

SPHERICAL PROJECTIVE PATH TRACKING FOR HOMOTOPY CONTINUATION METHODS

TIANRAN CHEN* AND TIEN-YIEN LI†

Abstract. Solving systems of polynomial equations is an important problem in mathematics with a wide range of applications in many fields. The homotopy continuation method is a large class of reliable and efficient numerical methods for solving systems of polynomial equations. An essential component in the homotopy continuation method is the path tracking algorithm for tracking smooth paths of one real dimension. In this regard, “divergent paths” pose a tough challenge as the tracking of such paths is generally impossible. The existence of such paths is, in part, caused by \mathbb{C}^n , the space in which homotopy methods usually operate, being non-compact. A well known remedy is to operate inside the complex projective space $\mathbb{C}\mathbb{P}^n$ instead. Path tracking inside $\mathbb{C}\mathbb{P}^n$ is the focus of this article. Taking the Riemannian geometry of $\mathbb{C}\mathbb{P}^n$ into account, we derive the basic algorithm for projective path tracking using the sphere, S^{2n+1} , as the model of computation. Remarkable results from numerical experiments using this method are presented.

1. Introduction. Solving systems of polynomial equations is an important problem in mathematics. It has a wide range of applications in many fields. In this article we restrict our attention to systems of n polynomial equations in n variables with complex coefficients of the form

$$P(\mathbf{x}) = \begin{cases} p_1(x_1, \dots, x_n) = \sum_{k_1, \dots, k_n} c_{k_1, \dots, k_n}^{(1)} x_1^{k_1} \dots x_n^{k_n} = 0 \\ \vdots \\ p_n(x_1, \dots, x_n) = \sum_{k_1, \dots, k_n} c_{k_1, \dots, k_n}^{(n)} x_1^{k_1} \dots x_n^{k_n} = 0 \end{cases}$$

which will simply be called **polynomial systems**. By the Abel’s impossibility theorem and Galois theory, explicit formulae for solutions to such systems by radicals are unlikely to exist. As a consequence, numerical computation arises naturally in searching for solutions to such systems. Homotopy continuation methods represent a major class of numerical methods for this purpose.

Instead of attacking a polynomial system $P(\mathbf{x}) = \mathbf{0}$ head on, the homotopy continuation methods consider it as a member of a family of closely related polynomial systems parametrized by a single real parameter. One member $Q(\mathbf{x}) = \mathbf{0}$ of this family should be trivial to solve, and solutions of this trivial system should be connected via smooth solution paths to all isolated solutions of the target system $P(\mathbf{x}) = \mathbf{0}$. More precisely, we construct a homotopy $H : \mathbb{C}^n \times [0, 1] \rightarrow \mathbb{C}^n$ between the given polynomial system P and some chosen system Q : H is a continuous map from the product space $\mathbb{C}^n \times [0, 1]$ to \mathbb{C}^n such that $H(\mathbf{x}, 0) \equiv Q(\mathbf{x})$ and $H(\mathbf{x}, 1) \equiv P(\mathbf{x})$. It is common

* Department of Mathematics, Michigan State University, East Lansing, MI 48823. (E-mail: chentia1@msu.edu) Research supported in part by NSF under Grant DMS 11-15587

† Department of Mathematics, Michigan State University, East Lansing, MI 48823. (E-mail: li1@math.msu.edu) Research supported in part by NSF under Grant DMS 11-15587

to further require $H(\mathbf{x}, t)$ to have the **smoothness property**: The solution set of $H(\mathbf{x}, t) = \mathbf{0}$ for $t \in (0, 1)$ consists of a finite number of smooth paths in $\mathbb{C}^n \times (0, 1)$, each parametrized by t . Such solution paths can then be traced from the initial points, the solutions of $Q(\mathbf{x}) = \mathbf{0}$, at $t = 0$ to solutions of the target problem $P(\mathbf{x}) = \mathbf{0}$ using standard numerical techniques.

However, it could happen that a solution path does not converge to any point in \mathbb{C}^n , and instead its norm grows unboundedly as $t \rightarrow 1$. Such a path is called a **divergent path**. Divergent paths pose tough challenges to path tracking algorithms. In particular, they have infinite arc length, and thus tracking such paths directly is generally impossible. Since the genesis of general homotopy continuation methods for solving polynomial systems, much effort has been devoted to constructing the homotopy which minimizes the number of divergent paths. Despite the tremendous progress made in recent years, the handling of divergent paths remains an important problem.

