ROBUST CERTIFIED NUMERICAL HOMOTOPY TRACKING

CARLOS BELTRÁN AND ANTON LEYKIN

Abstract. We describe, for the first time, a completely rigorous homotopy (path–following) algorithm (in the Turing machine model) to find approximate zeros of systems of polynomial equations. If the coordinates of the input systems and the initial zero are rational our algorithm involves only rational computations and if the homotopy is well posed an approximate zero with integer coordinates of the target system is obtained. The total bit complexity is linear in the length of the path in the condition metric, and polynomial in the logarithm of the maximum of the condition number along the path, and in the size of the input.

1. Introduction

The research on solving systems of polynomial equations has experienced a rapid development in the last two decades, both from a theoretical and from a practical perspective. Many of the recent advances are based on the study of the general idea of homotopy continuation methods: let f be the system whose solutions we want to find, and let g be another system whose solutions we already know. Then, join g and f with a *homotopy*, that is a curve f_t in the vector space of polynomial systems, such that $f_0 = g$ and $f_1 = f$, and try to follow the curves (homotopy paths) produced as a solution ζ_0 of g is continued along the homotopy to a solution ζ_t of f_t. When t approaches 1, an approximation of a zero ζ_1 of f is obtained.

In order to describe such a method explicitly, we need two essential ingredients:

(1) A construction of the starting system g and the homotopy path f_t, and,
(2) once this path f_t is chosen, a procedure to approximate ζ_t (for a finite sequence of values of t starting with $t = 0$ and ending with $t = 1$).

The first of these two ingredients has been intensively studied from many perspectives, mainly using linear homotopy paths, i.e. once (g, ζ_0) is chosen, ζ_0 a solution of g, we just consider the path $f_t = (1 - t)g + tf$. In [42] a particularly simple choice of (g, ζ_0) was conjectured to be a good candidate for a initial pair, i.e. the complexity of homotopy methods with this starting pair could be polynomial on the average. This is still a challenging open conjecture that has been experimentally confirmed in [5]. In [6], [7], [8] it was proved that randomly chosen pairs (g, ζ_0) guarantee average polynomial complexity. In [13] a system whose zeros have coordinates equal to the roots of unity of appropriate degrees was proved to guarantee average quasi–polynomial complexity (polynomial for fixed degree.)

Date: October 22, 2012.

2010 *Mathematics Subject Classification.* 14Q20,65H20,68W30.

Key words and phrases. Symbolic–numeric methods, polynomial systems, complexity, condition metric, homotopy method, rational computation, computer proof.

C. Beltrán (corresponding author). Departamento de Matemáticas, Estadística y Computación, Universidad de Cantabria, Spain (beltranc@unican.es). Partially supported by MTM2010-16051, Spanish Ministry of Science (MICINN).

Anton Leykin. School of Mathematics, Georgia Tech, Atlanta GA, USA (leykin@math.gatech.edu). Partially supported by NSF grant DMS-0914802

Communicated by Peter Buergisser.

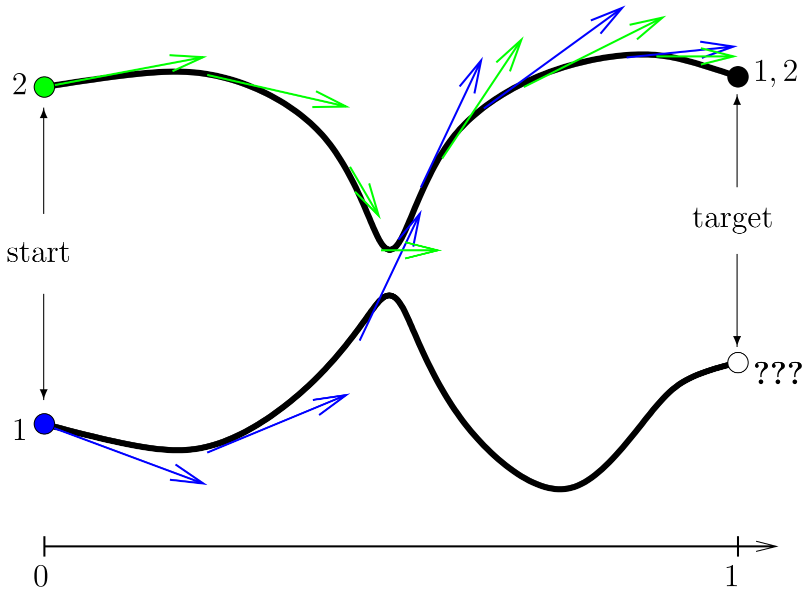

FIGURE 1. Path–jumping scenario: we start following the path corresponding to a certain solution of f_0 but we may jump to another path in the middle.

In this paper we deal with the second of the two questions above, restricting ourselves to the case when the homotopy path that is followed is regular, i.e., ζ_t is a regular zero of f_t for every $t \in [0, 1]$.

There exist several software packages which perform the path–following task of item (2) above (here is an incomplete list: Bertini [1], HOM4PS2 [24], NAG4M2 [27], and PHCpack [46]). In general, an initial step t_0 is chosen, and a *predictor* step (a numerical integration step of the differential equation $d(f_t(\zeta_t))/dt = 0$) is made to approximate ζ_{t_0}. A *corrector* step (several steps of Newton's method) is then used to get a better approximation of ζ_{t_0}. This process is repeated by choosing $t_1, t_2, \dots$ until $t = 1$ is reached. The software implementations mentioned above achieve spectacular practical results, with huge systems solved in a surprisingly short time. As a drawback, these fast methods make heuristic choices (notably the choice of t_i), which may introduce uncertainty in the quality of the solutions they provide: *how close to an actual zero is the given output? is the method actually following the path ζ_t or maybe a path–jumping occurred in the middle?*

In Figure 1 we illustrate a *path–jumping* phenomenon that may occur when a heuristic predictor–corrector path–tracking procedure is used.

In the problems with aim at computing all target solutions, this scenario can be detected simply by observing that two approximation sequences produce approximations to the same regular zero. The Kim–Smale α–test from [26], [44] can be used to make this task rigorously, see [20] for an implementation of that test. Once detected, this can be remedied by rerunning the heuristic procedure with tighter tolerances and higher precision of the computation.

However, an analysis of the end solutions would fail to detect the shortcomings of a heuristic method in the scenario with two approximation sequences "swapping" two paths as in Figure 2

In [42], a method which guarantees that path–jumping does not occur was shown for the first time, and its complexity (number of homotopy steps) was bounded above by a quantity depending on the maximum of the so–called *condition number* along the path (f_t, ζ_t), see (2.2) below. This result was recently improved in [39], changing the maximum of the condition number along the path to the length of the path (f_t, ζ_t) in the "condition metric" (see Section 2.4 for details). However, the result in [39] does not fully describe an algorithm, for the explicit choice of the steps t_i is not

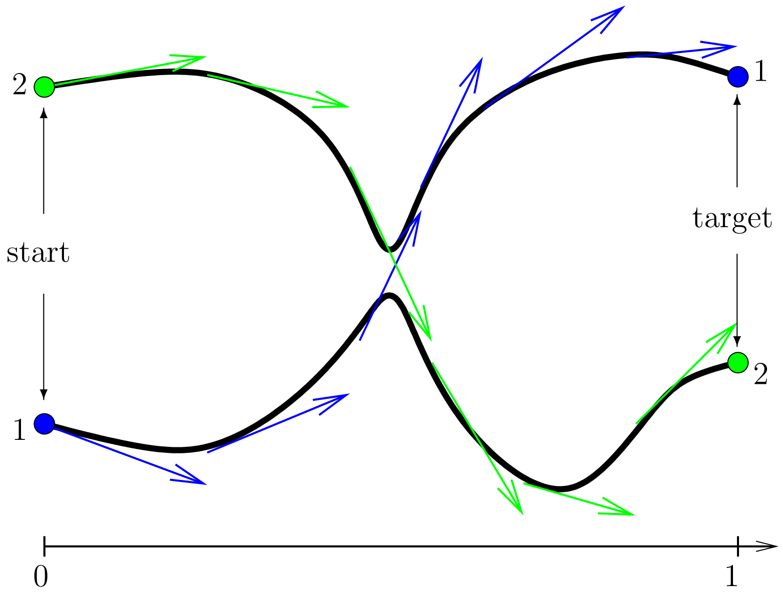

FIGURE 2. Path–swapping scenario: two path–jumps occur producing correct (although permuted) target solutions.

TABLE 1. The existing methods with certified output and complexity analysis for path–tracking. By arithmetic on $\mathbb{R}$ we mean that the BSS computation model [12] is assumed, that is exact arithmetical operations between real numbers are allowed. C.L. means length in the condition metric.

Method	Complexity	Arithmetic	Assumptions on f_t and comments
[39]	C.L.	$\mathbb{R}$	C^1 paths f_t. Not constructive.
[4]	C.L.	$\mathbb{R}$	C^{1+Lip} paths f_t.
[13]	$>\approx$C.L.	$\mathbb{R}$	Linear paths.
[17]	C.L.	$\mathbb{R}$	C^1 paths f_t with special properties.
This paper	C.L.	$\mathbb{Q}$	Linear paths.

given. Describing a way to actually choose these steps is a nontrivial task that can be done in several fashions. There are three independent papers doing this job: [4], [13], [17]. We briefly summarize in table 1 the properties of the algorithms in those papers and of the algorithm in this paper too. The proof of the algorithm in [13] is probably the shortest of the ones mentioned. As a small drawback, its complexity is not bounded above by the length of the path (f_t, ζ_t) in the condition metric, but by the integral of the square of the condition number along the path f_t, which is in general a (non–dramatically) greater quantity.

All methods in Table 1 are originally designed for systems of homogeneous polynomials, and then an argument like that in [7] is used to produce affine approximate zeros of the original system, if this one was not homogeneous.

Another difference from the heuristic methods described above is that there is no predictor step in these methods; only Newton's method is used (more exactly, projective Newton's method [38] described below): once t_0 is chosen, one obtains z_{t_0} as the result of (projective) Newton's method with base system f_{t_0} and base point $z_0 = \zeta_0$ (or z_0 an approximate zero of ζ_0). The idea is repeated, again, until $t = 1$ is reached.

Using the algorithm of [4], the main result of [39] reads as follows.

Theorem 1. [39], [4] *Assuming exact numerical computations, and assuming that the path f_t is a great circle in the sphere $\mathbb{S}$ in the space of homogeneous systems, the steps $t_0, t_1, \ldots$ can be chosen in such a way that the homotopy method outlined above produces an approximate zero of $f = f_1$ with associated exact zero ζ_1. The total number of steps is at most a small constant $71d^{3/2}$ (d the maximum of the degrees of the polynomials in f) times $\mathcal{C}_0 = \mathcal{C}_0(f_t, \zeta_t)$, that is the length of the path (f_t, ζ_t) in the so–called condition metric (if that length is infinity, the algorithm may never finish.)*

Here, the sphere $\mathbb{S}$ is the set of systems f such that $\|f\| = 1$ where $\|\cdot\|$ is the Bombieri–Weyl norm described below, and the condition metric is the usual product metric in $\mathbb{S}\times\mathbb{P}(\mathbb{C}^{n+1})$, point–wise multiplied by the condition number squared, see (2.3) below for a precise definition. The main result of [39], [4] also applies to paths which are not great circles, although the constant $71d^{3/2}$ may then vary.

In recent works [5], [27] we described a practical implementation of the "certified" method of [4] and used it in various experiments for estimating the average complexity of homotopy tracking.

As we have already pointed out, Theorem 1 (and the other existing algorithms for path–tracking) needs *exact numerical computations* on real numbers. More precisely, it assumes computations under the BSS model of computation, [12], [11]. While this assumption permits simplified analysis of the method, it is unrealistic in practice. Our main result here is to remove this assumption: if the input polynomials g and f and the initial approximate zero z_0 are given in (Gaussian) rational coordinates, then all the computations can be made over the rationals. We center our attention in linear paths, that is paths of the form $f_t = (1-t)g + tf$ where $t \in [0, 1]$. It is a natural fact that the condition number plays an important role in the translation of the real–number arithmetic results to rational arithmetic results. We will see that this just produces an extra factor (the logarithm of the maximum of the condition number along the path) in the complexity bounds. Our main result thus reads as follows.

Theorem 2 (Main)**.** *Assuming that g, f and z_0 are given in rational coordinates, and assuming the extra hypotheses (2.1) below, the homotopy method can be designed (see* TrackSegment *below) to produce an approximate zero z_* with integer coordinates of $f = f_1$ with associated exact zero ζ_1. The total number of steps is a small constant $C\sqrt{n}d^{3/2}$ (d the maximum of the degrees of the polynomials in f, and $n+1$ the number of variables) times $\mathcal{C}_0$. The bit complexity of the algorithm is linear in $\mathcal{C}_0$ and polynomial in the following quantities:*

- *n, S, d, h, where S is the number of nonzero monomials in the dense expansions of f and g and h is a bound for the bit length of the rational numbers appearing in the description of f, g, z_0,*
- $\log_2(\max_{t\in[0,1]}\{\mu(f_t, \zeta_t)\})$*, and*
- *The quantity*

$$\log_2\left(\max\left(1, \frac{\|f-g\|}{\min\{\|f_t\| : 0 \le t \le 1\}}\right)\right),$$

Here, $\|\cdot\|$ is Bombieri–Weyl norm (we recall its definition in Section 2.1) and $\mu(f_t, \zeta_t)$ is condition number at the pair (f_t, ζ_t), see (2.2) below. The bit size of the integer numbers in the coordinates of z_* is at most $O(\log_2(n) + \log_2 \mu(f, \zeta_1))$.

The extra hypotheses that will be specified in (2.1) is, in words, that the angle between g and $f - g$ is not too close to π, that is that the segment $\{f_t : t \in [0, 1]\}$ does not point too straight–forward to the origin.

The reader may note that, once this method is proved to work and programmed, it provides a status of mathematical proof to the path–following procedure. Moreover, its complexity does not differ much from the method of Theorem 1 and the size of the integer numbers involved is controlled.

In particular, the method can be used to give rigorous proofs to the results obtained via monodromy computation by algorithms in [28], [2]. Note that the implementations of the α-test carried out in [20], [31] would not be sufficient for the above applications even in the situations where the zero count is known and the α-test is capable of certifying the exact expected number of distinct zeroes at the ends of the homotopy paths. This is due to a potential multiple path–jumping (which we just call path–swapping) resulting in a wrong permutation of zeroes produced by tracking monodromy loops. In other words, as we have already pointed out, while in certain cases the α-test for the end solutions can resolve the scenario of Figure 1, it is powerless in the scenario of Figure 2.

1.1. **Previous works and historical remarks.**

Smale [43] proved the first results on the average complexity of polynomial zero–finding using Newton's method. The problem of solving systems of polynomial equations became later one of the cornerstones of complexity theory in the BSS model. In the first paragraph of [43] we read *"Also this work has the effect of helping bring the discrete mathematics of complexity theory of computer science closer to classical calculus and geometry"*. This paper is written with the intention of making another step in that direction. Theorem 2 establishes a strong link between the BSS model of computation and the classical Turing machine model, and we believe that this link strengthens both models at the same time. The Turing machine model is proved to accomplish a difficult task until now reserved to numerical solvers. An algorithm designed and analyzed in the BSS model is successfully translated (including its complexity analysis) to the classical model by carefully studying its condition number. Adding up, in the case of path–tracking methods for polynomial system solving,

$$\text{BSS model} \ + \ \text{condition number analysis} \ = \text{Turing machine model},$$

the equality being strong in the sense that the complexity of the discrete algorithm is similar to the complexity of the BSS algorithm. This "translation" of computational models can probably be done for many of the algorithms originally designed in the BSS model. The reader may find many of our techniques useful for such a task.

A natural precedent to our work is the algorithm in Malajovich's Ph.D. Theses [32] (see also [33], [34]) where a homotopy method for polynomial systems with coefficients in $\mathbb{Z}[\mathbf{i}]$ is presented. The algorithm in [32] is certified under reasonable assumptions for ε–machines (i.e. floating point machines), in which some intermediate computations are made. Its total complexity is bounded by a number which does not explicitly depend on the condition number of the systems found in the path. In [32], [34] there are some "gap theorems" which give universal bounds for the value of the condition number in the case that the coefficients of the polynomials are integers (and assuming that the solutions are not singular). The biggest advantage of that approach is that a global complexity bound is found. By "global" we mean that it is valid for solving every *generic* system; as a drawback, that complexity bound is exponential on the bit size of the entries.

In [14, Cor. 2.1], a universal upper bound for the bit size of rational approximate zeros of smooth zeros of systems with integer coefficients and degrees at most 2 is given. In general, one expects the condition number to be much smaller than in the worst case (see [40], [15] for results in this direction), and hence the bit size of the output of our algorithm will in general be much smaller than the upper bound of [14, Cor. 2.1].

One alternative to the rational arithmetic approach of this paper to the certification of homotopies could be using interval arithmetic. For example, in a more general setting, [25] proposes step control by means of isolating a homotopy path in a box around an approximation of a point on the path. While the implementation of [25] does not provide certification, in principle, interval arithmetic can be used in an attempt to certify homotopy tracking using similar isolation ideas (see [21] for an ongoing work in this direction).

1.2. **Acknowledgments.** In earlier stages of the ideas behind this work, we maintained many related conversations with Clement Pernet; thanks go to him for helpful discussions and comments. We also thank Gregorio Malajovich, Luis Miguel Pardo and Michael Shub for their questions and answers. Our beloved friend and colleague Jean Pierre Dedieu also inspired us in many occasions. The second author thanks Institut Mittag-Leffler for hosting him in the Spring semester of 2011. A part of this work was done while we were participating in several workshops related to Foundations of Computational Mathematics in the Fields Institute. We thank this institution for its kind support.

