



Recent Advances in the Computation of the Homology of Semialgebraic Sets

Felipe Cucker^(✉)

Department of Mathematics, City University of Hong Kong,
Kowloon Tong, Hong Kong
macucker@cityu.edu.hk

Abstract. This article describes recent advances in the computation of the homology groups of semialgebraic sets. It summarizes a series of papers by the author and several coauthors (P. Bürgisser, T. Krick, P. Lairez, M. Shub, and J. Tonelli-Cueto) on which a sequence of ideas and techniques were deployed to tackle the problem at increasing levels of generality. The goal is not to provide a detailed technical picture but rather to throw light on the main features of this technical picture, the complexity results obtained, and how the new algorithms fit into the landscape of existing results.

Keywords: Semialgebraic sets · Homology groups · Weak complexity

1 Semialgebraic Sets

Simply put, a *semialgebraic set* is a subset of \mathbb{R}^n which can be described as a Boolean combination of the solution sets of polynomial equalities and inequalities. But let's be more precise.

An *atomic* set is the solution set of one of the following five expressions

$$f(x) = 0, \quad f(x) \geq 0, \quad f(x) > 0, \quad f(x) \leq 0, \quad f(x) < 0, \quad (1)$$

where $f \in \mathbb{R}[X_1, \dots, X_n]$. A Boolean combination of subsets of \mathbb{R}^n is a set obtained by sequence of Boolean operations (unions, intersections and complements) starting from these subsets. For instance,

$$((A^c \cup B) \cap C)^c \cup (A \cap B) \quad (2)$$

(here S^c denotes the complement of S) is a Boolean combination of A, B, C .

There is a close relationship between the set-operations context of Boolean combinations and the syntax of formulas involving polynomials. Indeed, unions, intersections and complements of the sets defined in (1) correspond, respectively, to disjunctions, conjunctions and negations of the formulas defining these sets.

Partially supported by a GRF grant from the Research Grants Council of the Hong Kong SAR (project number CityU 11202017).

If, for instance, A, B, C are given by $f(x) > 0$, $g(x) = 0$ and $h(x) \leq 0$ then the set in (2) is given by the formula

$$\neg((\neg(f(x) > 0) \vee g(x) = 0) \wedge h(x) \leq 0) \vee (f(x) > 0 \wedge g(x) = 0).$$

A formula is *monotone* if it does not contain negations. Any formula can be rewritten into an equivalent monotone formula. A formula is *purely conjunctive* if it contains neither negations nor disjunctions. It is therefore just a conjunction of atomic formulas. The associated semialgebraic subsets of \mathbb{R}^n are said to be *basic*.

By definition, the class of semialgebraic sets is closed under unions, intersections and complements. It is much less obvious, but also true, that it is closed under projections. That is, if $\pi : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a projection map and $S \subset \mathbb{R}^n$ is semialgebraic then so is $\pi(S)$. It follows that if Ψ is a formula involving polynomials f_1, \dots, f_q in the variables X_1, \dots, X_n and $\bar{X}_1, \dots, \bar{X}_\ell$ are (pairwise disjoint) subsets of these variables then the set of solutions of the formula

$$\mathbf{Q}_1 \bar{X}_1 \mathbf{Q}_2 \bar{X}_2 \dots \mathbf{Q}_\ell \bar{X}_\ell \Psi \tag{3}$$

where $\mathbf{Q}_i \in \{\exists, \forall\}$, is semialgebraic as well. Note that this is a subset of \mathbb{R}^s where $s = n - (\bar{n}_1 + \dots + \bar{n}_\ell)$ with \bar{n}_i being the number of variables in \bar{X}_i . A well-known example is the following. Consider the set of points $(a, b, c, z) \in \mathbb{R}^4$ satisfying $az^2 + bz + c = 0$. This is, clearly, a semialgebraic subset of \mathbb{R}^4 . Its projection onto the first three coordinates is the set

$$\exists z (az^2 + bz + c = 0)$$

which is the same as

$$(a \neq 0 \wedge (b^2 - 4ac \geq 0)) \vee (a = 0 \wedge b \neq 0) \vee (a = 0 \wedge b = 0 \wedge c = 0)$$

which is a semialgebraic subset of \mathbb{R}^3 .

Some books with good expositions of semialgebraic sets are [6–8].

2 Some Computational Problems

Because of the pervasive presence of semialgebraic sets in all sorts of contexts, a variety of computational problems for these sets have been studied in the literature. In these problems, one or more semialgebraic sets are given as the data of the problem. The most common way to specify such a set $S \subseteq \mathbb{R}^n$ is by providing a tuple $f = (f_1, \dots, f_q)$ of polynomials in $\mathbb{R}[X_1, \dots, X_n]$ and a formula Ψ on these polynomials. We will denote by $S(f, \Psi)$ the semialgebraic set defined by the pair (f, Ψ) .