Divergent paths exist, in part, because \mathbb{C}^n is not compact as a topological space. If we replace \mathbb{C}^n with a compact topological space W , a **compactification** of \mathbb{C}^n , in which \mathbb{C}^n is embedded as a dense subset, then one can show that all homotopy paths, now in $W \times [0, 1]$, must converge to points inside W at $t = 1$ and have finite arc length [22]. One of the most commonly used compactification in the context of algebraic geometry is the complex projective space $\mathbb{C}\mathbb{P}^n$. For a given homotopy $H = (h_1, \dots, h_n)$, its **homogenization** $\hat{H}(x_0, x_1, \dots, x_n)$ with respect to the variables (x_1, \dots, x_n) , as defined in (3) in Section 3, allows us to lift the problem into $\mathbb{C}\mathbb{P}^n$, since one can consider the equation $\hat{H} = \mathbf{0}$ to define solution paths in $\mathbb{C}\mathbb{P}^n$.

The focus of this article is to explore the path tracking algorithm for solutions in $\mathbb{C}\mathbb{P}^n$ from the point of view of the Riemannian geometry of $\mathbb{C}\mathbb{P}^n$. In the following sections, we start with an overview of the basic path tracking techniques in the affine space \mathbb{C}^n . Then we briefly outline the Riemannian geometry of $\mathbb{C}\mathbb{P}^n$ as the quotient manifold S^{2n+1}/S^1 to make this article self-contained. From this geometric structure, we derive the projective path tracking algorithm that works on the unit sphere S^{2n+1} . As a numerical algorithm, its numerical quality must be justified. We do so via the analysis of *path condition*, a concept we shall introduce in Section 9. A simple yet important technique of “dynamic row scaling” is then discussed as it is almost always necessary in implementing a robust numerical path tracking algorithm. Furthermore, the path condition is analyzed to show that the proposed algorithm does not artificially *pollute* the path condition. Very successful computational results on a case study of the 5-body central configuration problem with 20 equations, 20 unknowns, and a total degree of $1,787,822,080 = 4^{10} \cdot 3^{10}$ are presented in Section 10. In our algorithm, we chose the predictor-corrector scheme for the path tracking. There is a similar path tracking algorithm using “Projective Newton’s iterations” alone which has been intensively studied theoretically such as in [5], [6], [7], [8], [9], [10], [11], [12], [15], [25],

[26], [28], [29], [30], and [31], to list a few. In Section 11, numerical results are presented in comparing these two approaches. Remarkable efficiency of our algorithm in actual computing justifies our choice of predictor-corrector scheme in path tracking as in all of the software packages for solving polynomial system based on homotopy continuation methods, including BERTINI[4], PHOM[16], HOM4PS-2.0[20], and PHCPACK[33]. Purely technical and well known results from Riemannian geometry are listed in Appendices.

2. Affine path tracking. We shall briefly outline the basics of path tracking algorithms in \mathbb{C}^n . Fix any path $\gamma \subset \mathbb{C}^n \times (0, 1)$ defined by the homotopy $H(\mathbf{x}, t) = \mathbf{0}$, by the smoothness assumption, γ can be parametrized by the t -variable, and \mathbf{x} can be written as a smooth function $\mathbf{x}(t)$ of t which satisfies $H(\mathbf{x}(t), t) = \mathbf{0}$. Then it is easy to see that its tangent vector $\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt}$ must satisfy the system of ordinary differential equation

$$(1) \quad H_{\mathbf{x}}(\mathbf{x}(t), t) \cdot \dot{\mathbf{x}}(t) + H_t(\mathbf{x}(t), t) = \mathbf{0},$$

or simply $H_{\mathbf{x}} \cdot \dot{\mathbf{x}} = -H_t$, commonly known as the DAVIDENKO differential equation [1]. This forms the basis of the numerical path tracking algorithms with which one can trace a solution path from its starting point. While any numerical ordinary differential equation solver can, in principle, be applied to Equation (1) and thus be used for path tracking, the special class of *predictor-corrector* method is generally preferred. In such a scheme, an efficient but potentially inaccurate “predictor” is responsible for producing a rough estimate of the next point on the path using the information of known points on the path. Then a series of Newton-like “corrector” iterations is employed to bring the point approximately back to the path.