2. Technical background

2.1. **The vector space of polynomial systems.** As mentioned above, we will center our attention on homogeneous systems of equations. For fixed $n \geq 1$ and an integer $l \geq 1$, let $\mathcal{H}_l \subseteq \mathbb{C}[X_0, \ldots, X_n]$ be the vector space of degree l homogeneous polynomials with unknowns $X_0, \ldots, X_n$. As in [41], we consider Bombieri–Weyl's Hermitian product $\langle \cdot, \cdot \rangle_{\mathcal{H}_l}$ which preserves the orthogonality of different monomials and satisfies

$$\langle X_0^{\alpha_0} \cdots X_n^{\alpha_n}, X_0^{\alpha_0} \cdots X_n^{\alpha_n} \rangle_{\mathcal{H}_l} = \frac{\alpha_1! \cdots \alpha_n!}{l!}.$$

For integers $d_i \geq 1$, $1 \leq i \leq n$ and an n–tuple $(d) = (d_1, \ldots, d_n)$, we denote by $\mathcal{H}_{(d)}$ the vector space of systems of n homogeneous polynomials of respective degrees $d_1, \ldots, d_n$ in unknowns $X_0, \ldots, X_n$. That is,

$$\mathcal{H}_{(d)} = \mathcal{H}_{d_1} \times \cdots \times \mathcal{H}_{d_n}.$$

The Hermitian product in $\mathcal{H}_{(d)}$ is then

$$\langle f, g \rangle = \langle f_1, g_1 \rangle_{\mathcal{H}_{d_1}} + \cdots + \langle f_n, g_n \rangle_{\mathcal{H}_{d_n}}$$

where $f = (f_1, \ldots, f_n),\ g = (g_1, \ldots, g_n) \in \mathcal{H}_{(d)}$. We also define

$$\|f\| = \langle f, f \rangle^{1/2}.$$

The Hermitian product (norm) given by $\langle \cdot, \cdot \rangle$ ($\| \cdot \|$) in $\mathcal{H}_{(d)}$ is also called Bombieri–Weyl product (norm). It has a number of nice properties such as invariance under composition with a unitary change of coordinates, see for example [11, Section 12.1]. We denote

$$\mathbb{S} = \{f \in \mathcal{H}_{(d)} : \|f\| = 1\}.$$

Our main algorithm (Algorithm 1 below) will need to compute $\langle \cdot, \cdot \rangle$ and $\| \cdot \|^2$. This can obviously be done over the (complex) rationals $\mathbb{Q}[\mathbf{i}]$, if the polynomials involved have coordinates in $\mathbb{Q}[\mathbf{i}]$.

Remark 3. *The extra hypothesis Theorem 2 needs is*

$$-L_0 \|g\| \, \|f - g\| \leq \operatorname{Re} \langle g, f - g \rangle \leq \|g\| \, \|f - g\|, \tag{2.1}$$

where $L_0 = 1 - 10^{-3}$. While this hypothesis is not actually needed for the method to work (an output will equally be obtained if the hypothesis is not satisfied), it does simplify the complexity analysis of the algorithm significantly. Note that (2.1) means that the angle between the two vectors g and $f - g$ in the space of polynomial systems is not too close to π. This is satisfied by most reasonable choices of paths f_t.

2.2. **Projective Newton method.** We now describe the projective Newton method of [38]. Let $f \in \mathcal{H}_{(d)}$ and $z \in \mathbb{P}(\mathbb{C}^{n+1})$. Then,

$$N_{\mathbb{P}}(f)(z) = z - \left(Df(z) \mid_{z^{\perp}}\right)^{-1} f(z),$$

where $Df(z)$ is the $n \times (n+1)$ Jacobian matrix of f at $z \in \mathbb{P}(\mathbb{C}^{n+1})$, and

$$Df(z) \mid_{z^{\perp}}$$

is the restriction of the linear operator defined by $Df(z) : \mathbb{C}^{n+1} \to \mathbb{C}^n$ to the orthogonal complement $z^{\perp}$ of z. The reader may check that $N_{\mathbb{P}}(f)(\lambda z) = \lambda N_{\mathbb{P}}(f)(z)$, namely $N_{\mathbb{P}}(f)$ is a well–defined projective operator as long as the linear map $Df(z) \mid_{z^{\perp}}$ has an inverse. An equivalent expression, better suited for computations, is

$$N_{\mathbb{P}}(f)(z) = z - \begin{pmatrix} Df(z) \\ z^* \end{pmatrix}^{-1} \begin{pmatrix} f(z) \\ 0 \end{pmatrix},$$

where z^* is the conjugate transpose of z. We denote by $N_{\mathbb{P}}(f)^l(z)$ the result of l consecutive applications of $N_{\mathbb{P}}(f)$ on initial point z.

In general, one cannot expect that the solutions of systems of polynomials have rational coordinates. The goal of solvers is thus to produce rational points which are "close" in some sense to actual zeros. Following the approach of [44], [41] we will consider that a point is an "approximate zero" of a system of equations if it is in the strong (quadratic) basin of attraction of the projective Newton method. Namely:

Definition 4. *We say that $z \in \mathbb{P}(\mathbb{C}^{n+1})$ is an approximate zero of $f \in \mathcal{H}_{(d)}$ with associated (exact) zero $\zeta \in \mathbb{P}(\mathbb{C}^{n+1})$ if $N_{\mathbb{P}}(f)^l(z)$ is defined for all $l \geq 0$ and*

$$d_R(N_{\mathbb{P}}(f)^l(z), \zeta) \leq \frac{d_R(z, \zeta)}{2^{2^l - 1}}, \quad l \geq 0.$$

Here d_R is the Riemann distance in $\mathbb{P}(\mathbb{C}^{n+1})$, namely

$$d_R(z, z') = \arccos \frac{|\langle z, z' \rangle|}{\|z\| \, \|z'\|} \in [0, \pi/2],$$

where $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ are the usual Hermitian product and norm in $\mathbb{C}^{n+1}$. Note that $d_R(z, z')$ is the length of the shortest C^1 curve with extremes $z, z' \in \mathbb{P}(\mathbb{C}^{n+1})$, when $\mathbb{P}(\mathbb{C}^{n+1})$ is endowed with the usual Hermitian structure (see for example [11, Page 226].)

2.3. **The condition number.** The condition number at $(f, z) \in \mathcal{H}_{(d)} \times \mathbb{P}(\mathbb{C}^{n+1})$ introduced in [41] is defined as follows:

$$\mu(f, z) = \|f\| \, \|(Df(z) \mid_{z^{\perp}})^{-1} \mathrm{Diag}(\|z\|^{d_i - 1} d_i^{1/2})\|, \tag{2.2}$$

or $\mu(f, z) = \infty$ if $Df(z) \mid_{z^{\perp}}$ is not invertible. Here, $\|f\|$ is the Bombieri-Weyl norm of f and the second norm in the product is the operator norm of that linear operator. Note that $\mu(f, z)$ is, up to some normalizing factors, essentially equal to the operator norm of the inverse of the Jacobian $Df(z)$, restricted to the orthogonal complement of z. Sometimes μ is denoted μ_{norm} or μ_{proj}, but we keep the simplest notation here. One of the main properties of μ is[1] that it bounds the norm of the implicit function of the mapping $(f, z) \mapsto f(z)$. In other words, following [11, Sec. 12.3 and 12.4]:

[1] This property inspired its definition, see [41].

Lemma 5. *Let $(g,\zeta_0)\in\mathcal{H}_{(d)}\times\mathbb{P}(\mathbb{C}^{n+1})$ be such that $g(\zeta_0)=0$, and $\mu(g,\zeta_0)<\infty$. Let $f_t, t\in[0,\varepsilon)$ be a C^1 curve in $\mathcal{H}_{(d)}$, $f_0=g$. Then, for sufficiently small $t<\varepsilon$, ζ_0 can be continued to a zero ζ_t of f_t, that is there exists a C^1 curve $t\mapsto\zeta(t)\subseteq\mathbb{P}(\mathbb{C}^{n+1})$ such that $\zeta(0)=\zeta_0$ and, denoting $\zeta(t)=\zeta_t$, we have $f_t(\zeta_t)=0$ for every sufficiently small t. Moreover, the tangent vectors satisfy:*

$$\|\dot\zeta_0\|\leq\mu(g,\zeta_0)\|\dot f_0\|.$$

We will also use a variation of this condition number, namely χ_1 in equation (3.4) below.

The following result is a version of Smale's γ–theorem (cf. [44]), and follows from the study of the condition number in [41], [39].

Proposition 6. [4, Lemma 6] *Let $\zeta\in\mathbb{P}(\mathbb{C}^{n+1})$ be a zero of $f\in\mathcal{H}_{(d)}$ and let $z\in\mathbb{P}(\mathbb{C}^{n+1})$ be such that*

$$d_R(z,\zeta)\leq\frac{u_0}{d^{3/2}\mu(f,\zeta)},\qquad \textit{where } u_0=0.17586.$$

Then z is an approximate zero of f with associated zero ζ.

2.4. Complexity and the condition metric.

According to [39], the complexity (dominated by the number of Newton steps or number of while loops) of an algorithm performing the homotopy method should depend on the so–called condition length of the homotopy path. Given a path (h_t,ζ_t), $t_0\leq t\leq t_1$ where ζ_t is a zero of h_t and $h_t\in\mathbb{P}(\mathcal{H}_{(d)})$, $\zeta_t\in\mathbb{P}(\mathbb{C}^{n+1})$, the length of the path (h_t,ζ_t) in $\mathbb{P}(\mathcal{H}_{(d)})\times\mathbb{P}(\mathbb{C}^{n+1})$ is given by the integral

$$\int_{t_0}^{t_1}\left\|\frac{d}{dt}(h_t,\zeta_t)\right\|_{T_{(h_t,\zeta_t)}\left(\mathbb{P}(\mathcal{H}_{(d)})\times\mathbb{P}(\mathbb{C}^{n+1})\right)}dt$$
$$=\int_{t_0}^{t_1}\sqrt{\|\dot h_t\|^2_{T_{h_t}\mathbb{P}(\mathcal{H}_{(d)})}+\|\dot\zeta_t\|^2_{T_{\zeta_t}\mathbb{P}(\mathbb{C}^{n+1})}}\,dt$$

One must here understand $\dot h_t$ and $\dot\zeta_t$ as tangent vectors in $T_{h_t}\mathbb{P}(\mathcal{H}_{(d)})$ and $T_{\zeta_t}\mathbb{P}(\mathbb{C}^{n+1})$ respectively. If h_t and ζ_t are given in coordinates, this means:

$$\|\dot h_t\|^2_{T_{h_t}\mathbb{P}(\mathcal{H}_{(d)})}=\frac{\|\dot h_t\|^2}{\|h_t\|^2}-\frac{|\langle\dot h_t,h_t\rangle|^2}{\|h_t\|^4},$$
$$\|\dot\zeta_t\|^2_{T_{\zeta_t}\mathbb{P}(\mathbb{C}^{n+1})}=\frac{\|\dot\zeta_t\|^2}{\|\zeta_t\|^2}-\frac{|\langle\dot\zeta_t,\zeta_t\rangle|^2}{\|\zeta_t\|^4},$$

Now, the condition length (or length in the condition metric) of the same path is defined in [39] as

$$\int_{t_0}^{t_1}\mu(h_t,\zeta_t)\sqrt{\|\dot h_t\|^2_{T_{h_t}\mathbb{P}(\mathcal{H}_{(d)})}+\|\dot\zeta_t\|^2_{T_{\zeta_t}\mathbb{P}(\mathbb{C}^{n+1})}}\,dt.$$

Note that, from (2.2), $\mu(h_t,\zeta_t)$ only depends on the projective classes of h_t and ζ_t and this last integral is thus well defined.

Now, given a path $(f_t,\zeta_t)\in\mathcal{H}_{(d)}\times\mathbb{P}(\mathbb{C}^{n+1})$ where $f_t(\zeta_t)=0$, we define its condition length as

$$\mathcal{C}_0=\mathcal{C}_0(f_t,\zeta_t)=\int_{t_0}^{t_1}\mu(f_t,\zeta_t)\sqrt{\frac{\|\dot f_t\|^2}{\|f_t\|^2}-\frac{\operatorname{Re}\left(\langle\dot f_t,f_t\rangle\right)^2}{\|f_t\|^4}+\|\dot\zeta_t\|^2_{T_{\zeta_t}\mathbb{P}(\mathbb{C}^{n+1})}}\,dt.\tag{2.3}$$

That is, $\mathcal{C}_0(f_t,\zeta_t)$ is the length in the condition metric on $\mathbb{S}\times\mathbb{P}(\mathbb{C}^{n+1})$ of the path obtained by projecting $(f_t,\zeta_t)\in\mathcal{H}_{(d)}\times\mathbb{P}(\mathbb{C}^{n+1})$ on $\mathbb{S}\times\mathbb{P}(\mathbb{C}^{n+1})$. The reason of this change is that segments in $\mathcal{H}_{(d)}$ project nicely on the sphere $\mathbb{S}$ (indeed, they project onto pieces of great circles in $\mathbb{S}$), but they do not project so nicely on $\mathbb{P}(\mathcal{H}_{(d)})$. This makes our analysis easier and, presumably, has little effect in the complexity bounds.

Note that, if $f_t \in \mathcal{H}_{(d)}$ is the horizontal lift of $h_t \in \mathbb{P}(\mathcal{H}_{(d)})$ then the condition lengths of (f_t, ζ_t) and (h_t, ζ_t) coincide. In the general case, however, the condition length of (f_t, ζ_t) is greater than that of $(\pi_{\mathbb{P}(\mathcal{H}_{(d)})}(f_t), \zeta_t)$, because in general $\operatorname{Re}\left(\langle \dot{f}_t, f_t\rangle\right)^2 \leq |\langle \dot{f}_t, f_t\rangle|^2$.

3. A robust homotopy step

In this section we set up the backbone of our main algorithm: how to correctly choose a homotopy step. We do this by stating a theorem that, given sufficiently close polynomial systems $g, f \in \mathcal{H}_{(d)}$ and an approximate zero z_0 of g associated to an actual zero ζ_0 of g, guarantees that ζ_0 can be continued to a zero ζ_1 of f, and moreover a projective point sufficiently close (in a sense that we will precisely determine) to $N_{\mathbb{P}}(f)(z_0)$ is an approximate zero of f with associated zero ζ_1. There are some precedents to this result in [32] but our theorem is needed to get the sharp complexity bound of [39].

3.1. **Some constants.** As it is common in the explicit description of many numerical analysis algorithms, we will need to use some constants, that need to be described explicitly because they intervene in the definition of the algorithm. We will use a free parameter $1/2 < \delta < 1$ that will be set to $3/4$ in our implementation of the algorithm. The rest of the constants are:

$$u_0 = 0.17586 \text{ (the constant from Proposition 6)},$$

$$P = \sqrt{2} + \sqrt{4 + 5/8}, \qquad a = a_\delta = \frac{(2\delta - 1)u_0}{\sqrt{2} + 2\delta u_0} < \frac{1}{\sqrt{2}},$$

$$c' = c'_\delta = 1 - (1-a)^{\frac{P}{\sqrt{2}}} < 1. \tag{3.1}$$

Finally, let c be any number satisfying

$$c \leq c_\delta = \frac{(1 - \sqrt{2}u_0/2)^{\sqrt{2}}}{1 + \sqrt{2}u_0/2}\, c'. \tag{3.2}$$

Only the value of $c^2/(2P^2)$ will appear in the description of our main algorithm (see Algorithm TrackSegment below.) Hence, one can choose any value of $(c^2/2P^2) \in \mathbb{Q}$ such that

$$\frac{c^2}{2P^2} \leq \frac{c^2_{3/4}}{2P^2} = 0.00034412...$$

In our algorithm, we will choose $\frac{17}{50000} = 0.00034$. The reader may check that the following holds.

$$\frac{c'}{P(1-a)} \leq \frac{3\delta u_0}{2}, \qquad \frac{c'}{P} \leq \frac{3a}{2\sqrt{2}}. \tag{3.3}$$

3.2. **A version of the condition number.** The condition number $\mu(f, \zeta)$ of (2.2) above can be computed using a more amenable expression if $f(\zeta) = 0$.

Let $g, \dot{g} \in \mathcal{H}_{(d)}$ be two polynomial systems and let $z \in \mathbb{P}(\mathbb{C}^{n+1})$. Let $\chi_1 = \chi_1(g, z)$, $\chi_2 = \chi_2(g, \dot{g}, z)$ and $\varphi = \varphi(g, \dot{g}, z)$ be defined by

$$\chi_1 = \left\| \begin{pmatrix} Dg(z) \\ z^* \end{pmatrix}^{-1} \begin{pmatrix} \sqrt{d_1}\|g\|\|z\|^{d_1-1} & & & \\ & \ddots & & \\ & & \sqrt{d_n}\|g\|\|z\|^{d_n-1} & \\ & & & \|z\| \end{pmatrix} \right\|, \tag{3.4}$$

$$\chi_2 = \left(\|\dot{g}\|^2 + \frac{\|g\|^2}{\|z\|^2} \left\| \begin{pmatrix} Dg(z) \\ z^* \end{pmatrix}^{-1} \begin{pmatrix} \dot{g}(z) \\ 0 \end{pmatrix} \right\|^2 \right)^{1/2}, \tag{3.5}$$

$$\varphi = \chi_1\chi_2. \tag{3.6}$$

Note that these formulas do not depend on the representative of z and thus are well defined. Their value is also invariant under multiplication of g by a non–zero complex number $\lambda \in \mathbb{C}$.