A few examples (but this list is by no means exhaustive) are the following.

Membership. Given (f, Ψ) and $x \in \mathbb{R}^n$, decide whether $x \in S(f, \Psi)$.

Feasibility. Given (f, Ψ) decide whether $S(f, \Psi) \neq \emptyset$.

Dimension. Given (f, Ψ) compute $\dim \mathbf{S}(f, \Psi)$.

Counting. Given (f, Ψ) compute the cardinality $|\mathbf{S}(f, \Psi)|$ if $\dim \mathbf{S}(f, \Psi) = 0$ (and return ∞ if $\dim \mathbf{S}(f, \Psi) > 0$ or 0 if $\mathbf{S}(f, \Psi) = \emptyset$).

Connected Components. Given (f, Ψ) compute the number of connected components of $\mathbf{S}(f, \Psi)$.

Euler Characteristic. Given (f, Ψ) compute the Euler characteristic $\chi(\mathbf{S}(f, \Psi))$.

Homology. Given (f, Ψ) compute the homology groups of $\mathbf{S}(f, \Psi)$.

Another example is the following.

Quantifier Elimination. Given (f, Ψ) and a quantifier prefix as in (3) compute polynomials g and a quantifier-free formula Ψ' such that the set of solutions of (3) is $\mathbf{S}(g, \Psi')$.

In the particular case when the union of $\overline{X}_1, \dots, \overline{X}_\ell$ is $\{X_1, \dots, X_n\}$ (in other words, when (3) has no free variables) the formula (3) evaluates to either **True** or **False**. The problem above has thus a natural subproblem.

Decision of Tarski Algebra. Given (f, Ψ) and a quantifier prefix without free variables decide whether (3) is true.

A number of geometric problems can be stated as particular cases of the Decision of Tarski Algebra. For instance the Feasibility problem is so, as deciding whether $\mathbf{S}(f, \Psi) \neq \emptyset$ is equivalent to decide the truth of $\exists x (x \in \mathbf{S}(f, \Psi))$. It is not difficult to see that deciding whether $\mathbf{S}(f, \Psi)$ is closed, or compact, or convex, etc. can all be expressed as particular cases of the Decision of Tarski Algebra.

3 Algorithms and Complexity

3.1 Symbolic Algorithms

In 1939 Tarski proved that the first-order theory of the reals was decidable (the publication [39] of this work was delayed by the war). His result was actually stronger; he gave an algorithm that solved the Quantifier elimination problem. At that time, interest was put on computability only. But two decades later, when focus had mostly shifted to complexity, it was observed that the cost of Tarski's procedure (that is, the number of arithmetic operations and comparisons performed) was enormous: a tower of exponentials. Motivated by this bound, Collins [16] and Wüthrich [43] independently devised an algorithm with a better complexity. Given $f = (f_1, \dots, f_q)$ their algorithm computes a *Cylindrical Algebraic Decomposition* (CAD) associated to f . Once with this CAD at hand, all the problems mentioned in Sect. 2 can be solved (with a cost polynomial in the size of the CAD). The cost of computing the CAD of f is

$$(qD)^{2^{O(n)}}$$

where D is the maximum of the degrees of the f_i . Whereas this doubly exponential bound is much better than the tower of exponentials for Tarski's algorithm,

it is apparent that this approach may be inefficient. Indeed, one must compute a CAD before solving any of the problems above and one would expect that not all of them require the same computational effort.

In the late 1980s Grigoriev and Vorobjov introduced a breakthrough, the *Critical Points Method*. Using new algorithmic ideas they showed that the feasibility problem could be solved with cost $(qD)^{n^{\mathcal{O}(1)}}$ [24] and, more generally, that the truth of a quantifier-free formula as in (3) could be decided with cost $(qD)^{\mathcal{O}(n^{4\ell})}$ [23]. The new algorithmic ideas were further refined to obtain sharper bounds for both deciding emptiness [4, 32] and eliminating quantifiers [4] but were also quickly applied to solve other problems such as, for instance, counting connected components [5, 14, 15, 25, 27], computing the dimension [28], the Euler-Poincaré characteristic [2], and the first few Betti numbers [3]. For all of these problems, singly exponential time algorithms were devised. But for the computation of homology groups only partial advances were achieved. The generic doubly exponential behavior of CAD remained the only choice.