One of the most basic predictor-corrector configuration is the duet of Euler’s method and Newton’s iterations in which the prediction $\tilde{\mathbf{x}}(t_0 + \Delta t)$ for the value of \mathbf{x} at $t_1 = t_0 + \Delta t$ is given by

$$(2) \quad \tilde{\mathbf{x}}(t_0 + \Delta t) = \mathbf{x}(t_0) + \Delta t \cdot \dot{\mathbf{x}}(t_0) = \mathbf{x}(t_0) - \Delta t \cdot H_{\mathbf{x}}^{-1}(\mathbf{x}(t_0), t_0) \cdot H_t(\mathbf{x}(t_0), t_0),$$

where the existence of the inverse $H_{\mathbf{x}}^{-1}$ is warranted by the smoothness property of the homotopy construction. This prediction step is followed by a series of Newton’s iterations: at $t_1 = t_0 + \Delta t$, the equation $H(\mathbf{x}, t_1) = \mathbf{0}$ becomes a system of n equations in n unknowns. So Newton’s iterations can be used to refine the prediction $\tilde{\mathbf{x}}(t_1)$ with

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - [H_{\mathbf{x}}(\mathbf{x}^{(k)}, t_1)]^{-1} H(\mathbf{x}^{(k)}, t_1)$$

for $k = 0, 1, \dots$, where $\mathbf{x}^{(0)} = \tilde{\mathbf{x}}(t_1)$ is the starting point. It is the goal of this article to extend this predictor-corrector path tracking algorithm to the complex projective space.

3. Homotopy continuation in \mathbb{CP}^n . The existence of divergent paths calls for a compactification of \mathbb{C}^n , the space in which path tracking is performed. One of the most commonly used compactification of \mathbb{C}^n in the context of algebraic geometry is the complex projective space \mathbb{CP}^n :

$$\mathbb{CP}^n = (\mathbb{C}^{n+1} \setminus \{(0, \dots, 0)\}) / \sim$$

where $\mathbf{x} \sim \mathbf{y}$ for $\mathbf{x}, \mathbf{y} \in \mathbb{C}^{n+1}$ if there exists a $\lambda \in \mathbb{C} \setminus \{0\}$ such that $\mathbf{x} = \lambda \mathbf{y}$. In other words, points of \mathbb{CP}^n are one dimensional linear subspaces of \mathbb{C}^{n+1} with the “origin” deleted. It is common to use the notation $[x_0 : \dots : x_n]$ for the **homogeneous coordinate** of a point in \mathbb{CP}^n with $[x_0 : \dots : x_n]$ being equivalent to $[\lambda x_0 : \dots : \lambda x_n]$ for any $\lambda \in \mathbb{C} \setminus \{0\}$. With such coordinates \mathbb{CP}^n , as a set, can be covered by subsets $U_j = \{[x_0 : \dots : x_n] \mid x_j \neq 0\}$ for $j = 0, \dots, n$, called **standard charts**. Clearly, each standard chart U_j is equivalent to \mathbb{C}^n , as a set. These charts equip the set \mathbb{CP}^n the structure of a $2n$ -dimensional smooth manifold (as well as that of an n -dimensional complex manifold).

The zero sets of polynomials in \mathbb{CP}^n are not well defined in general since each point in \mathbb{CP}^n has infinitely many different coordinates. However, given any polynomial $f \in \mathbb{C}[x_1, \dots, x_n]$ of degree d , its **homogenization**

$$\hat{f}(x_0, \dots, x_n) = x_0^d \cdot f\left(\frac{x_1}{x_0}, \dots, \frac{x_n}{x_0}\right)$$

has the property that for $\mathbf{x} = (x_0, \dots, x_n)$, $\hat{f}(\lambda \cdot \mathbf{x}) = \lambda^d \cdot \hat{f}(\mathbf{x})$. Hence the zero set of \hat{f} is well defined in \mathbb{CP}^n , since for any $\lambda \neq 0$, $\hat{f}(\lambda \cdot \mathbf{x}) = 0$ if and only if $\hat{f}(\mathbf{x}) = 0$. Yet \hat{f} is still closely related to f in the sense that $\hat{f}(1, x_1, \dots, x_n) = f(x_1, \dots, x_n)$, i.e., whenever $x_0 \neq 0$, the zero sets of \hat{f} and f are equivalent. This common construction allows us to “lift” a problem into the complex projective space.