It was noted in [4, eq. (2.2)] that if $t \mapsto (f_t, \zeta_t) \subseteq \mathbb{S} \times \mathbb{P}(\mathbb{C}^{n+1})$ is a C^1 curve such that $f_t(\zeta_t) = 0$, then

$$\chi_1(f_t, \zeta_t) = \mu(f_t, \zeta_t), \quad \varphi(f_t, \dot{f}_t, \zeta_t) = \mu(f_t, \zeta_t)\|(\dot{f}_t, \dot{\zeta}_t)\|, \quad \forall\, t. \tag{3.7}$$

Thus, $\chi_1(f, z)$ is a version of the condition number $\mu(f, z)$ (equal if $f(z) = 0$) and φ is, if $(f_t, \zeta_t) \subseteq \mathbb{S} \times \mathbb{P}(\mathbb{C}^{n+1})$ and $f_t(\zeta_t) = 0$, the quantity inside of the integral defining $\mathcal{C}_0$ in (2.3).

From Lemma 5, with the notation of (3.7) we have:

$$\varphi(f_0, \dot{f}_0, \zeta_0) \leq \mu(f_0, \zeta_0)\|\dot{f}_0\|\sqrt{1 + \mu(f, \zeta_0)^2}. \tag{3.8}$$

The reason to use χ_1 instead of μ and φ instead of simply the term inside the integral defining $\mathcal{C}_0$ is that the rates of change of χ_1 and φ are easy to analyze, see lemmas 17 and 18 in Section 10.

3.3. **A robust homotopy step.** We now state our main technical tool, which is a more detailed and complete version of Lemma 5 about continuation of zeros, designed to answer the following questions

- how long can a zero of g be continued when g is moved?
- how do χ_1 and φ vary in this process?
- if an approximate zero of g is given, how long will it still be an approximate zero as g is moved?

We will use the constants defined in Section 3.1. Given two systems $f, g \in \mathcal{H}_{(d)}$, we consider the Riemannian distance in the sphere $\mathbb{S}$ from $g/\|g\|$ to $f/\|f\|$, that is:

$$d_{\mathbb{S}}\left(\frac{g}{\|g\|}, \frac{f}{\|f\|}\right) = \arccos \frac{\mathrm{Re}\,\langle g, f\rangle}{\|g\|\,\|f\|}.$$

Theorem 7. *Let $g, f \in \mathcal{H}_{(d)}$ be two systems of polynomial equations such that $g \neq \lambda f$ $\forall \lambda \in \mathbb{R}$. Let z_0 be an approximate zero of g satisfying*

$$d_R(z_0, \zeta_0) \leq \frac{u_0}{2d^{3/2}\mu(g, \zeta_0)} \tag{3.9}$$

for some exact zero ζ_0 of g. Let

$$\dot{g} = \frac{\|g\|^2 f - \mathrm{Re}(\langle f, g\rangle) g}{\|g\|\sqrt{\|f\|^2\|g\|^2 - \mathrm{Re}(\langle f, g\rangle)^2}}.$$

That is, $\dot{g}$ is the derivative at 0 of the arc–length parametrized short portion of the great circle in $\mathbb{S}$, from $g/\|g\|$ to $f/\|f\|$. Let

$$\chi_1 = \chi_1(g, z_0), \quad \chi_2 = \chi_2(g, \dot{g}, z_0), \quad \varphi = \varphi(g, \dot{g}, z_0).$$

Assume that

$$d_{\mathbb{S}}\left(\frac{g}{\|g\|}, \frac{f}{\|f\|}\right) \leq \frac{c}{Pd^{3/2}\varphi}. \tag{3.10}$$

Then,

(1) *ζ_0 can be continued following the straight line homotopy*

$$f_t = (1 - t)g + tf \tag{3.11}$$

to a zero ζ_t of f_t, namely, there exists a C^1 curve $t \mapsto \zeta(t)$ such that $\zeta(0) = \zeta_0$ and, denoting $\zeta(t) = \zeta_t$, we have $f_t(\zeta_t) = 0$ for $t \in [0, 1]$.

(2) *We have the following inequality:*

$$\varphi \leq \frac{\sqrt{2}\mu(g,\zeta_0)^2}{(1-\sqrt{2}u_0/2)^{1+\sqrt{2}}}, \tag{3.12}$$

(3) *The condition length $\mathcal{C}_0(f_t,\zeta_t)$ of the path (f_t,ζ_t) as defined in (2.3) is essentially equal to $\varphi\, d_{\mathbb{S}}\left(\frac{f}{\|f\|},\frac{g}{\|g\|}\right)$. More exactly:*

$$\frac{(1-\sqrt{2}u_0/2)^{1+\sqrt{2}}\ln(1+c')}{c'} \leq \frac{\mathcal{C}_0(f_t,\zeta_t)}{\varphi d_{\mathbb{S}}\left(\frac{f}{\|f\|},\frac{g}{\|g\|}\right)} \leq \frac{1+\sqrt{2}u_0/2}{(1-\sqrt{2}u_0/2)^{\sqrt{2}}}. \tag{3.13}$$

(4) *For every $\tilde{z}\in\mathbb{P}(\mathbb{C}^{n+1})$ such that*

$$d_R(\tilde{z},N_{\mathbb{P}}(f)(z_0)) \leq \frac{(1-\delta)u_0}{2d^{3/2}(1+3\,\delta u_0/2)\chi_1} \tag{3.14}$$

we have that

$$d_R(\tilde{z},\zeta_1) \leq \frac{u_0}{2d^{3/2}\mu(f,\zeta_1)}. \tag{3.15}$$

In particular, $\tilde{z}$ is an approximate zero of f with associated zero ζ_1.

The proof of this theorem is a long and tedious computation. We delay it till Section 10.

4. A schematic description of the robust linear homotopy method

In this section we describe an algorithmic scheme for the linear homotopy method. The procedure in this section is not quite an algorithm, because we do not specify how to perform some of the tasks it requires. We will however prove that *any* actual algorithm designed to fit into the scheme of this section has certified output and the number of iterations it performs is essentially bounded by the condition length $\mathcal{C}_0$.

Let g,f be two non–collinear systems, that is $g\neq\lambda f$ for every $\lambda\in\mathbb{R}$. Let f_t be defined by (3.11), so $f_0=g$, $f_1=f$. Let $\delta\in\mathbb{Q}$, $R\in\sqrt{\mathbb{Q}}$ with $1/2<\delta<1$ and $R\geq\sqrt{2}$ be two arbitrary constants [2].

Algorithmic Scheme 1. $z_*=$ TrackSegment_Scheme(f,g,z_0)

```
Require: $f,g\in\mathcal{H}_{(d)}$ non–collinear with coefficients in $\mathbb{Q}[i]$;
    $z_0\in\mathbb{Q}[i]^{n+1}$ is an approximate zero of $g$ satisfying (3.9).
Ensure: $z_*\in\mathbb{Z}[i]^{n+1}$ is an approximate zero of $f$ associated to the end of the homotopy path
    starting at the zero of $g$ associated to $z_0$ and defined by the homotopy (3.11).
$i\leftarrow 0$; $s_i\leftarrow 0$.
while $s_i\neq 1$ do
    $g_i\leftarrow f_{s_i}$.
    Let
        $\dot{g}_i=\frac{\|g_i\|^2 f-\mathrm{Re}(\langle f,g_i\rangle)g_i}{\|g_i\|\sqrt{\|f\|^2\|g_i\|^2-\mathrm{Re}(\langle f,g_i\rangle)^2}}$.
    Let $\chi_{i,1}=\chi_{i,1}(g_i,z_i)$, $\chi_{i,2}=\chi_{i,2}(g_i,\dot{g}_i,z_i)$ and $\varphi_i=\varphi_i(g_i,\dot{g}_i,z_i)$ as defined in (3.4), (3.5) and
    (3.6).
```

[2] If $R\geq\sqrt{2}$ then values t_i as described in Algorithm 1 exist. They also exist for smaller values of $R>1$ like $R=1.0003$ but taking $R\geq\sqrt{2}$ will make our formulas look prettier. This assumption does not affect much to the running time of the algorithm. We will only use R^2 in the algorithm. Hence, we can take $R\in\sqrt{\mathbb{Q}}$.

```
    Let $t_i$ be any positive number such that
    $$L \leq \frac{\|g_i\|^2 + t_i \operatorname{Re}\langle g_i, f-g\rangle}{\|g_i\|\sqrt{\|g_i\|^2 + 2t_i \operatorname{Re}\langle g_i, f-g\rangle + t_i^2\|f-g\|^2}} \leq U_R \tag{4.1}$$
    where
    $$L = 1 - \frac{c^2}{2P^2d^3\varphi_i^2} + \frac{c^4}{24P^4d^6\varphi_i^4}, \qquad U_R = 1 - \frac{c^2}{2R^2P^2d^3\varphi_i^2}. \tag{4.2}$$
    If such $t_i$ does not exist, let $t_i \leftarrow 1$ (which will imply that the while loop finishes in this step).
    if $t_i > 1 - s_i$ then
      $t_i \leftarrow 1 - s_i$.
    end if
    $s_{i+1} \leftarrow s_i + t_i$;
    $\varepsilon \leftarrow \frac{(1-\delta)^2u_0^2}{4d^3(1+3\delta u_0/2)^2\chi_{i,1}^2}$.
    $z_{i+1} \leftarrow N_{\mathbb{P}}(g_{i+1})(z_i) = z_i - \left({}^{Dg_{i+1}(z_i)}_{z_i^*}\right)^{-1} g_{i+1}(z_i)$
    $\tilde{z}_{i+1} \leftarrow$ any vector in $\mathbb{Q}[\mathbf{i}]^{n+1}$ satisfying
    $$d_R(\tilde{z}_{i+1}, z_{i+1}) \leq \sqrt{\varepsilon} = \frac{(1-\delta)u_0}{2d^{3/2}(1+3\delta u_0/2)\chi_{i,1}}. \tag{4.3}$$
    $z_{i+1} \leftarrow \tilde{z}_{i+1}$
    $i \leftarrow i+1$.
end while
$z_* \leftarrow \tilde{z}_{i+1}$.
```

Note that (2.1) implies that for $i \geq 0$ we have

$$-L_0\|g_i\|\,\|f-g_i\| \leq \operatorname{Re}\langle g_i, f-g_i\rangle \leq \|g_i\|\,\|f-g_i\|, \tag{4.4}$$

where $L_0 = 1 - 10^{-3}$.

Theorem 8. *The output of any algorithm performing the instructions described in* TrackSegment_Scheme *is certified. Namely, for every $i \geq 0$, the point $\tilde{z}_i$ is an approximate zero of $g_i = f_{s_i}$, with associated zero* [3] *ζ_i, the unique zero of g_i such that (g_i, ζ_i) lies in the lifted path (f_t, ζ_t). Moreover,*

$$d_R(\tilde{z}_i, \zeta_i) \leq \frac{u_0}{2d^{3/2}\mu(g_i, \zeta_i)}, \quad i \geq 1.$$

Let $\mathcal{C}_0$ be defined by (2.3) and (3.11). If $\mathcal{C}_0 < \infty$, there exists $k \geq 0$ such that $f = g_k$. For the number of homotopy steps k the following bounds hold:

$$C'd^{3/2}\mathcal{C}_0 \leq k \leq \lceil Cd^{3/2}\mathcal{C}_0 \rceil,$$

where

$$C = \frac{c'RP}{c(1-\sqrt{2}u_0/2)^{1+\sqrt{2}}\ln(1+c')}, \qquad C' = \frac{P}{c'}.$$

In particular, if $\mathcal{C}_0 < \infty$, there exists a unique lift (f_t, ζ_t) of the path f_t, and the algorithm finishes and outputs z_*, an approximate zero of $f = g_k$ with associated zero ζ_k, the unique zero of f such that (f, ζ_k) lies in the lifted path (f_t, ζ_t).

Finally, the two following inequalities hold at every step of the algorithm:

$$\varphi_i \leq \frac{\sqrt{2}\mu(g_i, \zeta_i)^2}{(1-\sqrt{2}u_0/2)^{1+\sqrt{2}}}, \tag{4.5}$$

[3] Note the slight abuse of notation: we just use ζ_i the zero of $g_i = f_{s_i}$, so we should actually denote it by ζ_{s_i}.

$$U_R - L \geq \frac{\hat{c}}{d^3 \max_{t\in[0,1]}\{\mu(f_t,\zeta_t)^4\}}, \text{ where } \hat{c} \text{ is a universal constant.} \tag{4.6}$$

Remark 9. *As said above, we will choose $\delta = 3/4$ in our main algorithm* TrackSegment. *With that choice and the use of Frobenius norm instead of operator norm for the computation of $\chi_{i,1}$, in our practical implementation we will have*

$$28 \leq C' \leq C \leq 79\sqrt{n+1}.$$

Note that C is not a universal constant as it depends on n. The value of $\hat{c}$ is needed only for the bit–complexity analysis where it will be replaced by an $O(1)$. One can however estimate it as $\hat{c} \approx 0.00003$.

Proof. The proof of correctness of the algorithm is by induction on i. The base case of our induction $i = 0$ follows. Assume that

$$d_R(z_i, \zeta_i) \leq \frac{u_0}{2d^{3/2}\mu(g_i,\zeta_i)}. \tag{4.7}$$

We claim that we are under the hypotheses of Theorem 7. Indeed,

$$g_{i+1} = f_{s_{i+1}} = f_{s_i+t_i} = (1-s_i-t_i)\,g + (s_i+t_i)\,f =$$
$$(1-s_i)\,g + s_i f - t_i g + t_i f = g_i + t_i(f-g).$$

Thus,

$$\frac{\operatorname{Re}\langle g_i, g_{i+1}\rangle}{\|g_i\|\|g_{i+1}\|} = \frac{\|g_i\|^2 + t_i \operatorname{Re}\langle g_i, f-g\rangle}{\|g_i\|\sqrt{\|g_i\|^2 + 2t_i\operatorname{Re}\langle g_i, f-g\rangle + t_i^2\|f-g\|^2}} \geq L.$$

Thus, from Lemma 10 below we get

$$d_{\mathbb{S}}\left(\frac{g_i}{\|g_i\|}, \frac{g_{i+1}}{\|g_{i+1}\|}\right) = \arccos\frac{\operatorname{Re}\langle g_i, g_{i+1}\rangle}{\|g_i\|\,\|g_{i+1}\|} \leq \arccos L < \frac{c}{Pd^{3/2}\varphi_i}. \tag{4.8}$$

In particular, Theorem 7 applies to the segment $[g_i, g_{i+1}]$, proving our induction step and also proving (4.5) from (3.12). Additionally, if the i–th step is not the final step in our algorithm (equivalently, $g_{i+1} \neq f$ or $s_{i+1} < 1$) then we have

$$\frac{\operatorname{Re}\langle g_i, g_{i+1}\rangle}{\|g_i\|\|g_{i+1}\|} = \frac{\|g_i\|^2 + t_i \operatorname{Re}\langle g_i, f-g\rangle}{\|g_i\|\sqrt{\|g_i\|^2 + 2t_i\operatorname{Re}\langle g_i, f-g\rangle + t_i^2\|f-g\|^2}} \leq U_R,$$

which again using Lemma 10 yields:

$$d_{\mathbb{S}}\left(\frac{g_i}{\|g_i\|}, \frac{g_{i+1}}{\|g_{i+1}\|}\right) = \arccos\frac{\operatorname{Re}\langle g_i, g_{i+1}\rangle}{\|g_i\|\,\|g_{i+1}\|} \geq \arccos U_R > \frac{c}{RPd^{3/2}\varphi_i}. \tag{4.9}$$

Now we prove the bound on the number of steps. From (3.13) and (4.9) we have that, as long as $s_{i+1} < 1$,

$$\int_{s_i}^{s_{i+1}=s_i+t_i} \mu(f_t,\zeta_t)\sqrt{\frac{\|\dot{f}_t\|^2}{\|f_t\|^2} - \frac{\operatorname{Re}\left(\langle \dot{f}, f_t\rangle\right)^2}{\|f_t\|^4} + \|\dot{\zeta}_t\|^2_{T_{\zeta_t}\mathbb{P}(\mathbb{C}^{n+1})}}\, dt \geq$$
$$\varphi_i d_{\mathbb{S}}\left(\frac{g_i}{\|g_i\|}, \frac{g_{i+1}}{\|g_{i+1}\|}\right)\frac{(1-\sqrt{2}u_0/2)^{1+\sqrt{2}}\ln(1+c')}{c'} >$$
$$\frac{c(1-\sqrt{2}u_0/2)^{1+\sqrt{2}}\ln(1+c')}{c'RPd^{3/2}}.$$

Thus, as long as $s_{i+1} < 1$, we have

$$\mathcal{C}_0 = \int_0^1 \mu(f_t,\zeta_t)\sqrt{\frac{\|\dot{f}_t\|^2}{\|f_t\|^2} - \frac{\operatorname{Re}\left(\langle \dot{f}, f_t\rangle\right)^2}{\|f_t\|^4} + \|\dot{\zeta}_t\|^2_{T_{\zeta_t}\mathbb{P}(\mathbb{C}^{n+1})}}\, dt \geq$$

$$\int_0^{s_{i+1}} \mu(f_t,\zeta_t)\sqrt{\frac{\|\dot f_t\|^2}{\|f_t\|^2}-\frac{\mathrm{Re}\left(\langle \dot f, f_t\rangle\right)^2}{\|f_t\|^4}+\|\dot\zeta_t\|^2_{T_{\zeta_t}\mathbb{P}(\mathbb{C}^{n+1})}}\,dt =$$

$$\sum_{j=0}^{i}\int_{s_j}^{s_{j+1}} \mu(f_t,\zeta_t)\sqrt{\frac{\|\dot f_t\|^2}{\|f_t\|^2}-\frac{\mathrm{Re}\left(\langle \dot f, f_t\rangle\right)^2}{\|f_t\|^4}+\|\dot\zeta_t\|^2_{T_{\zeta_t}\mathbb{P}(\mathbb{C}^{n+1})}}\,dt >$$

$$\frac{(i+1)c(1-\sqrt{2}u_0/2)^{1+\sqrt{2}}\ln(1+c')}{c'RPd^{3/2}}.$$

In particular, if $s_{i+1}<1$ then

$$i+1<\frac{c'RPd^{3/2}}{c(1-\sqrt{2}u_0/2)^{1+\sqrt{2}}\ln(1+c')}\mathcal{C}_0.$$