All the algorithms mentioned above belong to what are commonly called *symbolic* algorithms. Although there is not an agreed upon definition for this class of algorithms, a characteristic feature of them is the use of exact arithmetic. Most of them consider input data to be arrays of integers (or rational numbers) and the size of these integers, as well as of all intermediate computations, is considered both to measure the input size and to define the cost of arithmetical operations. A formal model of such an algorithm is the Turing machine introduced by Turing in [40].

3.2 Numerical Algorithms

Shortly after the end of the war, Turing began working at the National Physics Laboratory in England. In his Turing Lecture [42] Wilkinson gives an account of this period. Central in this account is the story of how a linear system of 18 equations in 18 variables was solved using a computer and how, to understand the quality of the computed solution, Turing eventually came to write a paper [41] which is now considered the birth certificate of the modern approach to numerical algorithms. We will not enter into the details of this story (the reader is encouraged to read both Wilkinson's and Turing's papers). Suffice to say that the central issue here is that the underlying data (the coefficients of the linear system and the intermediate computations) are finite-precision numbers (such as the floating-point numbers dealt with by many computer languages). This implies that every operation—from the reading of the input data to each arithmetic operation—is affected by a small error. The problem is, the accumulation of these errors may end up on a very poor, or totally meaningless, approximation to the desired result. Turing realized that the magnitude of the final error depended on the quality of the input data, a quality that he measured with a *condition number*. This is a real number usually in the interval $[1, \infty)$. Data whose condition number is ∞ are said to be *ill-posed* and correspond to those inputs for which arbitrary small perturbations may have a qualitatively

different behavior with respect to the considered problem. For instance, non-invertible matrices viz linear equation solving (as arbitrary small perturbations may change the system from infeasible (no solutions) to feasible). Condition numbers were (and still are) eventually used both for numerical stability analysis (their original role) and for complexity analysis. A comprehensive view of this condition-based approach to algorithmic analysis can be found in [9].

Although there is not an agreed upon definition for the notion of numerical algorithm, a characteristic feature of them is a design with numerical stability in mind. It is not necessary that such an analysis will be provided with the algorithm but some justification of its expected merits in this regard is common. A cost depending on a condition number is also common. Sometimes it is this feature what distinguishes between a numerical algorithm and a symbolic one described in terms of exact arithmetic with data from \mathbb{R} .

Numerical algorithms (in this sense) were first developed in the context of linear algebra. In the 1990s Renegar, in a series of influential papers [33–35], introduced condition-based analysis in optimization while Shub and Smale were doing so for complex polynomial system solving [36]. Probably the first article to deal with semialgebraic sets was [19], where a condition number for the feasibility problem is defined and a numerical algorithm is exhibited solving the problem. Both the cost and the numerical stability of the algorithm were analyzed in terms of this condition number.

It is worth to describe the general idea of this algorithm. One first homogenizes all polynomials so that solutions are considered on the sphere \mathbb{S}^n instead of on \mathbb{R}^n . Then one constructs a grid \mathcal{G} of points on this sphere such that any point in \mathbb{S}^n is at a distance at most $\eta = \frac{1}{2}$ of some point in \mathcal{G} . At each point x in the grid one checks two conditions. The first one, involving the computation of $g(x)$ and the derivative $Dg(x)$ (for some polynomial tuples g), if satisfied, shows the existence of a point in $\mathcal{S}(f, \Psi)$. The second, involving only the computation of $g(x)$, if satisfied, shows that there are no points of $\mathcal{S}(f, \Psi)$ at a distance at most η to x . If the first condition is satisfied at some $x \in \mathcal{G}$ the algorithm stops and returns **feasible**. If the second condition holds for all $x \in \mathcal{G}$ the algorithm stops and returns **infeasible**. If none of these two events occurs the algorithm replaces $\eta \leftarrow \frac{\eta}{2}$ and repeats. The number of iterations depends on the condition number of the data f and it is infinite, that is, the algorithm loops forever, when the input is ill-posed. The cost of each iteration is exponential in n (because so is the number of points in the grid) and, hence, the total cost of the algorithm is also exponential in n . From a pure complexity viewpoint, a comparison with the symbolic algorithms for feasibility mentioned in Sect. 3.1 is far from immediate. But the motivating reasoning behind this algorithm was not pure complexity. One observes that the symbolic algorithms have ultimately to do computations with matrices of exponential size. In contrast, the numerical algorithm performs an exponential number of computations with polynomial size matrices. *As these computations are independent from one another one can expect a much better behavior under the presence of finite precision.*

3.3 Probabilistic Analyses

A shortcoming of a cost analysis in terms of a condition number is the fact that, unlike the size of the data, condition numbers are not known a priori. The established way of dealing with this shortcoming is to endow the space of data with a probability measure and to eliminate the condition number from the complexity (or numerical stability) bounds in exchange for a probabilistic reliance. We can no longer bound how long will it take to do a computation. We can only have some expectation with a certain confidence. This approach was pioneered by von Neumann and Goldstine [22] (who introduced condition in [30] along with Turing) and was subsequently advocated by Demmel [20] and Smale [37].