To apply this to the homotopy continuation method, given a homotopy $H(x_1, \dots, x_n, t) = (h_1, \dots, h_n)$ defined on $\mathbb{C}^n \times [0, 1]$ that is algebraic in the variables x_1, \dots, x_n , we shall consider their homogenizations with respect to the variables (x_1, \dots, x_n)

$$(3) \quad \hat{h}_j(x_0, \dots, x_n, t) = x_0^{d_j} \cdot h_j\left(\frac{x_1}{x_0}, \dots, \frac{x_n}{x_0}, t\right)$$

for each $j = 1, \dots, n$ where $d_j = \deg h_j$ and the new homotopy $\hat{H}(x_0, x_1, \dots, x_n, t) = (\hat{h}_1, \dots, \hat{h}_n)$, which is now defined on $\mathbb{C}^{n+1} \times [0, 1]$. Then for any fixed $t \in [0, 1]$ the common zero set of $\hat{H}(x_0, \dots, x_n, t)$ in \mathbb{CP}^n is well defined. To avoid confusion, the original solution paths defined by $H = \mathbf{0}$ in $\mathbb{C}^n \times [0, 1]$ will be called **affine paths**. Clearly, for any such affine path $\gamma \subset \mathbb{C}^n \times (0, 1)$, the corresponding path $\hat{\gamma} = \{([1, x_1, \dots, x_n], t) \mid (x_1, \dots, x_n, t) \in \gamma\} \subset \mathbb{CP}^n \times (0, 1)$ must satisfy the equation $\hat{H} = \mathbf{0}$. $\hat{\gamma}$ will be called a **projective path** corresponds to γ . One key advantage of working in \mathbb{CP}^n is that it is *compact* as a topological space, thus all projective paths

defined by $\hat{H} = \mathbf{0}$ must converge and be of finite length. The focus of this article is to derive a numerical algorithm for tracking the projective paths defined by $\hat{H} = \mathbf{0}$. To do so, the unit sphere S^{2n+1} is chosen to be our model of computation via the well known construction of $\mathbb{C}\mathbb{P}^n$ as the quotient manifold S^{2n+1}/S^1 .

4. The geometry of $\mathbb{C}\mathbb{P}^n$. Let $S^{2n+1} = \{\mathbf{x} \in \mathbb{C}^{n+1} : \|\mathbf{x}\|_2 = 1\}$ be the unit sphere of \mathbb{C}^{n+1} , which is a smooth manifold of $2n+1$ (real) dimension. It is a standard construction to view $\mathbb{C}\mathbb{P}^n$ as the quotient of S^{2n+1} under the action of the circle group: First of all, each point $(x_0, \dots, x_n) \in S^{2n+1}$ represents a point in $\mathbb{C}\mathbb{P}^n$ via the map $\pi : S^{2n+1} \rightarrow \mathbb{C}\mathbb{P}^n$ given by $(x_0, \dots, x_n) \mapsto [x_0 : \dots : x_n]$, which is clearly onto. However, the representative of a point in $\mathbb{C}\mathbb{P}^n$ is not unique, i.e., π is not 1-to-1, as $\pi(\mathbf{x}) = \pi(\lambda \mathbf{x})$ for any $\lambda \in \mathbb{C}^* = \mathbb{C} \setminus \{0\}$. But to leave S^{2n+1} invariant, we must have $|\lambda| = 1$, i.e., $\lambda = e^{i\theta}$. So for $\mathbf{x} \in S^{2n+1}$, the points of the form $e^{i\theta} \mathbf{x}$ with $\theta \in \mathbb{R}$ are exactly those that represent the same point as \mathbf{x} itself. Formally,

$$\pi^{-1}(\pi(\mathbf{x})) = \{e^{i\theta} \mathbf{x} \mid \theta \in \mathbb{R}\}.$$