The first non–negative integer i which violates this inequality is thus an upper bound for the number of iterations of the algorithm. This finishes the proof of the upper bound on the number of steps. For the lower bound, note that, even if $s_{i+1}=1$, from (3.13) and (4.8) we have

$$\int_{s_i}^{s_{i+1}} \mu(f_t,\zeta_t)\sqrt{\frac{\|\dot f_t\|^2}{\|f_t\|^2}-\frac{\mathrm{Re}\left(\langle \dot f, f_t\rangle\right)^2}{\|f_t\|^4}+\|\dot\zeta_t\|^2_{T_{\zeta_t}\mathbb{P}(\mathbb{C}^{n+1})}}\,dt \le$$

$$\varphi_i d_{\mathbb{S}}\left(\frac{g_i}{\|g_i\|},\frac{g_{i+1}}{\|g_{i+1}\|}\right)\frac{1+\sqrt{2}u_0/2}{(1-\sqrt{2}u_0/2)^{\sqrt{2}}}<\frac{c(1+\sqrt{2}u_0/2)}{Pd^{3/2}(1-\sqrt{2}u_0/2)^{\sqrt{2}}}\le\frac{c'}{Pd^{3/2}}.$$

Thus, if k is the number of iterations needed by the algorithm (i.e. $s_{k-1}<s_k=1$) then

$$\mathcal{C}_0=\int_0^1 \mu(f_t,\zeta_t)\sqrt{\frac{\|\dot f_t\|^2}{\|f_t\|^2}-\frac{\mathrm{Re}\left(\langle \dot f, f_t\rangle\right)^2}{\|f_t\|^4}+\|\dot\zeta_t\|^2_{T_{\zeta_t}\mathbb{P}(\mathbb{C}^{n+1})}}\,dt =$$

$$\sum_{j=0}^{k-1}\int_{s_j}^{s_{j+1}} \mu(f_t,\zeta_t)\sqrt{\frac{\|\dot f_t\|^2}{\|f_t\|^2}-\frac{\mathrm{Re}\left(\langle \dot f, f_t\rangle\right)^2}{\|f_t\|^4}+\|\dot\zeta_t\|^2_{T_{\zeta_t}\mathbb{P}(\mathbb{C}^{n+1})}}\,dt < k\frac{c'}{Pd^{3/2}}.$$

In particular, we conclude that the total number of iterations is

$$k\ge\frac{Pd^{3/2}\mathcal{C}_0}{c'},$$

which is the lower bound on k claimed in the theorem.

For (4.6), note that

$$U_R-L=\frac{c^2}{2P^2d^3\varphi_i^2}\left(1-\frac{1}{R^2}-\frac{c^2}{12P^2d^3\varphi_i^2}\right).$$

Using $R\ge\sqrt{2}$ and roughly bounding the term inside the parenthesis, we get

$$U_R-L\ge\frac{c^2}{5P^2d^3\varphi_i^2}\underset{(4.5)}{\ge}\frac{c^2\left((1-\sqrt{2}u_0/2)^{1+\sqrt{2}}\right)^2}{10P^2d^3\mu(g_i,\zeta_i)^4},$$

which implies (4.6).

□

Lemma 10. *Let L,U_R be defined as in the algorithm. Then,*

$$\arccos U_R>\frac{c}{RPd^{3/2}\varphi_i},\qquad \arccos L<\frac{c}{Pd^{3/2}\varphi_i}.$$

Proof. We prove the second inequality. Recall the elementary fact that for $0 < s < 1$ we have

$$\cos(s) < 1 - \frac{s^2}{2} + \frac{s^4}{24}.$$

Then, because arccos is a decreasing function in $[0, 1]$,

$$\arccos\left(1 - \frac{s^2}{2} + \frac{s^4}{24}\right) < s, \qquad s \in (0, 1). \tag{4.10}$$

In particular,

$$\arccos L = \arccos\left(1 - \frac{c^2}{2P^2d^3\varphi_i^2} + \frac{c^4}{24P^4d^6\varphi_i^4}\right) < \frac{c}{Pd^{3/2}\varphi_i},$$

as desired. The first inequality is proved in the same way, using that for $0 < s < 1$ we have

$$\cos(s) > 1 - \frac{s^2}{2}.$$

□

5. Computational considerations

Provided Theorem 8, the rigorously certified homotopy tracking could be accomplished by way of exact rational arithmetic employed in all of the computations described in TrackSegment_Scheme. In this section we discuss some of the aspects of this issue, to facilitate the reading of our main algorithm TrackSegment below.

5.1. **Operator norm vs. Frobenius norm.** In Step 4 of TrackSegment_Scheme we need to compute the operator norm of a matrix, which is a non–trivial task. Actually, one just needs to the square of such norm, to use it in steps 5 and 10. Instead of computing the square of the operator norm, one can just compute the square of the Frobenius norm $\|(a_{ij})\|_F^2 = \sum_{i,j} |a_{ij}|^2$, which involves only rational computations. Both norms are related by the inequalities

$$\|\cdot\|^2 \leq \|\cdot\|_F^2 \leq (n+1)\|\cdot\|^2.$$

On the other hand, $\chi_{i,2}^2$ which is the squared norm of a vector involves only rational computations and thus can be computed exactly. Then, instead of φ_i^2 in Step 5 we can use the product of $\chi_{i,2}^2$ and a version of $\chi_{i,1}^2$ using the squared Frobenius norm.

Let us put this in a general framework. Assume that we can compute some quantity $\tilde{\chi}_{i,1}^2$ satisfying $\chi_{i,1}^2 \leq \tilde{\chi}_{i,1}^2 \leq S^2\chi_{i,1}^2$ for some $S \geq 1$. Let $\tilde{L}, \tilde{U}_{\sqrt{2}}$ be computed with the same formulas as $L, U_{\sqrt{2}}$ but using $\tilde{\chi}_{i,1}$ instead of $\chi_{i,1}$. Then, it is easy to see that $\tilde{L} \geq L$ and

$$\tilde{U}_{\sqrt{2}} = 1 - \frac{c^2}{4P^2d^3\tilde{\varphi}_i^2} \leq 1 - \frac{c^2}{4S^2P^2d^3\varphi_i^2} = U_{\sqrt{2}S}.$$

Thus, if we find t_i such that

$$\tilde{L} \leq \frac{\|g_i\|^2 + t_i \operatorname{Re}\langle g_i, f-g\rangle}{\|g_i\|\sqrt{\|g_i\|^2 + 2t_i \operatorname{Re}\langle g_i, f-g\rangle + t_i^2\|f-g\|^2}} \leq \tilde{U}_{\sqrt{2}} \leq U_{\sqrt{2}S},$$

then in particular the hypotheses of Theorem 8 are fulfilled changing R to $\sqrt{2}S$ and the number of steps is at most

$$k \leq \lceil \frac{\sqrt{2}SPc'}{c(1-\sqrt{2}u_0/2)^{1+\sqrt{2}}\ln(1+c')} d^{3/2}\mathcal{C}_0 \rceil.$$

In particular, we have proved the following.

Lemma 11. *If in* TrackSegment_Scheme*, R is changed to $\sqrt{2}$ and $\chi_{i,1}^2$ is changed to $\tilde{\chi}_{i,1}^2$ (defined the same way as $\chi_{i,1}^2$ but using Frobenius norm instead of the operator norm), then any algorithm performing the computations in* TrackSegment_Scheme *has certified output in the sense of Theorem 8. The number of iterations is at most*

$$k \leq \lceil \frac{\sqrt{2(n+1)}Pc'}{c(1-\sqrt{2}u_0/2)^{1+\sqrt{2}}\ln(1+c')} d^{3/2}\mathcal{C}_0 \rceil \underset{\text{with } \delta=3/4}{\approx} \lceil 79\sqrt{n+1}d^{3/2}\mathcal{C}_0 \rceil.$$

Moreover, at every step we have

$$\tilde{\chi}_{i,1}\chi_{i,2} \leq \frac{\sqrt{2(n+1)}\mu(g_i,\zeta_i)^2}{(1-\sqrt{2}u_0/2)^{1+\sqrt{2}}}, \tag{5.1}$$

$$U_{\sqrt{2(n+1)}} - L \geq \frac{\hat{c}}{nd^3 \max\{\mu(f_t,\zeta_t)^4\}}, \tag{5.2}$$

where $\hat{c}$ is a universal constant.

5.2. **Computing the step size.** Step 5 of TrackSegment_Scheme requires finding a t satisfying (4.1), which *a priori* means computing approximately the smallest positive roots of two quadratic polynomials. An easier way to get this is using bisection method to locate a root of the equation

$$\alpha(t) = \frac{\theta_1 + t\theta_2}{\sqrt{\theta_1(\theta_1 + 2t\theta_2 + t^2\theta_3)}} - \frac{L+U}{2} = 0,$$

with stopping criterion given by

$$|\alpha(t)| \leq \frac{U-L}{2}.$$

If $t \in \mathbb{R}$ satisfies this stopping criterion, then (taking $\theta_1 = \|g_i\|^2, \theta_2 = \operatorname{Re}\langle g_i, f-g\rangle, \theta_3 = \|f-g\|^2$) it also satisfies (4.1). When applying bisection, we need to be able to determine the sign of $\alpha(t)$. It is not hard to accomplish this without computing square roots using the following subroutine.

Algorithm 1. $(s,r) =$ ComputeSign$(\theta_1,\theta_2,\theta_3,t,L,U)$
Require: $\theta_1,\theta_2,\theta_3,t,L,U \in \mathbb{Q}$, $\theta_1,\theta_3 > 0$, $\theta_2^2 < \theta_1\theta_3$, $0 < L < U < 1$.
Ensure: $s = 1$ *if* $\alpha(t) > 0$, $s = -1$ *otherwise, and*

$$r = \frac{(\theta_1 + t\theta_2)^2}{\theta_1(\theta_1 + 2t\theta_2 + t^2\theta_3)}$$

```
r ← $\frac{(\theta_1 + t\theta_2)^2}{\theta_1(\theta_1 + 2t\theta_2 + t^2\theta_3)}$
if $\theta_1 + t\theta_2 > 0$ and $r > (L+U)^2/4$ then
    $s \leftarrow 1$
else
    $s \leftarrow -1$.
end if
```

The bisection method mentioned above is then as follows. The requirement $1 - 10^{-3} < L$ in the description of the following algorithm corresponds to the extra hypotheses (2.1) in our main algorithm.

Algorithm 2. $t =$ LUquadratic$(\theta_1,\theta_2,\theta_3,L,U)$
Require: $\theta_1,\theta_2,\theta_3,L,U \in \mathbb{Q}$, $\theta_1,\theta_3 > 0$, $-L\sqrt{\theta_1\theta_3} \leq \theta_2 < \sqrt{\theta_1\theta_3}$, $1 - 10^{-3} < L < U < 1$.

Ensure: *$t = \frac{m}{2^l} \in \mathbb{Q} \cap (0,1]$, $m \in \mathbb{Z}$, $0 \leq l \in \mathbb{Z}$ such that*

$$L \leq \frac{\theta_1 + t\theta_2}{\sqrt{\theta_1(\theta_1 + 2t\theta_2 + t^2\theta_3)}} \leq U, \tag{5.3}$$

if such t exists (otherwise, output $t = 1$), and such that

$$0 < m \leq 2^l \leq \max\left(1, \frac{16\theta_3}{\theta_1(U-L)}\right). \tag{5.4}$$

```
1: $t_1 \leftarrow 1$
2: $L_2 \leftarrow L^2$
3: $(s_1, r_1) \leftarrow$ ComputeSign$(\theta_1, \theta_2, \theta_3, t_1, L, U)$
4: if $\theta_1 + \theta_2 > 0$ and $r_1 \geq L_2$ then
5:    $t \leftarrow t_1$
6: else
7:    $U_2 \leftarrow U^2$
8:    $t_0 \leftarrow 0$
9:    $t_2 \leftarrow \frac{t_0 + t_1}{2}$
10:   $(s_2, r_2) \leftarrow$ ComputeSign$(\theta_1, \theta_2, \theta_3, t_2, L, U)$
11:   $l \leftarrow 0$
12:   while $L_2 > r_2$ or $U_2 < r_2$ or $\theta_1 + t_2\theta_2 < 0$ do
13:      if $s_2 = 1$ then
14:         $t_0 \leftarrow t_2$
15:      else
16:         $t_1 \leftarrow t_2$
17:      end if
18:      $t_2 \leftarrow \frac{t_0 + t_1}{2}$
19:      $(s_2, r_2) \leftarrow$ ComputeSign$(\theta_1, \theta_2, \theta_3, t_2, L, U)$
20:      $l \leftarrow l + 1$
21:   end while
22:   $t \leftarrow t_2$
23: end if
```

Lemma 12. LUquadratic *produces $t = m/2^l$ satisfying (5.3) and (5.4), or $t = 1$ in case there exists no t that satisfies (5.3). Moreover, the number of iterations it performs is at most*

$$O\left(\log_2 \frac{\theta_3}{\theta_1(U-L)}\right).$$

Proof. Let

$$\beta(t) = \frac{\theta_1 + t\theta_2}{\sqrt{\theta_1(\theta_1 + 2t\theta_2 + t^2\theta_3)}}.$$

We first claim that

$$\beta'(t) = \alpha'(t) = -\frac{\left(\theta_1\,\theta_3 - {\theta_2}^2\right)\,t}{\sqrt{\theta_1}\left(\theta_3\,t^2 + 2\,\theta_2\,t + \theta_1\right)^{\frac{3}{2}}}$$

This is a routine computation and is left to the reader. In particular, $\theta_2^2 < \theta_1\theta_3$ implies that $\alpha(t)$ and $\beta(t)$ are decreasing functions for $t \geq 0$. If $\beta(1) \geq L$ (which is decided in Step 4 of Algorithm 2) then there are two possible scenarios:

(1) If $\beta(1) > U$ then a t satisfying (5.3) does not exist and the output of the algorithm is $t = 1$ as claimed.

(2) If $\beta(1) \leq U$ then the output of the algorithm $t = 1$ satisfies (5.3) and (5.4) as claimed. On the other hand, if $\beta(1) \leq L$, then

$$\alpha(0) = 1 - \frac{L+U}{2} > 0, \qquad \alpha(1) = \beta(1) - \frac{L+U}{2} \leq \frac{L-U}{2} < 0,$$

which implies that the bisection method used in the algorithm produces an approximation of the unique root $t_* \in (0,1)$ of $\alpha(t)$. In particular, note that $\alpha(t_*) = 0$ implies that $\theta_1 + t_*\theta_2 > 0$ and

$$\beta(t_*) = \frac{L+U}{2} \in (L,U),$$

which yields $\beta(t_*)^2 \in (L^2, U^2)$. By continuity of β, we conclude that the algorithm will at some point compute a t_2 such that $L^2 \leq r_2 \leq U^2$ and $\theta_1 + t_2\theta_2 > 0$. That is, the algorithm finishes at some point, and the output satisfies (5.3) as claimed. It is a simple induction exercise to prove that, whenever the condition of Line 12 is satisfied (that is to say, at every step of the algorithm, except possibly at the last one) we have

$$[p,q] \subseteq [t_0, t_1],$$

where

$$\begin{aligned}[p,q] &= \{t \in [0,1] : |\alpha(t)| \leq (U-L)/2\} \\ &= \{t \in [0,1] : \beta(t) \in [L,U]\} = \beta^{-1}([L,U]).\end{aligned}$$

At every step of the algorithm, the bisection method satisfies

$$|t_1 - t_0| = \frac{1}{2^l}.$$

Thus, if the l–th step is not the last step of the algorithm then we have

$$q - p \leq \frac{1}{2^l}.$$

From the Mean Value Theorem of calculus, we have

$$\frac{U-L}{q-p} = \frac{|\beta(q) - \beta(p)|}{|q-p|} = \beta'(\hat{t}\,),$$

for some $\hat{t} \in [p,q] = \beta^{-1}([L,U])$. Thus,

$$2^l \leq \frac{1}{q-p} = \frac{\beta'(\hat{t}\,)}{U-L} \leq \frac{\max\{|\beta'(t)| : \beta(t) \in [L,U]\}}{U-L}. \tag{5.5}$$

Note moreover that

$$\frac{|\beta'|}{\beta^3} = \frac{(\theta_1\theta_3 - \theta_2^2)\theta_1 t}{|\theta_1 + \theta_2 t|^3}. \tag{5.6}$$

Now we have to distinguish two cases:

(1) If $\theta_2 \geq 0$ then (5.6) and $t \leq 1$ yield

$$|\beta'(t)| \leq \frac{(\theta_1\theta_3 - \theta_2^2)\theta_1}{\theta_1^3} \leq \frac{\theta_3}{\theta_1}.$$

(2) If $\theta_2 < 0$ then by hypotheses we have $\theta_2 = -e$ with $0 < e \leq L\sqrt{\theta_1\theta_3}$. Thus,

$$\beta\left(\frac{\theta_1}{2e}\right) = \frac{\theta_1/2}{\sqrt{\theta_1}\sqrt{\frac{\theta_3\theta_1^2}{4e^2}}} = \frac{e}{\sqrt{\theta_1\theta_3}} \leq L,$$

and hence $\beta(t) \geq L$ implies that $t \leq \theta_1/(2e)$. Thus, (5.6) implies that

$$\frac{|\beta'|}{\beta(t)^3} \leq \frac{(\theta_1\theta_3 - e^2)\theta_1 t}{(\theta_1/2)^3} \underset{t\leq 1}{\leq} \frac{8(\theta_1\theta_3 - e^2)}{\theta_1^2} \leq \frac{8\theta_3}{\theta_1}, \qquad t \in [p,q],$$

which readily gives

$$|\beta'(t)| \leq \beta^3(t)\frac{8\theta_3}{\theta_1} \leq \frac{8\theta_3}{\theta_1}, \qquad t \in [p,q].$$

Thus, for every possible value of $\theta_2 \in (-L\sqrt{\theta_1\theta_3}, \sqrt{\theta_1\theta_3})$ we have that

$$\max\{|\beta'(t)| : \beta(t) \in [L,U]\} \leq \frac{8\theta_3}{\theta_1}.$$

This together with (5.5) proves that, if the l-th step is not the last step of the algorithm, we have

$$2^l \leq \frac{8\theta_3}{\theta_1(U-L)}.$$

For the last step, this quantity has to be multiplied by 2. The last claim of the lemma follows. □

Remark 13. *Our implementation of* LUquadratic *continues bisection if the denominator of its output t is larger than the denominator of s_i in step 23 of* TrackSegment *(Algorithm 4) until the denominators match. This is done in order to reduce the size of the denominator of s_{i+1}.*

5.3. Finding a close-by number with small integer coordinates.

In Step 12 of TrackSegment_Scheme we change z_{i+1} to a close-by vector $\tilde{z}_{i+1}$ with rational coordinates. Although z_{i+1} already has rational coordinates, we need to replace z_{i+1} with a nearby vector whose coordinates are integer numbers of bounded (small) absolute value. If this step is not performed, the number of bits required to write up z_{i+1} might increase at each loop, which is to be avoided. In this section we show how to deal with the general problem of, given $z \in \mathbb{Q}[\mathbf{i}]^{n+1}$ and $\varepsilon \in \mathbb{Q}$, $\varepsilon > 0$, finding $\tilde{z} \in \mathbb{Z}[\mathbf{i}]^{n+1}$ such that[4] $d_R(\tilde{z}, z) \leq \sqrt{\varepsilon}$, and such a way that the absolute value of the coordinates of $\tilde{z}$ is relatively small.

