Verlag, Berlin, 2006 (second edition). MR 1998147 (2004g:14064) 4, 5, 6, 7, 10, 11, 13, 14, 16, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31, 32, 34, 35, 36, 37, 38
7. S. Basu, M.-F. Roy, M. Safey El Din, and É. Schost, *A baby step–giant step roadmap algorithm for general algebraic sets*, Found. Comput. Math. **14** (2014), no. 6, 1117–1172. MR 3273675 11
8. Saugata Basu, *Algorithms in real algebraic geometry: a survey*, Real algebraic geometry, Panor. Synthèses, vol. 51, Soc. Math. France, Paris, 2017, pp. 107–153. MR 3701212 3
9. Saugata Basu and Negin Karisani, *Efficient simplicial replacement of semialgebraic sets*, Forum of Mathematics, Sigma **11** (2023), e41. 9, 10, 17, 34, 38, 39
10. Saugata Basu and Negin Karisani, *Persistent homology of semi-algebraic sets*, SIAM J. Appl. Algebra Geom. (to appear). 19, 20
11. Saugata Basu, Richard Pollack, and Marie-Françoise Roy, *Computing the first Betti number of a semi-algebraic set*, Found. Comput. Math. **8** (2008), no. 1, 97–136. MR 2403532 3
12. Saugata Basu and Marie-Françoise Roy, *Divide and conquer roadmap for algebraic sets*, Discrete Comput. Geom. **52** (2014), no. 2, 278–343. MR 3249383 11
13. L. Blum, F. Cucker, M. Shub, and S. Smale, *Complexity and real computation*, Springer-Verlag, New York, 1998, With a foreword by Richard M. Karp. MR 1479636 (99a:68070) 4
14. P. Bürgisser, F. Cucker, and P. Lairez, *Computing the homology of basic semialgebraic sets in weak exponential time*, J. ACM **66** (2019), no. 1, Art. 5, 30, [Publication date initially given as 2018]. MR 3892564 4
15. P. Bürgisser, F. Cucker, and J. Tonelli-Cueto, *Computing the homology of semialgebraic sets. I: Lax formulas*, Found. Comput. Math. **20** (2020), no. 1, 71–118. MR 4056926 4
16. P. Bürgisser, F. Cucker, and J. Tonelli-Cueto, *Computing the Homology of Semialgebraic Sets. II: General formulas*, Found. Comput. Math. (2021). 4
17. John Canny, *Computing roadmaps of general semi-algebraic sets*, Comput. J. **36** (1993), no. 5, 504–514. MR 1234122 3, 7, 8, 11, 24
18. Henri Cartan and Samuel Eilenberg, *Homological algebra*, Princeton Landmarks in Mathematics, Princeton University Press, Princeton, NJ, 1999, With an appendix by David A. Buchsbaum, Reprint of the 1956 original. MR 1731415 17, 33
19. Tamal K. Dey, Tao Hou, and Sayan Mandal, *Computing minimal persistent cycles: polynomial and hard cases*, Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, PA, 2020, pp. 2587–2606. MR 4141340 3
20. Jeff Erickson and Kim Whittlesey, *Greedy optimal homotopy and homology generators*, Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, 2005, pp. 1038–1046. MR 2298363 3
21. L. Gournay and J.-J. Risler, *Construction of roadmaps in semi-algebraic sets*, Appl. Algebra Engrg. Comm. Comput. **4** (1993), no. 4, 239–252. MR 1235859 3, 7, 8, 11, 24
22. D. Yu. Grigor′ ev and N. N. Vorobjov, Jr., *Counting connected components of a semialgebraic set in subexponential time*, Comput. Complexity **2** (1992), no. 2, 133–186. MR 1190827 3, 7, 8, 11, 24
23. D. Yu. Grigoriev and N. N. Vorobjov, Jr., *Solving systems of polynomial inequalities in subexponential time*, J. Symbolic Comput. **5** (1988), no. 1-2, 37–64. MR 949112 (89h:13001) 7

24. Joos Heintz, Marie-Françoise Roy, and Pablo Solernó, *Single exponential path finding in semi-algebraic sets. II. The general case*, Algebraic geometry and its applications (West Lafayette, IN, 1990), Springer, New York, 1994, pp. 449–465. MR 1272047 3, 7, 8, 11, 24
25. J. Milnor, *On the Betti numbers of real varieties*, Proc. Amer. Math. Soc. **15** (1964), 275–280. MR 0161339 (28 #4547) 6
26. Madhav V. Nori, *Constructible sheaves*, Algebra, arithmetic and geometry, Part I, II (Mumbai, 2000), Tata Inst. Fund. Res. Stud. Math., vol. 16, Tata Inst. Fund. Res., Bombay, 2002, pp. 471–491. MR 1940678 9
27. Ippei Obayashi, *Volume-optimal cycle: tightest representative cycle of a generator in persistent homology*, SIAM J. Appl. Algebra Geom. **2** (2018), no. 4, 508–534. MR 3867608 3
28. I. G. Petrovskiĭ and O. A. Oleĭnik, *On the topology of real algebraic surfaces*, Izvestiya Akad. Nauk SSSR. Ser. Mat. **13** (1949), 389–402. MR 0034600 (11,613h) 6
29. J. J. Rotman, *An introduction to algebraic topology*, Springer-Verlag, 1988. 21
30. Jacob T. Schwartz and Micha Sharir, *On the "piano movers" problem. II. General techniques for computing topological properties of real algebraic manifolds*, Adv. in Appl. Math. **4** (1983), no. 3, 298–351. MR 712908 3
31. R. Thom, *Sur l'homologie des variétés algébriques réelles*, Differential and Combinatorial Topology (A Symposium in Honor of Marston Morse), Princeton Univ. Press, Princeton, N.J., 1965, pp. 255–265. MR 0200942 (34 #828) 6

Department of Mathematics, Purdue University, West Lafayette, IN 47907, U.S.A.

Email address: sbasu@math.purdue.edu

Department of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, MI 48824, U.S.A.

Email address: perciva9@msu.edu