Therefore, $\mathbb{C}\mathbb{P}^n$ can be identified with the set of equivalent classes $\{[\mathbf{x}] : \mathbf{x} \in S^{2n+1}\}$ where

$$[\mathbf{x}] := \{e^{i\theta} \mathbf{x} \mid \theta \in \mathbb{R}\}.$$

In fact, this identification is more than set theoretic. Let $S^1 = \{e^{i\theta} \mid \theta \in \mathbb{R}\}$ be the unit circle of \mathbb{C} which is a compact Lie group. Then the set $[\mathbf{x}]$ can be considered as the orbit of \mathbf{x} under the action of S^1 . So $\mathbb{C}\mathbb{P}^n$ can be identified with the quotient S^{2n+1}/S^1 of S^{2n+1} under the action of the compact Lie group S^1 . This quotient is a smooth manifold in its own right; on the other hand, it has a unique smooth structure for which π is a smooth submersion. One can show that, with this smooth structure, S^{2n+1}/S^1 is diffeomorphic to $\mathbb{C}\mathbb{P}^n$ whose smooth structure is given by the standard charts. Thus we shall use π to denote both the onto map $\pi : S^{2n+1} \rightarrow \mathbb{C}\mathbb{P}^n$ and the quotient map $\pi : S^{2n+1} \rightarrow S^{2n+1}/S^1$. This is a well known generalization of the Hopf fibration. In the rest of this article, unless otherwise specified, we shall simply equate $\mathbb{C}\mathbb{P}^n$ with the quotient manifold S^{2n+1}/S^1 .

To take one step further, since S^{2n+1} is a Riemannian manifold, with its Riemannian metric $g_{S^{2n+1}}$ inherited from the standard inner product of $\mathbb{C}^{n+1} \approx \mathbb{R}^{2n+2}$, the quotient map π also gives us a natural choice of the Riemannian metric on $\mathbb{C}\mathbb{P}^n \approx S^{2n+1}/S^1$. Since π is a submersion, at each point $\mathbf{x} \in S^{2n+1}$, its pushforward π_* has a constant rank of $2n$. Its kernel $\mathcal{V}_{\mathbf{x}} \subset T_{\mathbf{x}}S^{2n+1}$, of real-dimension 1, is known as the **vertical space**, which is simply the tangent space of the fiber over $\pi(\mathbf{x}) = [\mathbf{x}]$. Its orthogonal complement with respect to $g_{S^{2n+1}}$

$$\mathcal{H}_{\mathbf{x}} = \{\mathbf{h} \in T_{\mathbf{x}}S^{2n+1} \mid g_{S^{2n+1}}(\mathbf{h}, \mathbf{v}) = 0 \forall \mathbf{v} \in \mathcal{V}_{\mathbf{x}}\}$$

is known as the **horizontal space**, and it is a representation of the tangent space of the quotient S^{2n+1}/S^1 . There is a unique Riemannian metric g , called **Fubini-Study metric**, on $\mathbb{C}\mathbb{P}^n$, such that π is also a *Riemannian submersion*, i.e., at each point $\mathbf{x} \in S^{2n+1}$,

$$g_{S^{2n+1}}(\mathbf{h}_1, \mathbf{h}_2) = g(\pi_*(\mathbf{h}_1), \pi_*(\mathbf{h}_2))$$

for any $\mathbf{h}_1, \mathbf{h}_2 \in \mathcal{H}_{\mathbf{x}}$. In other words, π_* is an isometry on the horizontal space $\mathcal{H}_{\mathbf{x}}$.

5. Tracking projective paths via horizontal lifts. In the path tracking algorithm to be proposed, S^{2n+1} is chosen to be our model. That is, in actual computation, points of S^{2n+1} are used to represent points of $\mathbb{C}\mathbb{P}^n$. Just like $\mathbb{C}\mathbb{P}^n$, S^{2n+1} is compact as a topological space. In addition, all points in S^{2n+1} have coordinates with norm 1, a numerically favorable situation.