Let us consider the following algorithm.

Algorithm 3. $\tilde{z} =$ ShortZero(z, ε)

Require: $z \in \mathbb{Q}[\mathbf{i}]^{n+1}$; $\varepsilon \in \left(0, \frac{1}{5}\right) \cap \mathbb{Q}$.

Ensure: $\tilde{z} \in \mathbb{Z}[i]^{n+1}$ *such that*

$$d_R(\tilde{z}, z) \leq \sqrt{\varepsilon}, \tag{5.7}$$

and such that the integer numbers appearing in the expression of $\tilde{z}$ are bounded in absolute value by $3\sqrt{\frac{n+1}{\varepsilon}}$.

```
Let $\frac{a_i}{c_i} + \mathbf{i}\frac{b_i}{e_i}$, $a_i, b_i, c_i, e_i \in \mathbb{Z}$, $c_i, e_i > 0$, $0 \leq i \leq n$ be the coordinates of $z$.
$m \leftarrow (c_0 \cdots c_n) \cdot (e_0 \cdots e_n)$.
$x \leftarrow m \cdot z$.
$r \leftarrow \left(\frac{21}{20}\right)^2$
$k \leftarrow 0$
$\alpha \leftarrow 4$
while $\alpha \leq \frac{\varepsilon\|x\|^2}{2(n+1)r}$ do
    $\alpha \leftarrow 4\alpha$
    $k \leftarrow k + 1$
```

[4] Recall that $d_R(x,y) = \arccos \frac{\langle x,y \rangle}{\|x\|\|y\|}$ is the usual distance from x to y as projective points in $\mathbb{P}(\mathbb{C}^{n+1})$.

```
10: end while
11: $\tilde{z} \leftarrow [2^{-k}x]$.
```

Here, by $[y]$ ($y \in \mathbb{C}^{n+1}$) we mean the following: if $y = (a_0 + \mathbf{i}b_0, \dots, a_n + \mathbf{i}b_n)$ then

$$[y] = ([a_0] + \mathbf{i}[b_0], \dots, [a_n] + \mathbf{i}[b_n]),$$

where for $t \in \mathbb{R}$, $[t]$ is the integer number which is closest to t and is smaller than t in absolute value (that is, $[t] = \lfloor t \rfloor$ if $t \geq 0$ and $[t] = \lceil t \rceil$ if $t < 0$).

Lemma 14. *Let $0 \leq \theta_3 < \theta_1$. Then, the function*

$$w(\theta_2) = \frac{\theta_1 + \theta_2}{\sqrt{\theta_1(\theta_1 + 2\theta_2 + \theta_3)}}, \qquad \theta_2 \in [-\sqrt{\theta_1\theta_3}, \sqrt{\theta_1\theta_3}]$$

has a global minimum value equal to $\sqrt{1 - \theta_3/\theta_1}$.

Proof. Note that w is a differentiable function and

$$w'(\theta_2) = \frac{\theta_3 + \theta_2}{\sqrt{\theta_1}(\theta_1 + 2\theta_2 + \theta_3)^{3/2}}.$$

Hence, the minimum of w is attained at $\theta_2 = -\sqrt{\theta_1\theta_3}$, $\theta_2 = \sqrt{\theta_1\theta_3}$ or $\theta_2 = -\theta_3$. Now, $w(\sqrt{\theta_1\theta_3}) = w(-\sqrt{\theta_1\theta_3}) = 1$ and $w(-\theta_3) = \sqrt{1 - \theta_3/\theta_1} \leq 1$. The lemma follows. □

Lemma 15. *Algorithm 3 produces $\tilde{z} = (\tilde{\alpha}_0 + \mathbf{i}\tilde{\beta}_0, \dots, \tilde{\alpha}_n + \mathbf{i}\tilde{\beta}_n) \in \mathbb{Z}[\mathbf{i}]^{n+1}$ satisfying (5.7) and such that*

$$|\tilde{\alpha}_i|, |\tilde{\beta}_i| \leq 3\sqrt{\frac{n+1}{\varepsilon}} \qquad \forall\, 0 \leq i \leq n.$$

Proof. First note that, if the stopping condition of the loop is satisfied at the first step, then the output $\tilde{z} = x$ of the algorithm satisfies $d_R(\tilde{z}, z) = 0$ and

$$\|x\| \leq \sqrt{\frac{8(n+1)r}{\varepsilon}} \leq 3\sqrt{\frac{n+1}{\varepsilon}},$$

and hence the claim of the lemma follows. Otherwise, the numbers α, k computed by the algorithm satisfy

$$\alpha = 4^{k+1}, \qquad 4^k \leq \frac{\varepsilon\|x\|^2}{2(n+1)r} < 4^{k+1}. \tag{5.8}$$

Let $x = (\alpha_0 + \mathbf{i}\beta_0, \dots, \alpha_n + \mathbf{i}\beta_n)$ be the coordinates of x. Then, for $i = 0, \dots, n$, we have:

$$|(\tilde{\alpha}_i + \mathbf{i}\tilde{\beta}_i) - 2^{-k}(\alpha_i + \mathbf{i}\beta_i)|^2 = ([2^{-k}\alpha_i] - 2^{-k}\alpha_i)^2 + ([2^{-k}\beta_i] - 2^{-k}\beta_i)^2 < 2.$$

Hence, denoting $y = 2^{-k}x$ and $v = \tilde{z} - y$ we have

$$\|v\|^2 = \sum_{i=0}^{n} |(\tilde{\alpha}_i + \mathbf{i}\tilde{\beta}_i) - 2^{-k}(\alpha_i + \mathbf{i}\beta_i)|^2 \leq 2(n+1). \tag{5.9}$$

On the other hand,

$$\begin{aligned}\|y\|^2-\|v\|^2=\|y\|^2-\|\tilde z-y\|^2&=2\,\mathrm{Re}\,\langle \tilde z,y\rangle-\|\tilde z\|^2\\ &=2\left(\sum_{i=0}^n 2^{-k}\alpha_i[2^{-k}\alpha_i]+2^{-k}\beta_i[2^{-k}\beta_i]\right)-\|\tilde z\|^2\\ &\geq 2\left(\sum_{i=0}^n [2^{-k}\alpha_i]^2+[2^{-k}\beta_i]^2\right)-\|\tilde z\|^2\\ &=2\|\tilde z\|^2-\|\tilde z\|^2\geq 0.\end{aligned}$$

That is, $\|v\|^2\leq\|y\|^2$. Hence, the use of Lemma 14 in the following chain of inequalities is justified:

$$\frac{|\langle y+v,y\rangle|}{\|y\|\|y+v\|}\geq\frac{\mathrm{Re}\,\langle y+v,y\rangle}{\|y\|\|y+v\|}=\frac{\|y\|^2+\mathrm{Re}\,\langle v,y\rangle}{\|y\|\|y+v\|}=$$

$$\frac{\|y\|^2+\mathrm{Re}\,\langle v,y\rangle}{\|y\|\sqrt{\|y\|^2+2\,\mathrm{Re}\,\langle v,y\rangle+\|v\|^2}}\underset{\textit{Lemma 14}}{\geq}\sqrt{1-\frac{\|v\|^2}{\|y\|^2}}.$$

Thus,

$$d_R(\tilde z,z)=d_R(\tilde z,x)=d_R(\tilde z,y)=d_R(y+v,y)=\arccos\frac{|\langle y+v,y\rangle|}{\|y\|\|y+v\|}\leq$$

$$\arccos\sqrt{1-\frac{\|v\|^2}{\|y\|^2}}=\arcsin\frac{\|v\|}{\|y\|}=\arcsin\frac{2^k\|v\|}{\|x\|}\underset{(5.9)}{\leq}\arcsin\frac{2^k\sqrt{2(n+1)}}{\|x\|}.$$

Note from (5.8) that

$$\frac{2^k\sqrt{2(n+1)}}{\|x\|}\leq\sqrt{\frac{\varepsilon}{r}}\leq\sqrt{\varepsilon}\leq\frac{1}{2}. \tag{5.10}$$

The reader can check that the function $s\mapsto s^{-1}\arcsin s$, $s\in[0,1)$ is an increasing function. From this fact and (5.10) we get:

$$\frac{\arcsin\frac{2^k\sqrt{2(n+1)}}{\|x\|}}{\frac{2^k\sqrt{2(n+1)}}{\|x\|}}\leq\frac{\arcsin\frac{1}{2}}{\frac{1}{2}}\leq\frac{21}{20}=\sqrt{r},$$

which readily implies

$$d_R(\tilde z,z)\leq\arcsin\frac{2^k\sqrt{2(n+1)}}{\|x\|}\leq\sqrt{r}\,\frac{2^k\sqrt{2(n+1)}}{\|x\|}\underset{(5.8)}{\leq}\sqrt{\varepsilon},$$

as wanted.

For the bound on $|\tilde\alpha_i|$ note that

$$|\tilde\alpha_i|=\left|\left[2^{-k}\alpha_i\right]\right|\leq\left|2^{-k}\alpha_i\right|\underset{(5.8)}{\leq}\frac{\sqrt{8(n+1)r}}{\|x\|\sqrt{\varepsilon}}|\alpha_i|\leq 3\sqrt{\frac{n+1}{\varepsilon}},$$

where we have used $\|x\|\geq|\alpha_i|$. An identical chain of inequalities works for $\tilde\beta_i$.

□

TABLE 2. Notation of TRACKSEGMENT and TRACKSEGMENT_SCHEME.

TRACKSEGMENT	TRACKSEGMENT_SCHEME
n_1	$\|f\|^2$
n_2	$\|g\|^2$
n_3	$\mathrm{Re}\,\langle f, g\rangle$
$\dot{n}$	$\|f-g\|^2$
n_4	$\|g_i\|^2$
n_5	$\mathrm{Re}\,\langle f, g_i\rangle$
n_6	$\mathrm{Re}\,\langle f-g, g_i\rangle$
n_7	$\|z_i\|^2$
v_1	$f(z_i)$
v_2	$g_i(z_i)$
M	$\binom{Dg_i(z_i)}{z_i{}^*}^{-1}$
$\tilde{M}$	$\binom{Dg_{i+1}(z_i)}{z_i{}^*}$
$\mathfrak{a}$	$\tilde{\chi}_1^2$ ($= \chi_1^2$ computed with Frobenius norm)
$\mathfrak{b}$	χ_2^2
$\mathfrak{ab}$	$\tilde{\varphi}^2 = \tilde{\chi}_1^2\chi_2^2$ (plays the role of $\varphi^2 = \chi_1^2\chi_2^2$)
W	$\frac{c^2}{2P^2d^3\tilde{\varphi}^2}$

6. THE MAIN ALGORITHM

We now describe the pseudo-code of an actual algorithm that performs the instructions described in TRACKSEGMENT_SCHEME and is thus certified.

There are two choices in TRACKSEGMENT_SCHEME: R and δ. We choose $R = \sqrt{2}$ and $\delta = 3/4$ which make the computations simple. Besides, instead of using operator norm for the computation of $\chi_{i,1}^2$ we use Frobenius norm, which according to Section 5.1 multiplies by a factor of $\sqrt{n+1}$ the upper bound for the number of homotopy steps. The reader may find helpful Table 2 for comparing the names of the variables in TRACKSEGMENT_SCHEME and TRACKSEGMENT:

Algorithm 4. $z_* = \text{TRACKSEGMENT}(f, g, z_0)$

```
Require: $f, g \in \mathcal{H}_{(d)}$; $z_0 \in \mathbb{Q}[i]^{n+1}$ is an approximate zero of $g$ satisfying (3.9).
Ensure: $z_* \in \mathbb{Z}[i]^{n+1}$ is an approximate zero of $f$ associated to the end of the homotopy path
    starting at the zero of $g$ associated to $z_0$ and defined by the homotopy (3.11).
$i \leftarrow 0$; $s_i = 0$.
$n_1 \leftarrow \|f\|^2$.
$n_2 \leftarrow \|g\|^2$.
$n_3 \leftarrow \mathrm{Re}\,\langle f, g\rangle$.
$\dot{n} \leftarrow n_1^2 + n_2^2 - 2n_3$.
$\varepsilon_0 \leftarrow \dfrac{u_0^2}{(4d)^3(1+9u_0/8)^2}$
$W_0 \leftarrow \dfrac{17}{50000}\dfrac{1}{d^3}$
while $s_i < 1$ do
    $n_4 \leftarrow (1-s_i)^2 n_2 + s_i^2 n_1 + 2s_i(1-s_i)n_3$
    $n_5 \leftarrow (1-s_i)n_3 + s_i n_1$
    $n_6 \leftarrow s_i n_1 - (1-s_i)n_2 + (1-2s_i)n_3$
    $n_7 \leftarrow \|z_i\|^2$
    $M_1 \leftarrow Dg(z_i)$; $M_2 \leftarrow Df(z_i)$.
```

```
14:     $M = (m_{ij}) \leftarrow \begin{pmatrix} (1-s_i)M_1 + s_iM_2 \\ {z_i}^* \end{pmatrix}^{-1} \in \mathcal{M}_{n+1}(\mathbb{C}).$
15:     $\mathfrak{a} \leftarrow \left( n_4 \sum_{k=0}^{n} \sum_{l=0}^{n-1} d_{l+1} |m_{kl}|^2 n_7^{d_{l+1}-1} \right) + \left( \sum_{k=0}^{n} |m_{kn}|^2 n_7 \right)$
16:     $v_1 \leftarrow f(z_i) \in \mathbb{C}^n$
17:     $v_2 \leftarrow g_i(z_i) = (1-s_i)g(z_i) + s_iv_1 \in \mathbb{C}^n$
18:     $v_3 \leftarrow n_4v_1 - n_5v_2$
19:     $v_4 \leftarrow M\binom{v_3}{0} \in \mathbb{C}^{n+1}$.
20:     $\mathfrak{b} \leftarrow 1 + \frac{\|v_4\|^2}{n_7(n_1n_4 - n_5^2)}.$
21:     $W \leftarrow W_0/(\mathfrak{a}\mathfrak{b})$
22:     $L \leftarrow 1 - W + W^2/6$; $U \leftarrow 1 - W/2$.
23:     $t_i \leftarrow LUquadratic(n_4, n_6, \dot{n}, L, U)$.
24:     $s_{i+1} \leftarrow \min\{1, s_i + t_i\}$;
25:     $\varepsilon \leftarrow \varepsilon_0/\mathfrak{a}$
26:     $\tilde{M} \leftarrow \begin{pmatrix} (1-s_{i+1})M_1 + s_{i+1}M_2 \\ {z_i}^* \end{pmatrix} \in \mathcal{M}_{n+1}(\mathbb{C})$.
27:     $v_5 \leftarrow g_{i+1}(z_i) = (1-s_{i+1})g(z_i) + s_{i+1}v_1 \in \mathbb{C}^n$
28:     $z_{i+1} \leftarrow z_i - \tilde{M}^{-1}\binom{v_5}{0} \in \mathbb{C}^{n+1}$.
29:     $\tilde{z}_{i+1} \leftarrow$ ShortZero$(z_{i+1}, \varepsilon)$.
30:     $z_{i+1} \leftarrow \tilde{z}_{i+1}$.
31:     $i \leftarrow i + 1$.
32: end while
33: $z_* \leftarrow z_i$.
```

We should point out that in our practical implementation of the algorithm lines 13, 14, and 19 as well as lines 26, 27, and 28 correspond to the calls to the subroutine executing one step of Newton's method for a specialization of the system (3.11). We break this up into smaller steps above for the purpose of the complexity analysis performed in Subsection 7.4.