The most extended probabilistic analysis in the analysis of algorithms is the *average-case*. Cost bounds of this form show a bound on the expected value of the cost (as opposed to its worst-case). A more recent form of probabilistic analysis is the *smoothed analysis* of Spielman and Teng. This attempts to extrapolate between worst-case and average-case by considering the worst case over the data a of the expectation of the quantity under study over random perturbations of a (see [38] for a vindication of this analysis). Even more recently a third approach, called *weak analysis*, was proposed in [1] by Amelunxen and Lotz. Its aim is to give a theoretical explanation of the efficiency in practice of numerical algorithms whose average complexity is too high. A paradigm of this situation is the power method to compute the leading eigenpair of a Hermitian matrix: this algorithm is very fast in practice, yet the average number of iterations it performs has been shown to be infinite [29]. Amelunxen and Lotz realized that here, as in many other problems, this disagreement between theory and practice is due to the presence of a vanishingly small (more precisely, having a measure exponentially small with respect to the input size) set of outliers, outside of which the algorithm can be shown to be efficient. They called *weak* any complexity bound holding outside such small set of outliers. One can prove, for instance, that the numerical algorithm for the feasibility problem in [19] has weak single exponential cost.

4 Computing the Homology of Semialgebraic Sets

It is against the background of the preceding sections that we can describe some recent progress in the computation of homology groups of semialgebraic sets. As the algorithms behind this progress are numerical a good starting point for its discussion is understanding ill-posedness and condition for the problem.

4.1 Ill-Posedness and Condition

Recall, our data is a pair (f, Ψ) where $f = (f_1, \dots, f_q)$ is a tuple of polynomials. Consider, to begin with, the case where $q \leq n$ and Ψ corresponds to $f = 0$. This equality defines a real algebraic subset V of \mathbb{R}^n . It is well known that in general, if V is non-singular then sufficiently small perturbation in the coefficients of

f will define an algebraic set with the same topology as V . The one exception to this fact is the case where the projective closure of V intersects the hyperplane at infinity in a non-transversal manner. If V is singular, in contrast, arbitrary small perturbations can change its topology. The simplest instance of this occurs when considering a degree two polynomial in one variable. In this case, its zero set is singular if and only if it consists of a single point (which is a double zero). Arbitrarily small perturbation of the coefficients can change the discriminant of the polynomial to either negative (V becomes empty) or positive (the number of connected components of V becomes two). A similar behavior can be observed when the intersection of V with the hyperplane at infinity is singular. The simplest example is now given by a degree two polynomial in two variables. The intersection above is singular if and only if V is a parabola. Again, arbitrarily small perturbations may change V into a hyperbola (and hence increase the number of connected components) or into an ellipse (which, unlike the parabola, is not simply connected).

We conclude that f is ill-posed when either its zero set in \mathbb{R}^n has a singularity or the zero set of (f^h, X_0) in \mathbb{P}^n has one. Here f^h is the homogenization of f with respect to the variable X_0 and \mathbb{P}^n is n -dimensional projective space. This is equivalent to say that the zero set of either f^h or (f^h, X_0) has a singular point in \mathbb{P}^n .

Exporting this understanding from algebraic sets to closed semialgebraic sets is not difficult. One only needs to notice that when a closed semialgebraic set (given by a lax formula Ψ) has a change in topology due to a perturbation, there is a change of topology in its boundary as well. As this boundary is the union of (parts of) zero sets of subsets of f it follows that we can define the *set of ill-posed tuples* f to be

$$\Sigma := \left\{ f \in \mathbb{R}[X_1, \dots, X_n] \mid \exists J \subset \{0, 1, \dots, q\}, |J| \leq n+1, \exists \xi \in \mathbb{P}^n \right. \\ \left. (f_j^h(\xi) = 0 \text{ for } j \in J \text{ and } \text{rank } D_\xi f_j^h < |J|) \right\}$$

where $f_0^h := X_0$ and $f_j^h = (f_j^h)_{i \in J}$. One can then show that a tuple f is in Σ if and only if for some lax formula Ψ there exist arbitrarily small perturbations of f changing the topology of $S(f, \Psi)$.