Tracking a smooth solution path $\hat{\gamma} \subset \mathbb{C}\mathbb{P}^n \times [0, 1]$ defined by the equation $\hat{H} = \mathbf{0}$ with parametrization $\hat{\mathbf{x}} : [0, 1] \rightarrow \mathbb{C}\mathbb{P}^n$, it is sufficient to track a representation $\mathbf{x} : [0, 1] \rightarrow S^{2n+1}$ in S^{2n+1} of the projective path $\hat{\mathbf{x}}$ in the sense that $\pi(\mathbf{x}(t)) = \hat{\mathbf{x}}(t)$ for all $t \in [0, 1]$. Unfortunately, there are infinitely many such representations in S^{2n+1} . In particular, if $\mathbf{x} : [0, 1] \rightarrow S^{2n+1}$ is such a representation, then so is

$$\mathbf{x}^{(1)}(t) = e^{i \cdot \theta(t)} \mathbf{x}(t)$$

for any smooth function $\theta : [0, 1] \rightarrow \mathbb{R}$. While, in principle, any choice of the representation would allow us to obtain the end point $\hat{\mathbf{x}}(1)$ that we are interested in, the Riemannian geometry of $\mathbb{C}\mathbb{P}^n$ suggests a natural choice: the *horizontal lift* of $\hat{\mathbf{x}}$. To explain this term, we shall first briefly outline the related concepts in Riemannian geometry.

Recall the orthogonal decomposition of the tangent space of S^{2n+1} at a fixed point \mathbf{x} into the *vertical* and the *horizontal* space with respect to the Riemannian metric $g_{S^{2n+1}}$:

$$T_{\mathbf{x}}S^{2n+1} = \mathcal{V}_{\mathbf{x}} \oplus \mathcal{H}_{\mathbf{x}}.$$

In this context, a tangent vector $\mathbf{v} \in T_{\mathbf{x}}S^{2n+1}$ is said to be **horizontal** if $\mathbf{v} \in \mathcal{H}_{\mathbf{x}}$. Similarly, a smooth vector field on S^{2n+1} is said to be **horizontal** if it is horizontal at any point in its domain. An important consequence of π being a Riemannian submersion is that for a given smooth vector field \hat{V} defined on some domain in $\mathbb{C}\mathbb{P}^n \approx S^{2n+1}/S^1$, there is a unique smooth *horizontal* vector field V on S^{2n+1} , called the **horizontal lift** of \hat{V} , that is π -related to \hat{V} , i.e. $\pi_*V_{\mathbf{x}} = \hat{V}_{\pi(\mathbf{x})}$ for any $\mathbf{x} \in S^{2n+1}$ where the vector fields are defined.

Among the infinite number of representations of the solution path $\hat{\gamma} \subset \mathbb{C}\mathbb{P}^n \times [0, 1] \approx S^{2n+1}/S^1 \times [0, 1]$ defined by $\hat{H} = \mathbf{0}$ with a fixed parametrization $\hat{\mathbf{x}} : [0, 1] \rightarrow S^{2n+1}/S^1$, we choose the *horizontal lift* of $\hat{\mathbf{x}}$, $\mathbf{x} : [0, 1] \rightarrow S^{2n+1}$, in the following

sense: When using t as the “time” parameter, we can consider the DAVIDENKO differential equation induced by $\hat{H} = \mathbf{0}$ to define a smooth *time-dependent* vector field $\hat{V} : (0, 1) \times S^{2n+1}/S^1 \rightarrow TS^{2n+1}/S^1$ on certain domain for which $\hat{\mathbf{x}}(t)$ is a (time-dependent) flow. Then this vector field has a unique π -related horizontal lift $V : (0, 1) \times S^{2n+1} \rightarrow TS^{2n+1}$ on S^{2n+1} . Fix any starting point $\mathbf{x}^{(0)} \in \pi^{-1}(\hat{\mathbf{x}}(0))$, there is a unique maximally defined solution to the initial value problem

$$\begin{aligned}\dot{\mathbf{x}}(t) &= V(t, \mathbf{x}(t)) \\ \mathbf{x}(0) &= \mathbf{x}^{(0)}\end{aligned}$$

which must be defined on the entire t -interval $(0, 1)$ by the smoothness condition of the homotopy construction. We shall call $\mathbf{x}(t)$ the **horizontal lift** of $\hat{\mathbf{x}}(t)$ with starting point $\mathbf{x}^{(0)}$. That is, we shall track the unique smooth curve $\gamma \subset S^{2n+1} \times [0, 1]$ parametrized by $\mathbf{x} : [0, 1] \rightarrow S^{2n+1}$ that satisfies

$$(4) \quad \begin{aligned}\mathbf{x}(0) &= \mathbf{x}^{(0)} \\ \dot{\mathbf{x}}(t) &\in \mathcal{H}_{\mathbf{x}(t)} \\ D_{\mathbf{x}}\hat{H}(\mathbf{x}(t), t)\dot{\mathbf{x}}(t) &= -D_t\hat{H}(\mathbf{x}(t), t)\end{aligned}$$

This is the projective analog of the DAVIDENKO differential equation (1).