Remark 16. *From (5.2) and (5.4), for every $i \geq 0$ the number of iterations of* LUquadratic *at Step 23 is at most*

$$O\left(\log_2 \max\left(1, \frac{\|f-g\| n d^3 \max\{\mu(f_t, \zeta_t) : 0 \leq t \leq 1\}}{\min\{\|f_t\| : 0 \leq t \leq 1\}}\right)\right).$$

7. Complexity analysis

In this section we analyze the bit complexity of TrackSegment. Given a rational number $p/q \in \mathbb{Q}$, $gcd(p,q) = 1$, the *bit length* of p/q is defined as

$$\mathrm{bl}(p/q) = \log_2(\max|p|, |q|) + 1.$$

We also define $\mathrm{bl}(0) = 1$. Note that $\mathrm{bl}(p/q)$ is a (tight) upper bound for the number of binary digits required to write up p or q. writing p/q thus takes at most 2bl(p/q) bits.

Recall that an algorithm (i.e. a Turing machine) is said to have running time polynomial on quantities $c_1(x), c_2(x), \ldots, c_l(x)$ (where the $c_i(x)$ are quantities depending on the input x of the machine) if there exists a polynomial $p(X) \in \mathbb{R}[X_1, \ldots, X_l]$ such that the running time of the machine on input x is bounded above by $p(c_1(x),\ldots,c_l(x))$. A convenient notation is the following: given some function $f(x)$ depending on the input x, we say that

$$f(x) \leq (c_1(x),\ldots,c_l(x))^{O(1)}$$

if a polynomial p exists such that $f(x) \leq p(c_1(x),\ldots,c_l(x))$ for all possible input x. If a machine has running time which is polynomial in the (bit) size of its input, that is if the running time of the machine is $\mathit{input\_size}^{O(1)}$ then we say that the machine works in polynomial time. The reader does not need be very familiar with the concepts of computational complexity or Turing machine model to understand this section. However, we quote [11, Introduction] and its references for a brief yet illustrating introduction to the different concepts of algorithms, and [36] for a systematic introduction to Turing machines and their complexity.

When it comes to adding or multiplying rational numbers, there exist smart ways of designing the operations which can notoriously speed up the elementary algorithms, see for example [16]. However, we will not search for the optimal upper bounds on the complexity of our algorithm, because our intention is just to prove that it is polynomial in certain quantities as claimed in Theorem 2. We just recall from [16] that k arithmetic operations[5] (a.o. from now on) can be performed on rational inputs of bit length at most h, in time which is polynomial in k and h, that is in time $(kh)^{O(1)}$, and the result of this sequence of a.o. is a rational number $r \in \mathbb{Q}$ such that $\mathrm{bl}(r) \leq (kh)^{O(1)}$.

Given a vector $v \in \mathbb{Q}[\mathbf{i}]^k$, we define its bit length as

$$\mathrm{bl}(v) = \max\{\mathrm{bl}(a_i), \mathrm{bl}(b_i) : v = (a_1 + \mathbf{i}b_1, \ldots, a_k + \mathbf{i}b_k)\}.$$

7.1. **Bit complexity of** Computesign**.** Let h be an upper bound for the bit length of the input (a,b,c,t,L,U) of Computesign. The algorithm performs a fixed number of arithmetic operations on the rational numbers which are its input. Hence, the bit complexity of Computesign is $h^{O(1)}$.

7.2. **Bit complexity of** LUquadratic**.** Let h be an upper bound for the bit length of the input (a,b,c,L,U) of LUquadratic. Until Step 11, LUquadratic performs a fixed number of arithmetic operations on the rational numbers which are its input (including two applications of Computesign). The bit complexity of LUquadratic until Step 11 is thus $h^{O(1)}$. Each of the loops starting at Step 12 also performs a fixed number of arithmetic operations, but now the bit length of the number t_2 invoked in Computesign at line 19 grows with each loop. More precisely, after i iterations,

$$\mathrm{bl}(t_2) \leq O(i),$$

and thus the maximum bit length in all the numbers appearing at the algorithm in the i–th loop is $(h+i)^{O(1)}$. The total bit complexity is thus

$$O(h) + \sum_{i=1}^{\sharp\text{loops}} (h+i)^{O(1)} \leq (h + \sharp\text{loops})^{O(1)}.$$

From Remark 16, during an application of TrackSegment

$$\sharp\text{loops} \leq O\left(\log_2 \max\left(1, \frac{\|f-g\| n d^3 \max\{\mu(f_t,\zeta_t) : 0 \leq t \leq 1\}}{\min\{\|f_t\| : 0 \leq t \leq 1\}}\right)\right)$$

[5] By a.o. we mean an operation of the form $+,-,\times,/$, or a comparison $<,\leq$ or an assignment of a value to a variable, or computation of the integer part of a number.

Thus, the bit complexity of LUQUADRATIC on inputs of bit length at most h is, during an application of TRACKSEGMENT, at most

$$\left(h+\log_2 \max\left(1, \frac{\|f-g\|nd^3 \max\{\mu(f_t,\zeta_t) : 0\le t\le 1\}}{\min\{\|f_t\| : 0\le t\le 1\}}\right)\right)^{O(1)}.$$

7.3. **Bit complexity of** SHORTZERO. Let h be an upper bound for the bit length of the input (z,ε) of SHORTZERO. Steps 1 to 6 of SHORTZERO perform $O(n)$ a.o. on inputs of bit length bounded by h and thus these steps take time

$$(nh)^{O(1)},$$

which is also a bound for the bit length of x (in the notations of Algorithm SHORTZERO). The number of iterations the algorithm will perform is then at most

$$O(\log_2\|x\|) \le O(\log_2(nh)).$$

at each step the bit length of α increases by a factor of 4, and checking the stopping criterion can be done in $(nh/\log_2(\varepsilon))^{O(1)}$. Hence the total bit complexity of the while loop is

$$\sum_{i=1}^{\sharp\text{loops}} ((nh/\log_2(\varepsilon))^{O(1)} + O(i)) \le (nh\log_2(\varepsilon)+\sharp\text{loops})^{(O(1)} \le (nh\log_2(\varepsilon))^{O(1)}.$$

Step 11 can then be done in $(nh)^{O(1)}$. Thus, the total bit complexity of SHORTZERO is $(nh\log_2(\varepsilon))^{O(1)}$.

7.4. **Bit complexity of** TRACKSEGMENT. Let h be an upper bound for the bit length of the input (f,g,z_0) of TRACKSEGMENT. Let $S>0$ be the number of non-zero monomials in the dense representations of f and g. We assume that

$$\mu_{max} = \max\{\mu(f_t,\zeta_t) : 0\le t\le 1\} < \infty,$$

which indeed implies that $\mathcal{C}_0<\infty$ and by Theorem 8 we know that TRACKSEGMENT actually produces an approximate zero of f. We now analyze the operations performed in each step of TRACKSEGMENT.

(1) The operations before the while loop:
   - Steps $2,3,4$: two squared–norm computations and one inner product computation. That is $O(S)$ a.o. with rationals of bit length $\max\{h,l\}$ where l is an upper bound for the bit length of the multinomial coefficients $\binom{d_i}{\alpha_i}$ which appear in the definition of Bombieri–Weyl's product (see Section 2.1). Note that $l\le\log(d!)\le d^{O(1)}$. Thus, $\max\{h,l\}\le(h+d)^{O(1)}\le(hd)^{O(1)}$ and the bit complexity of these steps is at most $(Shd)^{O(1)}$. The numbers they produce have bit length $(Shd)^{O(1)}$ as well.
   - Steps $1,5,6,7$: a constant number of a.o. with rationals of bit length $(Shd)^{O(1)}$ is again $(Shd)^{O(1)}$ (and the numbers produced have the bit length bounded by the same quantity).

(2) Step 8 (number of loops): from Theorem 8 and Lemma 11, the number of loops is at most $\lceil 79\sqrt{n+1}d^{3/2}\mathcal{C}_0\rceil$, where $\mathcal{C}_0$ is the length of the path (f_t,ζ_t) in the condition metric. For counting the bit complexity of each loop, let h_i be h or the maximum bit length of the rational numbers s_i, z_i, t_i (whichever is greater), and let $h_{max}=\max\{h_i\}$ (we will prove latter that $h_{max}<\infty$). The bit complexity of the i–th loop is bounded as follows.
   - Steps $9,10,11$: a constant number of a.o. with rationals of bit length $(h_i)^{O(1)}$ is again $(h_i)^{O(1)}$.
   - Step 12: computation of the squared norm of a $\mathbb{C}^{n+1}$ vector with rational coordinates of bit length bounded by h_i: bit complexity $(nh_i)^{O(1)}$ and n_7 has bit length at most $(nh_i)^{O(1)}$, as well.

- Step 13: computation of the derivative matrices of f and g at z_i, which is $(nSdh_i)^{O(1)}$ using the elementary evaluation method (see [3] for a faster but more complicated one), and the bit length of the numbers is at most $(nSdh_i)^{O(1)}$.
- Step 14: addition of two $n \times (n+1)$ matrices with rational entries of bit length at most $(nSdh_i)^{O(1)}$ is $(nSdh_i)^{O(1)}$, then an inverse matrix computation is $(nSdh_i)^{O(1)}$ using modular techniques[6]. Indeed, computing of the inverse is equivalent to solving $n+1$ systems of equations with rational coefficients. Each of these systems can be first normalized to systems with integer coefficients of size $(nSdh_i)^{O(1)}$, which (according to, e.g., [18]) can be solved in time $(nSdh_i)^{O(1)}$. The total bit complexity of this step is thus $(nSdh_i)^{O(1)}$.
- Step 15: $O(n^2 \log_2(d))$ arithmetic operations (the $\log_2 d$ in this formula is needed to compute $n_7^{d_{l+1}-1}$) with numbers of bit length $(nSdh_i)^{O(1)}$ is again $(nSdh_i)^{O(1)}$, and the bit length of a is again bounded by $(nSdh_i)^{O(1)}$.
- Steps $16, 17$: computation of $f(z_i)$ and $g(z_i)$ is $(nSdh_i)^{O(1)}$ because that is a bound for the bit length of the rational numbers appearing in the monomial expansion of f, g and also for the bit length of the coordinates of z_i. There are also a constant number of a.o. which is again $(nSdh_i)^{O(1)}$.
- Step 18: a constant number of a.o. is again $(nSdh_i)^{O(1)}$.
- Step 19: a matrix–vector product, $O(n^2)$ a.o. with rationals of bit length bounded by $(nSdh_i)^{O(1)}$, is again $(nSdh_i)^{O(1)}$.
- Steps $20, 21, 22$: a constant number of a.o. is again $(nSdh_i)^{O(1)}$.
- Step 23: an application of LUQUADRATIC with input data whose bit length is bounded by $(nSdh_i)^{O(1)}$, according to Section 7.2 costs

$$\left((nSdh_i)^{O(1)} + \log_2 \max\left(1, \frac{\|f-g\| nd^3 \mu_{max}}{\min\{\|f_t\| : 0 \le t \le 1\}}\right)\right)^{O(1)}.$$

  By hypothesis, the output of LUQUADRATIC has the bit length bounded by $h_{i+1} \le h_{max}$.
- Step 24: a constant number of a.o. is again $(nSdh_{max})^{O(1)}$.
- Step 25: a division of two rational numbers of the respective bit lengths $(nSdh_{max})^{O(1)}$ and

$$\mathrm{bl}(\mathfrak{a}) = \mathrm{bl}(\tilde{\varphi}_i^2) \underset{(4.5)}{\le} O(\sqrt{n} \log_2 \mu_{max}) \tag{7.1}$$

  costs $(nSdh_{max} \log_2 \mu_{max})^{O(1)}$.
- Step 26: adding two $n \times (n+1)$ matrices, $O(n^2)$ a.o. with rationals of bit length $(nSdh_{max} \log_2 \mu_{max})^{O(1)}$ has bit complexity $(nSdh_{max}\mu_{max})^{O(1)}$.
- Step 27: as in Step 17, this takes time $(nSdh_i)^{O(1)}$.
- Step 28: solving a system of equations and adding two vectors with bit lengths bounded by $(nSdh_{max} \log_2 \mu_{max})^{O(1)}$ is again $(nSdh_{max}\mu_{max})^{O(1)}$ according to [18].
- Step 29: an application of SHORTZERO with input whose bit length is bounded by $(nSdh_{max} \log_2 \mu_{max})^{O(1)}$. From Section 7.3, this has bit complexity $(nSdh_{max} \log_2 \mu_{max})^{O(1)}$.
- Step $30, 31$: a constant number of a.o. with rationals of bit length $(nSdh_{max} \log_2 \mu_{max})^{O(1)}$ is $(nSdh_{max} \log_2 \mu_{max})^{O(1)}$.

(3) Step 33: One a.o. is again $(nSdh_{max} \log_2 \mu_{max})^{O(1)}$.

[6] Exact linear algebra is a large research field, see `http://linalg.org/people.html` for a list of people working on the subject, as well as software and research articles.

The bit complexity of TrackSegment is thus

$$\left(\left(nSdh_{max}\log_2\mu_{max}\right)^{O(1)}+\log_2\max\left(1,\frac{\|f-g\|nd^3\mu_{max}}{\min\{\|f_t\|:0\leq t\leq 1\}}\right)\right)^{O(1)}\mathcal{C}_0,$$

where h_{max} is the maximum of h and the bit lengths of s_i, t_i and z_i. Now, all the s_i and t_i are numbers of the form $m/2^l$ where, from Remark 16,

$$\mathrm{bl}(m)\leq\mathrm{bl}(2^l)=l+1\leq O\left(\log_2\max\left(1,\frac{\|f-g\|nd^3\mu_{max}}{\min\{\|f_t\|:0\leq t\leq 1\}}\right)\right).$$

Thus, this is also an upper bound for the bit lengths of s_i and t_i. As for that of z_i, note that from Lemma 15 we have that at each step $i\geq 1$,

$$\mathrm{bl}(z_i)\leq O\left(\log_2\frac{\sqrt{n}}{\varepsilon}\right)\leq O(\log_2(\sqrt{n}\mathfrak{a}))\underset{(7.1)}{\leq}O(\log_2(n\mu_{max})).$$

Hence, we have

$$h_{max}\leq h+\left(\left(\log_2\max\left(1,\frac{\|f-g\|nd^3\mu_{max}}{\min\{\|f_t\|:0\leq t\leq 1\}}\right)\right)^2+\log_2(n\mu_{max})\right).$$

The bit complexity of TrackSegment is thus linear in $\mathcal{C}_0$ and polynomial in the following quantities:

- $n, S, d, h,$
- $\log_2\mu_{max},$
- $\log_2(\|f-g\|/\min\{\|f_t\|:0\leq t\leq 1\})$.

8. Proof of Theorem 2

We first note that TrackSegment, performs the operations described by TrackSegment_Scheme, except for the use of Frobenius norm instead of operator norm in the computation of $\chi_{i,1}$. This follows directly from the description of the two algorithms and from lemmas 12 and 15.

Thus, from Theorem 8 and Lemma 11, TrackSegment has certified output. Moreover, its total bit complexity has been proved in section 7.4 to satisfy the claim of Theorem 2. For the bound on the size of the output, let $i=k$ be the final step of the algorithm. Then, the output z_{k+1} of TrackSegment is the result of applying ShortZero to some (z_k,ε) where $z_k\in\mathbb{Q}[\mathbf{i}]^{n+1}$ and

$$\varepsilon=\frac{\varepsilon_0}{\mathfrak{a}}\underset{(5.1)}{\geq}\frac{c_0}{\sqrt{n+1}\mu(g_k,\zeta_k)^2}\underset{(10.7)}{\geq}\frac{c_1}{\sqrt{n}\mu(f,\zeta_{k+1})^2},$$

c_0 and c_1 some constants. It follows from Lemma 15 that z_{k+1} has integer coordinates of bit length at most $O(\log_2(n\mu(f,\zeta_{k+1})))$, as claimed. The proof is now complete.

9. Experiments

Our implementation of Algorithm 4 has been carried out in the top-level (interpreted) language of *Macaulay2* [19]. The exact linear algebra routines and evaluation of polynomials are inherently slow and there are many engineering improvements that can be made to speed up the execution; yet the computation takes reasonable time on the examples of modest size.

While more examples of computation along with the source code of the implementation are available at

`http://people.math.gatech.edu/~aleykin3/RobustCHT/`

here we describe two experiments. One of them involves a small family of equations, where most of the computation of the length of a homotopy path $\mathcal{C}_0$ can be carried out by hand. The other comes from an application in enumerative geometry and showcases the class of problems that can benefit from the developed certified algorithms.