In [12] a condition number $\kappa_{\text{aff}}(f)$ is defined that has several desirable features. Notably among them, the inequality (see [12, Thm. 7.3])

$$\kappa_{\text{aff}}(f) \leq 4D \frac{\|f\|}{d(f, \Sigma)}$$

where $\| \cdot \|$ is the Weyl norm and d is its induced distance, and the following result [12, Prop. 2.2].

Proposition 1. *There is an algorithm κ -ESTIMATE that, given $f \in \mathbb{R}[X_1, \dots, X_n]$, returns a number K such that*

$$0.99\kappa_{\text{aff}}(f) \leq K \leq \kappa_{\text{aff}}(f)$$

if $\kappa_{\text{aff}}(f) < \infty$ or loops forever otherwise. The cost of this algorithm is bounded by $(qnD\kappa_{\text{aff}}(f))^{\mathcal{O}(n)}$. \square

The condition number κ_{aff} is the last incarnation in a sequence of condition numbers that were defined for a corresponding sequence of problems: the Turing condition number for linear equation solving, the quantity $\mu_{\text{norm}}(f)$ defined by Shub and Smale [36] for complex polynomial equation solving, the value $\kappa(f)$ defined for the problem of real equation solving [17], and the condition number $\kappa^*(f; g)$ for the problem of computing the homology of *basic* semialgebraic sets.

4.2 Main Result

We can finally describe the main result in this overview. To do so, we first define a few notions we have been vague about till now.

Let $\mathbf{d} := (d_1, \dots, d_q)$ be a degree pattern and $\mathcal{P}_{\mathbf{d}}[q]$ be the linear space of polynomial tuples $f = (f_1, \dots, f_q)$ with $f_i \in \mathbb{R}[X_1, \dots, X_n]$ of degree at most d_i . We endow $\mathcal{P}_{\mathbf{d}}[q]$ with the Weyl inner product (see [9, §16.1] for details) and its induced norm and distance. This norm allows us to further endow $\mathcal{P}_{\mathbf{d}}[q]$ with the standard Gaussian measure with density

$$\rho(f) = \frac{1}{(2\pi)^{\frac{N}{2}}} e^{-\frac{\|f\|^2}{2}}.$$

Here $N := \dim \mathcal{P}_{\mathbf{d}}[q]$. Note that, as the condition number $\kappa_{\text{aff}}(f)$ is scale invariant, i.e., it satisfies $\kappa_{\text{aff}}(f) = \kappa_{\text{aff}}(\lambda f)$ for all real $\lambda \neq 0$, the tail $\text{Prob}\{\kappa_{\text{aff}}(f) \geq t\}$ has the same value when f is drawn from the Gaussian above and when it is drawn from the uniform distribution on the sphere $\mathbb{S}(\mathcal{P}_{\mathbf{d}}[q]) = \mathbb{S}^{N-1}$.

Theorem 1. *We exhibit a numerical algorithm `HOMOLOGY` that, given a tuple $f \in \mathcal{P}_{\mathbf{d}}[q]$ and a Boolean formula Φ over p , computes the homology groups of $\mathbb{S}(f, \Phi)$. The cost of `Homology` on input (f, Φ) , denoted $\text{cost}(f, \Phi)$, that is, the number of arithmetic operations and comparisons in \mathbb{R} , satisfies:*

$$\text{cost}(f, \Phi) \leq \text{size}(\Phi)q^{\mathcal{O}(n)}(nD\kappa_{\text{aff}}(p))^{\mathcal{O}(n^2)}.$$

Furthermore, if f is drawn from the Gaussian distribution on $\mathcal{P}_{\mathbf{d}}[q]$ (or, equivalently, from the uniform distribution on \mathbb{S}^{N-1}), then

$$\text{cost}(f, \Phi) \leq \text{size}(\Phi)q^{\mathcal{O}(n)}(nD)^{\mathcal{O}(n^3)}$$

with probability at least $1 - (nqD)^{-n}$. The algorithm is numerically stable. \square

The algorithm `HOMOLOGY` uses the same broad idea of the algorithm in [19] for deciding feasibility: it performs some simple computations on the (exponentially many) points of a grid in the sphere \mathbb{S}^n . The mesh of this grid, determining how many points it contains, is a function of $\kappa_{\text{aff}}(f)$.