Concerning Riemannian geometry, this choice is natural, because the submersion π acts as an isometry along such a curve. We shall further justify this choice from three different angles: First, among all smooth representation of $\hat{\gamma}$ in S^{2n+1} , the horizontal lift is the local minimizer of the length in the sense that over each infinitesimal t -interval, the horizontal lift has the minimum length among all representations of $\hat{\gamma}$, which is certainly a desirable property. Second, when the Fubini-Study metric is used, the horizontal lift has exactly the same length as $\hat{\gamma}$. Hence this choice of representation does not artificially stretch the curve in length. These two properties can be summarized by the following proposition.

PROPOSITION 1. *Fix the projective path $\hat{\gamma} \subset \mathbb{C}\mathbb{P}^n$ with its smooth parametrization $\hat{\mathbf{x}} : [0, 1] \rightarrow \mathbb{C}\mathbb{P}^n$ and a starting point $\mathbf{x}^{(0)} \in \pi^{-1}(\hat{\mathbf{x}}(0))$. Let Γ be the set of all smoothly parametrized curves $\mathbf{x} : [0, 1] \rightarrow S^{2n+1}$ such that $\pi \circ \mathbf{x} = \hat{\mathbf{x}}$ and $\mathbf{x}(0) = \mathbf{x}^{(0)}$. Also let $\mathbf{x}^{\mathcal{H}}$ be the unique horizontal lift of $\hat{\mathbf{x}}$ with starting point $\mathbf{x}^{(0)}$. Then*

- i. $\|\dot{\mathbf{x}}^{\mathcal{H}}(t)\| \leq \|\dot{\mathbf{x}}(t)\|$ for all $\mathbf{x}(t) \in \Gamma$.*
- ii. For any $\mathbf{x}(t) \in \Gamma$ and $t_0 \in (0, 1)$, there is a sufficiently small $\epsilon \in \mathbb{R}^+$ such that*

$$\int_{t_0}^{t_0+\epsilon} \|\dot{\mathbf{x}}^{\mathcal{H}}(t)\| dt \leq \int_{t_0}^{t_0+\epsilon} \|\dot{\mathbf{x}}(t)\| dt.$$
- iii. $\int_0^1 \|\dot{\mathbf{x}}^{\mathcal{H}}(t)\| dt = \int_0^1 \|\dot{\hat{\mathbf{x}}}(t)\| dt$.*

where $\|\bullet\|$ denotes norm operators induced by appropriate Riemannian metrics.

These are direct consequences of π being a Riemannian submersion, but its simple proof is included in Section C of Appendices for completeness.

Finally, as an arguably more important benefit for numerical algorithms, it will be shown later that the horizontal lift is the only choice that will never *pollute* the

numerical condition of the path tracking problem, a concept we shall introduce in Section 9.

6. Projective Davidenko equation in coordinates. To derive our numerical path tracking algorithm, we need the concrete numerical representation of the projective DAVIDENKO differential equation (4) in coordinates. Using \mathbb{C}^{n+1} as the ambient space, at each fixed $\mathbf{x} \in S^{2n+1} \subset \mathbb{C}^{n+1}$, the horizontal space $\mathcal{H}_{\mathbf{x}}$ has a simple numerical description:

PROPOSITION 2. *Via the isomorphism $T_{\mathbf{x}}\mathbb{C}^{n+1} \cong \mathbb{C}^{n+1}$, $\mathcal{H}_{\mathbf{x}}$ is given by the subspace*

$$\mathcal{H}_{\mathbf{x}} = \{\mathbf{v} \in \mathbb{C}^{n+1} \mid \langle \mathbf{x}, \mathbf{v} \rangle_{\mathbb{C}} = \mathbf{x}^H \mathbf{v} = 0\}$$

where \mathbf{x}^H is the conjugate transpose of vector \mathbf{x} .