9.1. **Actual number of steps vs. condition length.** In Lemma 11 we claim that the number of steps (i.e. number of while loops) needed by Algorithm 4 is at most $\lceil 79\sqrt{n+1}d^{3/2}\mathcal{C}_0\rceil$. In this section we consider a simple family of examples parametrized by $m \geq 0$ where the value of $\mathcal{C}_0$ can be approximated by quadrature formulas and show how the bounds based on $\mathcal{C}_0$ compare to the actual performance of the algorithm. Note that from (2.3) the condition length of a path $(f_t, \zeta_t) \subseteq \mathcal{H}_{(d)} \times \mathbb{P}(\mathbb{C}^{n+1})$ (with ζ_t given by a smooth curve of affine representatives) is

$$\int_0^1 \chi_1(t)\sqrt{\frac{\|\dot{f}_t\|^2}{\|f_t\|^2} - \frac{\operatorname{Re}\left(\langle \dot{f}_t, f_t\rangle\right)^2}{\|f_t\|^4} + \frac{\|\dot{\zeta}_t\|^2}{\|\zeta_t\|^2} - \frac{|\langle\dot{\zeta}_t,\zeta_t\rangle|^2}{\|\zeta_t\|^4}},$$

were

$$\chi_1(t) = \left\| \begin{pmatrix} Df_t(\zeta_t) \\ \zeta_t{}^* \end{pmatrix}^{-1} \begin{pmatrix} \sqrt{d_1}\|f_t\|\|\zeta_t\|^{d_1-1} & & & \\ & \ddots & & \\ & & \sqrt{d_n}\|f_t\|\|\zeta_t\|^{d_n-1} & \\ & & & \|\zeta_t\| \end{pmatrix} \right\|.$$

In general, it is extremely hard to compute a priori $\mathcal{C}_0$ (even approximately). We consider here the simple case

$$f_t(x_0,x_1) = x_1^2 - (1+mt)x_0^2, \qquad \zeta_t = (1, \sqrt{1+mt})^T.$$

Let $s = 1+mt$. We can easily compute:

$$\|f_t\|^2 = 1+s^2; \qquad \dot{f}_t = -mx_0^2; \qquad \|\dot{f}_t\|^2 = m^2; \qquad \langle \dot{f}_t, f_t\rangle = ms;$$

$$\dot{\zeta}_t = \left(0, \frac{m}{2\sqrt{s}}\right)^T; \qquad \|\zeta_t\|^2 = 1+s; \qquad \|\dot{\zeta}_t\|^2 = \frac{m^2}{4s}; \qquad |\langle\dot{\zeta}_t,\zeta_t\rangle|^2 = \frac{m^2}{4};$$

Thus,

$$\sqrt{\frac{\|\dot{f}_t\|^2}{\|f_t\|^2} - \frac{\operatorname{Re}\left(\langle \dot{f}_t, f_t\rangle\right)^2}{\|f_t\|^4} + \frac{\|\dot{\zeta}_t\|^2}{\|\zeta_t\|^2} - \frac{|\langle\dot{\zeta}_t,\zeta_t\rangle|^2}{\|\zeta_t\|^4}} =$$

$$m\sqrt{\frac{1}{1+s^2} - \frac{s^2}{(1+s^2)^2} + \frac{1}{4s(1+s)} - \frac{1}{4(1+s)^2}} = m\sqrt{\frac{1}{(1+s^2)^2} + \frac{1}{4s(1+s)^2}}.$$

On the other hand,

$$\begin{pmatrix} Df_t(\zeta_t) \\ \zeta_t^* \end{pmatrix}^{-1} = \begin{pmatrix} -2s & 2\sqrt{s} \\ 1 & \sqrt{s} \end{pmatrix}^{-1} = \begin{pmatrix} \frac{-1}{2(1+s)} & \frac{1}{1+s} \\ \frac{1}{2\sqrt{s}(1+s)} & \frac{\sqrt{s}}{1+s} \end{pmatrix}$$

$$\chi_1(t) = \left\| \begin{pmatrix} \frac{-1}{2(1+s)} & \frac{1}{1+s} \\ \frac{1}{2\sqrt{s}(1+s)} & \frac{\sqrt{s}}{1+s} \end{pmatrix} \begin{pmatrix} \sqrt{2}\sqrt{1+s^2}\sqrt{1+s} & 0 \\ 0 & \sqrt{1+s} \end{pmatrix} \right\| =$$

$$\frac{1}{\sqrt{1+s}} \left\| \begin{pmatrix} \frac{-\sqrt{1+s^2}}{\sqrt{2}} & 1 \\ \frac{\sqrt{1+s^2}}{\sqrt{2s}} & \sqrt{s} \end{pmatrix} \right\| = \frac{\sqrt{1+s^2}}{\sqrt{2s}},$$

where to get the the last equality we compute the matrix norm by hand. With the change of variables $s = 1+mt$ we have then proved that

$$\mathcal{C}_0(f_t,\zeta_t) = \int_1^{1+m} \frac{\sqrt{1+s^2}}{\sqrt{2s}} \sqrt{\frac{1}{(1+s^2)^2} + \frac{1}{4s(1+s)^2}}\, ds,$$

It is not an easy task to find this integral exactly, but we can at least try to approximate with some quadrature formula. In Octave-produced Table 3 and Figure 3 we compare the values of upper and lower bounds

$$\begin{aligned} LBound &\leq \sharp(steps) \leq UBound, \text{ where} \\ LBound &= 28d^{3/2}\mathcal{C}_0(f_t,\zeta_t) \approx 79\mathcal{C}_0, \\ UBound &= 79\sqrt{n+1}d^{3/2}\mathcal{C}_0(f_t,\zeta_t) = 316\mathcal{C}_0(f_t,\zeta_t) \end{aligned}$$

for different choices of $m \geq 0$ and the number of steps performed by our algorithm to follow the homotopy f_t.

Table 3. Comparison of the bound of number of steps given by Lemma 11 and the actual number of steps in the example given by $f_t = x_1^2 - (1+mt)x_0^2$.

m	LB	steps	UB	UB/steps
10	31	184	357	1.95
20	38	217	435	2.01
30	42	237	480	2.03
40	45	250	512	2.05
50	47	260	537	2.07
60	49	269	558	2.08
70	50	276	575	2.08
80	52	282	590	2.09
90	53	288	603	2.1
100	54	292	615	2.11
1000	77	395	872	2.21
2000	84	426	949	2.23
3000	88	446	995	2.23
4000	91	457	1027	2.25
5000	93	468	1052	2.25
10000	100	499	1129	2.26
20000	106	530	1207	2.28
30000	110	547	1252	2.29

9.2. **An application to a problem in Schubert calculus.** The computations of [28] confirmed the conjecture saying that the Galois group of a simple Schubert problem is the full symmetric group for "small" Grassmannians. These results produced using heuristic homotopy continuation methods take us far beyond the limitations of the symbolic methods.

Table 4, a copy of [28, Table 1]), shows the number of solutions for the largest problem on $G(k,n)$ and the number of permutations found in the Galois group by the algorithm sufficient to generate the full symmetric group. At the present all computations can be done within one day with a heuristic homotopy tracker employed.

This problem falls naturally in the class where the certified algorithms of this paper can be applied. With the current implementation the algorithm of this paper can provide the *status of a theorem* to all of the computational results on up to $\mathrm{Gr}(2,6)$: the cases that can be certified within a day appear in bold in Table 4.

The corresponding runs of the algorithm for $\mathrm{Gr}(2,6)$ involve tracking homotopies for six polynomial equations following the paths in $\mathbb{P}^6$ and have input, output, and all intermediate approximate

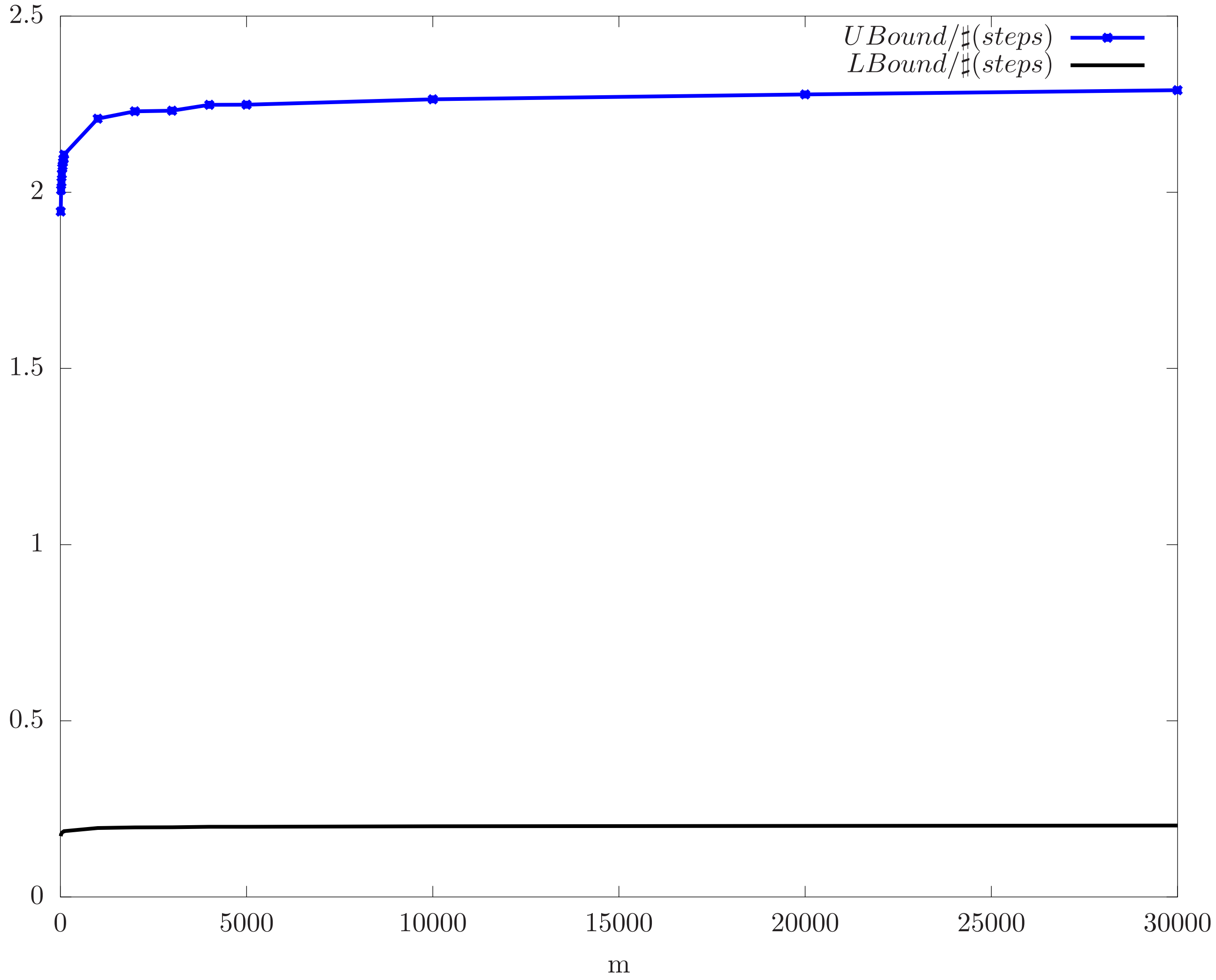

Figure 3. Comparison of the ratio between the actual number of steps and its lower and upper bound.

k, n	**2,4**	**2,5**	**2,6**	2,7	2,8	2,9	2,10
solutions	**2**	**5**	**14**	42	132	429	1430
permutations	**4**	**6**	**5**	6	7	4	7

k, n	3,5	3,6	3,7	3,8	3,9	4,6	4,7	4,8
solutions	5	42	462	6006	17589	14	462	8580
permutations	4	4	5	6	7	5	5	7

Table 4. Galois group computation for simple Schubert problems in $G(k, n)$.

zeroes defined over Gaussian integers $\mathbb{Z}[i]$. Due to the use of our Algorithm ShortZero to reduce the size of the integers in the intermediate steps, in this relatively large computation we do not encounter integers longer than *six* decimal digits amongst the coordinates of *all* approximate zeroes computed along all homotopy paths.

Let us remark that the largest certifiable case is already beyond the reach of purely symbolic algorithms (the problem with 14 solutions in $\mathrm{Gr}(2,6)$ is characterized as "not computationally feasible" in [10]). There are several ways to push the frontier of provable results further. One is a low-level optimized implementation of our algorithm. Another is using a fast heuristic homotopy tracker to find the "interesting" paths (e.g., the ones that do not lead to a redundant permutation in the Galois group computation), break them up into a union of smaller pieces, and then execute a certified homotopy tracker for every small piece. The last step is trivially parallelizable and can be sped up in practice by using distributed computing.

10. Proof of Theorem 7

We recall first two lemmas [4, Lemma 4 and Lemma 5]. The second of these two lemmas is recalled here in a less general version than the original.

Lemma 17. *Let $h_0, h \in \mathbb{S}$, $v \in \mathcal{H}_{(d)}$, $z_0, z \in \mathbb{P}(\mathbb{C}^{n+1})$. Assume that $\chi_1(h_0, z_0) < +\infty$. Assume moreover that*

$$d_R(z_0, z) \leq \frac{\hat{a}}{d^{3/2}\chi_1(h_0, z_0)},$$

$$d_{\mathbb{S}}(h_0, h) \leq \frac{3\hat{a}}{2d^{3/2}\chi_1(h_0, z_0)},$$

for some $\hat{a} < 1/\sqrt{2}$. Then,

$$\frac{\chi_1(h_0, z_0)}{1+\sqrt{2}\hat{a}} \leq \chi_1(h, z) \leq \frac{\chi_1(h_0, z_0)}{1-\sqrt{2}\hat{a}} \text{ and}$$

$$\varphi(h_0, v, z_0)\frac{(1-\sqrt{2}\,\hat{a})^{\sqrt{2}}}{1+\sqrt{2}\,\hat{a}} \leq \varphi(h, v, z) \leq \frac{\varphi(h_0, v, z_0)}{(1-\sqrt{2}\,\hat{a})^{1+\sqrt{2}}}.$$

Lemma 18. *Let $t \to h_s \in \mathbb{S}$, $0 \leq s \leq T$ be a piece of a great circle in $\mathbb{S}$, parametrized by arc–length. Let $\eta_0 \in \mathbb{P}(\mathbb{C}^{n+1})$ be a projective zero of h_0 such that $\mu(h_0, \eta_0) < +\infty$. Assume that*

$$T \leq \frac{1}{Pd^{3/2}\hat{\varphi}}, \text{ where } \hat{\varphi} = \varphi(h_0, \dot{h}_0, \eta_0).$$

Then, for $0 \leq s < T$, η_0 can be continued to a zero $\eta_s \in \mathbb{P}(\mathbb{C}^{n+1})$ of h_s in such a way that $s \to \eta_s$ is a C^{1+Lip} curve. Moreover, consider the curve $s \to (h_s, \dot{h}_s, \eta_s)$, $0 \leq s < T$. Then, the following inequalities hold for every $s \in [0, T]$:

$$\frac{\hat{\varphi}}{1+P\,d^{3/2}\hat{\varphi}s} \leq \varphi(h_s, \dot{h}_s, \eta_s) \leq \frac{\hat{\varphi}}{1-P\,d^{3/2}\hat{\varphi}s},$$

$$d_R(\eta_0, \eta_s) \leq \frac{1}{\sqrt{2}d^{3/2}\chi_1(h_0, \zeta_0)}\left(1-\left(1-Pd^{3/2}\hat{\varphi}s\right)^{\sqrt{2}/P}\right),$$

$$d_{\mathbb{S}}(h_0, h_s) \leq \frac{1}{d^{3/2}H}\log\frac{1}{1-d^{3/2}H\chi_2(h_0, \dot{h}_0, \eta_0)s}$$

Now we proceed to the proof of Theorem 7. Recall that we have defined $T = d_{\mathbb{S}}\left(\frac{g}{\|g\|}, \frac{f}{\|f\|}\right)$. Consider the path

$$s \to h_s = \frac{g}{\|g\|}\cos(s) + \frac{\frac{f}{\|f\|} - \mathrm{Re}(\langle\frac{f}{\|f\|}, \frac{g}{\|g\|}\rangle)\frac{g}{\|g\|}}{\sqrt{1-\mathrm{Re}(\langle\frac{f}{\|f\|}, \frac{g}{\|g\|}\rangle)^2}}\sin(s), \quad s \in [0, T], \tag{10.1}$$

That is, h_s is the arc–length parametrization of the short portion of the great circle joining $g/\|g\|$ and $f/\|f\|$. Note that $\dot{h}_0 = \dot{g}$ as was defined in Theorem 7.

Let

$$\hat{\chi}_1 = \chi_1(g, \zeta_0) = \mu(g, \zeta_0), \hat{\chi}_2 = \chi_2(g, \dot{g}, \zeta_0), \hat{\varphi} = \varphi(g, \dot{g}, \zeta_0).$$

From (3.9) and Lemma 17 we get

$$\hat{\varphi}\frac{(1-\sqrt{2}u_0/2)^{\sqrt{2}}}{1+\sqrt{2}u_0/2} \leq \varphi \leq \frac{\hat{\varphi}}{(1-\sqrt{2}u_0/2)^{1+\sqrt{2}}}. \tag{10.2}$$

From (3.7) and (3.8), we have:

$$\hat{\varphi} \leq \mu(g, \zeta_0)\|\dot{g}\|\sqrt{1+\mu(g,\zeta_0)^2} \leq \sqrt{2}\mu(g, \zeta_0)^2, \tag{10.3}$$

where for the last equality we have used that $\|\dot{h}_0\| = 1$ and $\mu(g, \zeta) \geq 1$. The second inequality of (10.2), together with (10.3), then implies (3.12).