Summarizing to the extreme, the algorithm proceeds as follows. Firstly, we homogenize the data. That is, we pass from $f = (f_1, \dots, f_q)$ to $f^h := (f_0^h, f_1^h, f_q^h)$ (recall, $f_0^h = X_0$) and from Ψ to $\Psi^h := \Psi \wedge (f_0^h > 0)$. Let $S_{\mathbb{S}}(f^h, \Psi^h)$ denote the solutions in \mathbb{S}^n for the pair (f^h, Ψ^h) . We then have the isomorphism of homology

$$H_*(S(f, \Psi)) \simeq H_*(S_{\mathbb{S}}(f^h, \Psi^h)).$$

Secondly, we use κ -ESTIMATE (recall Proposition 1) to approximate $\kappa_{\text{aff}}(f)$. With this approximation at hand we construct a sufficiently fine grid \mathcal{G} on the sphere. In the basic case, that is when Ψ is purely conjunctive, a subset $\mathcal{X} \subset \mathcal{G}$ and a real ε are constructed such that we have the following isomorphism of homology

$$H_*(S_{\mathbb{S}}(f^h, \Psi^h)) \simeq H_*(\cup_{x \in \mathcal{X}} B(x, \varepsilon)) \quad (4)$$

where $B(x, \varepsilon)$ is the open ball centered at x with radius ε . The Nerve Theorem [26, Corollary 4G.3] then ensures that

$$H_*(\cup_{x \in \mathcal{X}} B(x, \varepsilon)) \simeq H_*(\mathfrak{C}_{\varepsilon}(\mathcal{X})) \quad (5)$$

where $\mathfrak{C}_{\varepsilon}(\mathcal{X})$ is the Čech complex associated to the pair $(\mathcal{X}, \varepsilon)$. And computing the homology of a Čech complex is a problem with known algorithms. The cost analysis of the resulting procedure in terms of $\kappa_{\text{aff}}(f)$ —the first bound in Theorem 1—easily follows. The second bound in this theorem relies on a bound for the tail $\text{Prob}\{\kappa_{\text{aff}}(f) \geq t\}$ which follows from the geometry of Σ (its degree as an algebraic set) and the main result in [11].

This is the roadmap followed in [10]. A key ingredient in this roadmap is a bound of the radius of injectivity (or reach) of an algebraic manifold in terms of its condition number obtained in [18]. This bound allows one to get a constructive handle on a result by Niyogi, Smale and Weinberger [31, Prop. 7.1] leading to (4) and, more generally, it allowed to obtain a version of Theorem 1 in [18] for the case of smooth projective sets.

To obtain a pair $(\mathcal{X}, \varepsilon)$ satisfying (4) when Ψ is not purely conjunctive appears to be difficult. The equivalence in (4) follows from showing that, for a sufficiently small $s > 0$, the nearest-point map from the tube around $S_{\mathbb{S}}(f^h, \Psi^h)$ of radius s onto $S_{\mathbb{S}}(f^h, \Psi^h)$ is a retraction. This is no longer true if Ψ is not purely conjunctive.

In the case that Ψ is lax the problem was thus approached differently [12]. For every $\alpha \in \{\leq, =, \geq\}$ and every $i = 0, \dots, q$, let $\mathcal{X}_i^{\alpha} \subset \mathcal{G}$ and ε be as above, so that (4) holds with Ψ^h the atomic formula $f_i^h \alpha 0$. To each pair $(\mathcal{X}_i^{\alpha}, \varepsilon)$ we associate the Čech complex $\mathfrak{C}_{\varepsilon}(\mathcal{X}_i^{\alpha})$. Then, the set $S_{\mathbb{S}}(f^h, \Psi^h)$ and the complex

$$\Psi(\mathfrak{C}_{\varepsilon}(\mathcal{X}_0^{\leq}), \mathfrak{C}_{\varepsilon}(\mathcal{X}_0^=), \mathfrak{C}_{\varepsilon}(\mathcal{X}_0^{\geq}), \dots, \mathfrak{C}_{\varepsilon}(\mathcal{X}_q^{\leq}), \mathfrak{C}_{\varepsilon}(\mathcal{X}_q^=), \mathfrak{C}_{\varepsilon}(\mathcal{X}_q^{\geq}))$$

have the same homology groups. This complex is recursively built from the complexes $\mathfrak{C}_{\varepsilon}(\mathcal{X}_i^{\alpha})$ in the same manner (i.e., using the same sequence of Boolean operations) that $S_{\mathbb{S}}(f^h, \Psi^h)$ is built from the atomic sets $f_i^h(x) \alpha 0$. One can then algorithmically proceed as in [10]. The proof of this homological equivalence is

far from elementary; it relies on an inclusion-exclusion version of the Mayer-Vietoris Theorem and on the use of Thom’s first isotopy lemma on a convenient Whitney stratification of $S_{\mathbb{S}}(f^h, \Psi^h)$.

The extension to arbitrary (i.e., not necessarily lax) formulas has been recently done in [13]. It reduces this case to that of lax formulas through a construction of Gabrielov and Vorobjov [21]. This construction, however, was purely qualitative: the existence of a finite sequence of real numbers satisfying a desirable property was established, but no procedure to compute this sequence was given. The core of [13] consists of a quantitative version of Gabrielov and Vorobjov construction. Maybe not surprisingly, a distinguished role in this result is again played by the condition of the data: indeed, this sequence can be taken to be any sequence provided its largest element is less than $\frac{1}{\sqrt{2} \kappa_{\text{aff}}(f)}$.

4.3 Final Remarks

It is worth to conclude with some caveats about what exactly are the merits of Theorem 1. This can only be done by comparing this result with the computation of homology groups via a CAD described in Sect. 3.1, a comparison which is delicate as these algorithms are birds of different feather.

A first remark here is that, in the presence of finite precision, the CAD will behave appallingly whereas HOMOMOLOGY is likely do behave much better. With this in mind, let us assume that arithmetic is infinite-precision and focus in this case.

The main virtue of HOMOMOLOGY is the fact that, outside a set of exponentially small measure, it has a single exponential cost. Hence, outside this negligibly small set, it runs exponentially faster than CAD. Inside this set of data, however, it can take longer than CAD and will even loop forever for ill-posed data. But this shortcoming has an easy solution: one can run “in parallel” both HOMOMOLOGY and CAD and halt whenever one of them halts. This procedure results in a weak singly exponential cost with a doubly exponential worst-case.

Once said that, these considerations are of a theoretical nature. In practice, the quality of the bounds is such that only “small” data (i.e., polynomials in just a few variables) can be considered. In this case the difference between the two cost bounds depends much on the constants hidden in the big \mathcal{O} notation.

References

1. Amelunxen, D., Lotz, M.: Average-case complexity without the black swans. *J. Complexity* **41**, 82–101 (2017)
2. Basu, S.: On bounding the Betti numbers and computing the Euler characteristic of semi-algebraic sets. In: *Proceedings of the Twenty-eighth Annual ACM Symposium on the Theory of Computing* (Philadelphia, PA, 1996), pp. 408–417. ACM, New York (1996)
3. Basu, S.: Computing the first few Betti numbers of semi-algebraic sets in single exponential time. *J. Symbolic Comput.* **41**(10), 1125–1154 (2006)

4. Basu, S., Pollack, R., Roy, M.F.: On the combinatorial and algebraic complexity of quantifier elimination. *J. ACM* **43**, 1002–1045 (1996)
5. Basu, S., Pollack, R., Roy, M.F.: Computing roadmaps of semi-algebraic sets on a variety. *J. AMS* **33**, 55–82 (1999)
6. Basu, S., Pollack, R., Roy, M.F.: Algorithms in real algebraic geometry. *Algorithms and Computation in Mathematics*, vol. 10, 2nd edn. Springer, Berlin (2006)
7. Benedetti, R., Risler, J.J.: *Real Algebraic and Semi-algebraic Sets*, Hermann (1990)
8. Bochnak, J., Coste, M., Roy, M.F.: *Real Algebraic Geometry*. *Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge/A Series of Modern Surveys in Mathematics*, vol. 36. Springer, Heidelberg (1998). <https://doi.org/10.1007/978-3-662-03718-8>. Translated from the 1987 French original, Revised by the authors
9. Bürgisser, P., Cucker, F.: *Condition*. *Grundlehren der mathematischen Wissenschaften*, vol. 349. Springer, Heidelberg (2013). <https://doi.org/10.1007/978-3-642-38896-5>
10. Bürgisser, P., Cucker, F., Lairez, P.: Computing the homology of basic semialgebraic sets in weakly exponential time. *J. ACM* **66**(1), 5:1–5:30 (2019)
11. Bürgisser, P., Cucker, F., Lotz, M.: The probability that a slightly perturbed numerical analysis problem is difficult. *Math. Comput.* **77**, 1559–1583 (2008)
12. Bürgisser, P., Cucker, F., Tonelli-Cueto, J.: Computing the homology of semialgebraic sets. I: Lax formulas. *Found. Comput. Math.* [arXiv:1807.06435](https://arxiv.org/abs/1807.06435) (2018)
13. Bürgisser, P., Cucker, F., Tonelli-Cueto, J.: Computing the homology of semialgebraic sets. II: Arbitrary formulas (2019). preprint
14. Canny, J.: Computing roadmaps of general semi-algebraic sets. *Comput. J.* **36**(5), 504–514 (1993)
15. Canny, J., Grigorev, D., Vorobjov, N.: Finding connected components of a semialgebraic set in subexponential time. *Appl. Algebra Eng. Commun. Comput.* **2**(4), 217–238 (1992)
16. Collins, G.E.: Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In: Brakhage, H. (ed.) *GI-Fachtagung 1975*. LNCS, vol. 33, pp. 134–183. Springer, Heidelberg (1975). https://doi.org/10.1007/3-540-07407-4_17
17. Cucker, F.: Approximate zeros and condition numbers. *J. Complexity* **15**, 214–226 (1999)
18. Cucker, F., Krick, T., Shub, M.: Computing the homology of real projective sets. *Found. Comp. Math.* **18**, 929–970 (2018)
19. Cucker, F., Smale, S.: Complexity estimates depending on condition and round-off error. *J. ACM* **46**, 113–184 (1999)
20. Demmel, J.: The probability that a numerical analysis problem is difficult. *Math. Comput.* **50**, 449–480 (1988)
21. Gabriellov, A., Vorobjov, N.: Approximation of definable sets by compact families, and upper bounds on homotopy and homology. *J. Lond. Math. Soc.* **80**(1), 35–54 (2009)
22. Goldstine, H., von Neumann, J.: Numerical inverting matrices of high order, II. *Proc. AMS* **2**, 188–202 (1951)
23. Grigoriev, D.: Complexity of deciding Tarski algebra. *J. Symbolic Comput.* **5**, 65–108 (1988)
24. Grigoriev, D., Vorobjov, N.: Solving systems of polynomial inequalities in subexponential time. *J. Symbolic Comput.* **5**, 37–64 (1988)
25. Grigoriev, D., Vorobjov, N.: Counting connected components of a semialgebraic set in subexponential time. *Comput. Complexity* **2**, 133–186 (1992)
26. Hatcher, A.: *Algebraic Topology*. Cambridge University Press, Cambridge (2002)