It also follows that

$$T \underset{(3.10)}{\leq} \frac{c}{Pd^{3/2}\varphi} \underset{(10.2),(3.2)}{\leq} \frac{c'}{Pd^{3/2}\hat{\varphi}} < \frac{1}{d^{3/2}\hat{\varphi}}. \tag{10.4}$$

Thus, Lemma 18 applies and we conclude that $\zeta_0 = \eta_0$ can be continued to $\eta_s \in \mathbb{S}(\mathbb{C}^{n+1})$, a zero of h_s, for $0 \leq s \leq T$. Now, note that h_s is a reparametrization $s = s(t)$ of the projection of $f_t = (1-t)g + tf$ on $\mathbb{S}$. That is, $h_s = f_{t(s)}/\|f_{t(s)}\|$. Hence, ζ_0 can be continued to $\zeta_t = \eta_{s(t)}$, a zero of f_t as claimed in Theorem 7. Note that $\zeta_1 = \eta_{s(1)} = \eta_T$. Moreover, Lemma 18 and (3.7) also imply that, for $0 \leq s \leq T$,

$$\frac{\hat{\varphi}}{1+Pd^{3/2}\hat{\varphi}s} \leq \mu(h_s, \eta_s)\|(\dot{h}_s, \dot{\eta}_s)\|_{T_{(h_s,\eta_s)}\mathbb{S}\times\mathbb{P}(\mathbb{C}^{n+1})} \leq \frac{\hat{\varphi}}{1-Pd^{3/2}\hat{\varphi}s}, \tag{10.5}$$

and that

$$d_R(\zeta_0, \zeta_1) = d_R(\eta_0, \eta_T) \leq \frac{1}{\sqrt{2}d^{3/2}\hat{\chi}_1}\left(1-\left(1-Pd^{3/2}\hat{\varphi}\frac{c}{Pd^{3/2}\varphi}\right)^{\sqrt{2}/P}\right) \underset{(10.2),(3.2)}{\leq}$$

$$\frac{1}{\sqrt{2}d^{3/2}\hat{\chi}_1}\left(1-(1-c')^{\sqrt{2}/P}\right) \underset{(3.1)}{=} \frac{a}{\sqrt{2}d^{3/2}\hat{\chi}_1} \underset{(3.7)}{=} \frac{a}{\sqrt{2}d^{3/2}\mu(g,\zeta_0)}. \tag{10.6}$$

We have seen (10.4) that $T \leq c'(Pd^{3/2}\hat{\varphi})^{-1}$. Now, $\hat{\varphi} = \hat{\chi}_1\hat{\chi}_2$ and $\hat{\chi}_2 \geq \|\dot{h}_s\| = 1$, which implies

$$T \leq \frac{c'}{Pd^{3/2}\hat{\chi}_1} \underset{(3.7)}{=} \frac{c'}{Pd^{3/2}\mu(g,\zeta_0)} \underset{(3.3)}{\leq} \frac{3a}{2\sqrt{2}d^{3/2}\mu(g,\zeta_0)}.$$

Note that this last inequality, (10.6) and Lemma 17 imply

$$\frac{\mu(g,\zeta_0)}{1+a} \leq \mu(f, \zeta_1) \leq \frac{\mu(g,\zeta_0)}{1-a}. \tag{10.7}$$

Thus,

$$\mu(g, \zeta_0) \geq (1-a)\mu(f, \zeta_1), \tag{10.8}$$

and hence

$$T \leq \frac{c'}{Pd^{3/2}(1-a)\mu(f,\zeta_1)} \underset{(3.3)}{\leq} \frac{3\delta u_0}{2d^{3/2}\mu(f,\zeta_1)}. \tag{10.9}$$

Note that

$$\begin{aligned} d_R(z_0,\zeta_1)\mu(f,\zeta_1) &\leq (d_R(z_0,\zeta_0)+d_R(\zeta_0,\zeta_1))\mu(f,\zeta_1) \\ &\underset{(3.9),(10.6)}{\leq} \left(\frac{u_0}{2d^{3/2}\mu(g,\zeta_0)}+\frac{a}{\sqrt{2}d^{3/2}\mu(g,\zeta_0)}\right)\mu(f,\zeta_1) \\ &\underset{(10.7)}{\leq} \left(\frac{u_0}{2d^{3/2}}+\frac{a}{\sqrt{2}d^{3/2}}\right)\frac{1}{1-a}. \end{aligned}$$

Our choice of a is such that the right–hand term in this last equation is at most $\delta u_0/d^{3/2}$. Hence, we have

$$d_R(z_0,\zeta_1)\mu(f,\zeta_1) \leq \frac{\delta u_0}{d^{3/2}}. \tag{10.10}$$

From (10.10) and (10.9), Lemma 17 then yields

$$\frac{\mu(f,\zeta_1)}{1+\sqrt{2}\delta u_0} \leq \chi_1 \leq \frac{\mu(f,\zeta_1)}{1-\sqrt{2}\delta u_0}. \tag{10.11}$$

Moreover, from Lemma 6, (10.10) implies that z_0 is an approximate zero of f with associated zero ζ_1. In particular,

$$d_R(N_{\mathbb{P}}(f)(z_0),\zeta_1) \leq \frac{d_R(z_0,\zeta_1)}{2} \leq \frac{\delta u_0}{2d^{3/2}\mu(f,\zeta_1)}. \tag{10.12}$$

From this and (3.14) we have

$$\begin{gathered} d_R(\tilde{z},\zeta_1) \leq d_R(\tilde{z},N_{\mathbb{P}}(f)(z_0)) + d_R(N_{\mathbb{P}}(f)(z_0),\zeta_1) \leq \\ \frac{(1-\delta)u_0}{2d^{3/2}(1+3\delta u_0/2)\chi_1}+\frac{\delta u_0}{2d^{3/2}\mu(f,\zeta_1)} \leq \\ \frac{(1-\delta)u_0}{2d^{3/2}(1+\sqrt{2}\,\delta u_0)\chi_1}+\frac{\delta u_0}{2d^{3/2}\mu(f,\zeta_1)} \underset{(10.11)}{\leq} \\ \frac{u_0}{2d^{3/2}\mu(f,\zeta_1)}\left((1-\delta)+\delta\right) = \frac{u_0}{2d^{3/2}\mu(f,\zeta_1)}, \end{gathered}$$

proving (3.15).

As for (3.13), first note that from Lemma 19 below,

$$\mathcal{C}_0(f_t,\zeta_t) = \int_0^T \mu(h_s,\eta_s)\|(\dot{h}_s,\dot{\eta}_s)\|_{T_{(h_s,\eta_s)}\mathbb{S}\times\mathbb{P}(\mathbb{C}^{n+1})}\,ds. \tag{10.13}$$

Now, this last equality and (10.5) imply:

$$\begin{gathered} \mathcal{C}_0(f_t,\zeta_t) \geq \int_0^T \frac{\hat{\varphi}}{1+Pd^{3/2}\hat{\varphi}s}\,ds = \\ \frac{\ln(1+Pd^{3/2}\hat{\varphi}T)}{Pd^{3/2}} = \hat{\varphi}T\frac{\ln(1+Pd^{3/2}\hat{\varphi}T)}{Pd^{3/2}\hat{\varphi}T}. \end{gathered}$$

Because $\ln(1+t)/t$ is a decreasing function of $t>0$ and

$$Pd^{3/2}\hat{\varphi}T \underset{(10.2)}{\leq} Pd^{3/2}\frac{1+\sqrt{2}u_0/2}{(1-\sqrt{2}u_0/2)^{\sqrt{2}}}\varphi T \underset{(3.10)}{\leq} c\frac{1+\sqrt{2}u_0/2}{(1-\sqrt{2}u_0/2)^{\sqrt{2}}} \leq c',$$

we conclude that

$$\mathcal{C}_0(f_t,\zeta_t) \geq \hat{\varphi}T\frac{\ln(1+c')}{c'} \underset{(10.2)}{\geq} \tag{10.14}$$

$$\varphi T\frac{(1-\sqrt{2}u_0/2)^{1+\sqrt{2}}\ln(1+c')}{c'}.$$

On the other hand, using again (10.13), we have

$$\mathcal{C}_0(f_t,\zeta_t)\underset{(10.5)}{\le}\int_0^T\frac{\hat{\varphi}}{1-Pd^{3/2}\hat{\varphi}s}\,ds= \tag{10.15}$$

$$\hat{\varphi}T\frac{\log(1-Pd^{3/2}\hat{\varphi}T)}{-Pd^{3/2}\hat{\varphi}T}\le\hat{\varphi}T\underset{(10.2)}{\le}\varphi T\frac{1+\sqrt{2}u_0/2}{(1-\sqrt{2}u_0/2)^{\sqrt{2}}}.$$

Note that (10.14) and (10.15) prove (3.13). This finishes the proof of Theorem 7.

We have to prove a lemma that has been used in the proof of Theorem 7, and which is nothing but a change of variables:

Lemma 19. *In the notation of the proof of Theorem 7, we have:*

$$\mathcal{C}_0(f_t,\zeta_t)=\int_0^T\mu(h_s,\eta_s)\|(\dot{h}_s,\dot{\eta}_s)\|_{T_{(h_s,\eta_s)}\mathbb{S}\times\mathbb{P}(\mathbb{C}^{n+1})}\,ds.$$

Proof. One can just apply the change of variables formula to the change of variables $s=s(t)$ (so that $h_{s(t)}=f_t/\|f_t\|$ and $\eta_{s(t)}=\zeta_t$) and, after a long computation prove that the two integrals of the lemma are equal. However, we prefer the following geometric argument. The quantity $\mathcal{C}_0(f_t,\zeta_t)$ is by definition the length of the path $(f_t/\|f_t\|,\zeta_t)$ when $\mathbb{S}\times\mathbb{P}(\mathbb{C}^{n+1})$ is endowed with the condition metric, resulting from multiplying the usual product metric by the square of the condition number at each pair (f,z). Now, as a length, it is independent of the parametrization and thus $\mathcal{C}_0(f_t,\zeta_t)=\mathcal{C}_0(h_s,\eta_s)$. This is exactly the claim of the lemma. □

References

[1] D J. Bates, J D. Hauenstein, A J. Sommese, and C W. Wampler. Bertini: software for numerical algebraic geometry. Available at http://www.nd.edu/∼sommese/bertini.

[2] D J. Bates, C. Peterson, A J. Sommese, and C W. Wampler. Numerical computation of the genus of an irreducible curve within an algebraic set, Journal of Pure and Applied Algebra 215, no. 8 (2011), 1844–1851.

[3] W. Baur, and V. Strassen, The complexity of partial derivatives, Theoretical Computer Science 22, no. 3 (1983), 317–330.

[4] C. Beltrán, A continuation method to solve polynomial systems, and its complexity, Numerische Mathematik. 117, no. 1 (2011), 89–113.

[5] C. Beltrán and A. Leykin, Certified numerical homotopy tracking, Experimental Mathematics 21, no. 1 (2012), pp. 69–83.

[6] C. Beltrán and L.M. Pardo. On Smale's 17th problem: a probabilistic positive solution. Found. Comput. Math. 8, no. 1 (2008), 1–43.

[7] C. Beltrán and L.M. Pardo. Smale's 17th problem: Average polynomial time to compute affine and projective solutions. J. Amer. Math. Soc. 22 (2009), 363–385.

[8] C. Beltrán and L.M. Pardo. Fast linear homotopy to find approximate zeros of polynomial systems. Found. Comput. Math. 11, no. 1 (2011), 95–129.

[9] C. Beltrán and M. Shub. A note on the finite variance of the averaging function for polynomial system solving. Found. Comput. Math. 10, no. 1 (2010), 115–125.

[10] Sara Billey and Ravi Vakil, Intersections of schubert varieties and other permutation array schemes, in Algorithms in Algebraic Geometry (A. Dickenstein, F. O. Schreyer, and A J. Sommese, eds.), volume 146 of The IMA Vol. Math. Appl., Springer New York, 2008, pp. 21–54.

[11] L. Blum, F. Cucker, M. Shub, and S. Smale, Complexity and real computation, Springer-Verlag, New York, 1998.

[12] L. Blum, M. Shub, S. Smale. On a Theory of Computation and Complexity over the Real Numbers; NP Completeness, Recursive Functions and Universal Machines, Bull. Amer. Math. Soc. 21 (1989), 1–46.

[13] P. Bürguisser and F. Cucker, On a problem posed by Steve Smale, Annals of Mathematics 174 (2011), 1785–1836.

[14] D. Castro, K. Hägele, J. E. Morais, and L.M. Pardo. Kronecker's and Newton's approaches to solving: a first comparison, J. Complexity 17, no.1 (2001), 212–303.

[15] D. Castro, J.L. Montaña, L.M. Pardo and J. San Martín. The distribution of condition numbers of rational data of bounded bit length, Found. Comput. Math. 2–1 (2002), 1–52.

[16] T. H. Cormen, C.E. Leiserson, and R. L. Rivest, Introduction to algorithms, MIT Press, Cambridge, 1990.

[17] J-P. Dedieu, G. Malajovich and M. Shub, Adaptative step size selection for homotopy methods to solve polynomial equations, IMA Journal of Numerical Analysis, DOI: 10.1093/imanum/drs007

[18] J. D. Dixon. Exact solution of linear equations using p-adic expansions, Numer. Math. 40, no. 1 (1982), 137–141.

[19] D R. Grayson and M E. Stillman. Macaulay 2, a software system for research in algebraic geometry. Available at http://www.math.uiuc.edu/Macaulay2/.

[20] J. D. Hauenstein and F. Sottile. "alphacertified: certifying solutions to polynomial systems". arXiv:1011.1091v1, 2010.

[21] J. van der Hoeven. Reliable homotopy continuation, Technical Report, HAL 00589948, 2011.

[22] B. Huber, F. Sottile, and B. Sturmfels. Numerical Schubert calculus, J. Symbolic Comput. 26, no. 6 (1998), 767–788.

[23] B. Huber and B. Sturmfels. A polyhedral method for solving sparse polynomial systems, Math. Comp. 64, no. 212 (1995), 1541–1555.

[24] T. L. Lee, T. Y. Li, and C. H. Tsai. Hom4ps-2.0: A software package for solving polynomial systems by the polyhedral homotopy continuation method. Available at http://hom4ps.math.msu.edu/HOM4PS_soft.htm.

[25] R. B. Kearfott and Z. Xing. An interval step control for continuation methods, SIAM J. Numer. Anal. 31, no. 3 (1994), 892–914.

[26] M.H. Kim, Computational complexity of the Euler type algorithms for the roots of complex polynomials, PhD thesis, The City University of New York, 1985.

[27] A. Leykin. Numerical algebraic geometry for Macaulay2. J. of Software for Alg. and Geom. 3 (2011), 5–10.

[28] A. Leykin and F. Sottile. Galois groups of Schubert problems via homotopy computation, Math. Comp. 78, no. 267 (2009), 1749–1765.

[29] A. Leykin, J. Verschelde, and A. Zhao. Newton's method with deflation for isolated singularities of polynomial systems, Theoretical Computer Science 359, no. 1–3 (2006), 111–122.

[30] A. Leykin, J. Verschelde, and A. Zhao, Higher-order deflation for polynomial systems with isolated singular solutions, in Algorithms in algebraic geometry (A. Dickenstein, F. O. Schreyer, and A J. Sommese, eds.), volume 146 of The IMA Vol. Math. Appl., Springer, New York, 2008, pp. 79–97.

[31] G. Malajovich, PSS – Polynomial System Solver version 3.0.5. Available at http://www.labma.ufrj.br/ gregorio/software.php.

[32] G. Malajovich. On the complexity of path-following Newton algorithms for solving systems of polynomial equations with integer coefficients, PhD Thesis. Univ. California, Berkley, 1993.

[33] G. Malajovich. On generalized Newton algorithms : Quadratic convergence, path-following and error analysis, Theoretical Computer Science 133 (1994), 65–84.

[34] G. Malajovich. Condition number bounds for problems with integer coefficients, J. of Complexity 16, no. 3 (2000), 529–551.

[35] F. Mezzadri, How to generate random matrices from the classical compact groups, Notices of the American Mathematical Society 54, no. 5 (2007), 592–604.

[36] C.H. Papadimitriou, Computational complexity, Addison-Wesley Publishing Company, Reading, MA, 1994.

[37] J. Renegar. On the worst-case arithmetic complexity of approximating zeros of polynomials, Journal of Complexity, 3, no. 2 (1987), 90–113.

[38] M. Shub. Some remarks on Bezout's theorem and complexity theory, in From Topology to Computation: Proceedings of the Smalefest (M. W. Hirsch, J. E. Marsden, M. Shub eds.), Springer, New York, 1993, pp. 443–455.

[39] M. Shub. Complexity of Bézout's theorem. VI: Geodesics in the condition (number) metric. Found. Comput. Math. 9, no. 2 (2009), 171–178.

[40] M. Shub and S. Smale. Complexity of Bézout's theorem. II. Volumes and probabilities, in Computational algebraic geometry (Fr. Eyssette and A. Galligo, eds.), Progr. Math. 109. Birkhäuser, Boston, 1993, pp. 267-285.

[41] M. Shub and S. Smale. Complexity of Bézout's theorem. I. Geometric aspects, J. Amer. Math. Soc. 6, no. 2 (1993), 459–501.

[42] M. Shub and S. Smale. Complexity of Bezout's theorem. V. Polynomial time, Theoret. Comput. Sci. 133, no. 1 (1994), 141–164, Selected papers of the Workshop on Continuous Algorithms and Complexity (Barcelona, 1993).

[43] S. Smale. The Fundamental Theorem of Algebra and complexity theory, Bulletin of the Amer. Math. Soc. 4, no. 1 (1981), 1–36.

[44] S. Smale. Newton's method estimates from data at one point, in The merging of disciplines: new directions in pure, applied, and computational mathematics, Springer, New York, 1986, pp. 185–196.

[45] A J. Sommese and C W. Wampler, II, The numerical solution of systems of polynomials, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2005.

[46] J. Verschelde. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation, ACM Trans. Math. Softw., 25, no. 2 (1999), 251–276. Available at http://www.math.uic.edu/∼jan.

[47] K. Zyczkowski and M. Kus. Random unitary matrices. (English summary), J. Phys. A 133, no. 27 (1994), 4235–4245.