27. Heintz, J., Roy, M.F., Solerno, P.: Single exponential path finding in semi-algebraic sets II: the general case. In: Bajaj, C. (ed.) *Algebraic Geometry and its Applications*, pp. 449–465. Springer, New York (1994). https://doi.org/10.1007/978-1-4612-2628-4_28
28. Koiran, P.: The real dimension problem is $NP_{\mathbb{R}}$ -complete. *J. Complexity* **15**, 227–238 (1999)
29. Kostlan, E.: Complexity theory of numerical linear algebra. *J. Comput. Appl. Math.* **22**, 219–230 (1988)
30. von Neumann, J., Goldstine, H.: Numerical inverting matrices of high order. *Bull. AMS* **53**, 1021–1099 (1947)
31. Niyogi, P., Smale, S., Weinberger, S.: Finding the homology of submanifolds with high confidence from random samples. *Discrete Comput. Geom.* **39**, 419–441 (2008)
32. Renegar, J.: On the computational complexity and geometry of the first-order theory of the reals. Part I. *J. Symbolic Comput.* **13**, 255–299 (1992)
33. Renegar, J.: Some perturbation theory for linear programming. *Math. Program.* **65**, 73–91 (1994)
34. Renegar, J.: Incorporating condition measures into the complexity theory of linear programming. *SIAM J. Optim.* **5**, 506–524 (1995)
35. Renegar, J.: Linear programming, complexity theory and elementary functional analysis. *Math. Program.* **70**, 279–351 (1995)
36. Shub, M., Smale, S.: Complexity of Bézout’s theorem I: geometric aspects. *J. AMS* **6**, 459–501 (1993)
37. Smale, S.: Complexity theory and numerical analysis. In: Iserles, A. (ed.) *Acta Numerica*, pp. 523–551. Cambridge University Press (1997)
38. Spielman, D., Teng, S.H.: Smoothed analysis: an attempt to explain the behavior of algorithms in practice. *Commun. ACM* **52**(10), 77–84 (2009)
39. Tarski, A.: *A Decision Method for Elementary Algebra and Geometry*. University of California Press, Berkeley (1951)
40. Turing, A.: On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.* **S2-42**, 230–265 (1936)
41. Turing, A.: Rounding-off errors in matrix processes. *Quart. J. Mech. Appl. Math.* **1**, 287–308 (1948)
42. Wilkinson, J.: Some comments from a numerical analyst. *J. ACM* **18**, 137–147 (1971)
43. Wüthrich, H.R.: Ein Entscheidungsverfahren für die Theorie der reell-abgeschlossenen Körper. In: Strassen, V., Specker, E. (eds.) *Komplexität von Entscheidungsproblemen Ein Seminar*. LNCS, vol. 43, pp. 138–162. Springer, Heidelberg (1976). https://doi.org/10.1007/3-540-07805-3_10