The proof, while well known, is included in Section A of Appendices for completeness. Notice that this characterization of $\mathcal{H}_{\mathbf{x}}$ is invariant under the group action of S^1 , since if $\langle \mathbf{x}, \mathbf{v} \rangle_{\mathbb{C}} = 0$, then $\langle e^{i\theta} \mathbf{x}, \mathbf{v} \rangle_{\mathbb{C}} = 0$ for any $e^{i\theta} \in S^1$. With this formulation, the projective Davidenko differential equation (4) can be expressed in coordinate as

$$(5) \quad \begin{pmatrix} D_{\mathbf{x}} \hat{H}(\mathbf{x}, t) \\ \mathbf{x}^H \end{pmatrix} \cdot \dot{\mathbf{x}} = \begin{pmatrix} -D_t \hat{H}(\mathbf{x}, t) \\ 0 \end{pmatrix}.$$

It is clear that under the smoothness condition of the homotopy \hat{H} , the above system of ODE uniquely determines the tangent vector $\dot{\mathbf{x}}$ at each point along the curve $\mathbf{x}(t)$. So the projective path tracking can be reduced to the initial value problem given by (5) on the Riemannian manifold S^{2n+1} . This forms the foundation to build our projective path tracking algorithm. In the following sections we will outline the basic building blocks of the algorithm.

REMARK 1. *We should point out that the resulting formulation given above turns out to be very similar to a well known technique that makes use of affine charts of $\mathbb{C}\mathbb{P}^n$: For given $\mathbf{a} = (a_0, a_1, \dots, a_n) \in \mathbb{C}^{n+1}$, the linear equation $a_0 x_0 + a_1 x_1 + \dots + a_n x_n - 1 = 0$ defines a chart of $\mathbb{C}\mathbb{P}^n$ that is equivalent to a copy of \mathbb{C}^n . Restricting the homotopy construction to this particular chart yields the system:*

$$\begin{cases} \hat{H}(\mathbf{x}(t), t) & = 0 \\ \mathbf{a}^\top \mathbf{x}(t) - 1 & = 0 \end{cases}.$$

While it was originally proposed in [23] that one chooses \mathbf{a} to be a generic vector in \mathbb{C}^{n+1} , different techniques and heuristics have been developed to choose and change the affine charts [32]. In particular, as mentioned in [3], one would choose $\mathbf{a} = \bar{\mathbf{x}}/\|\mathbf{x}\|_2$ in certain situations; in those cases the resulting DAVIDENKO equation will be exactly the same as Equation (5). Therefore in one sense, Equation (5) looks like a result

of a specific choice of an affine chart: we always use $\langle \mathbf{x}(t)/\|\mathbf{x}(t)\|_2, \bullet \rangle_{\mathbb{C}} = 0$ as the affine chart. Nonetheless, to the best of our knowledge, it is the first time a geometric interpretation is provided in our work for the choice of this affine chart as the horizontal lift in the context of Riemannian submersion. Moreover, the sphere, S^{2n+1} , is built into our model of computation, that is, we track paths defined on S^{2n+1} rather than \mathbb{C}^{n+1} . Most importantly, we believe the basic idea behind this geometric interpolation may stimulate more general path tracking schemes in other quotient spaces such as the weighted projective space or toric varieties in general, which the authors intend to pursue in the near future.

7. Spherical projective Euler’s predictor. Given a point $\mathbf{x} = \mathbf{x}(t_0) \in S^{2n+1}$ on (or close to) a horizontal lift of a projective path and a step size Δt , the job of a predictor is to produce an approximation of the point on the path at $t = t_0 + \Delta t$. In light of Equation (5), with the ability to compute tangent vectors, almost any curve fitting or extrapolation scheme on the sphere S^{2n+1} can be used as predictors. For simplicity, we shall restrict our attention to the generalization of the Euler’s method.

A geometric interpretation of the Euler’s method in (2) is the movement of a point along the straight line defined by the tangent vector by a certain step length. The analogue in the context of Riemannian geometry is the exponential map $\text{Exp} : TS^{2n+1} \rightarrow S^{2n+1}$

$$\text{Exp}(\mathbf{x}, \mathbf{v}) := \gamma_{\mathbf{v}}(1)$$

where $\gamma_{\mathbf{v}} : \mathbb{R} \rightarrow S^{2n+1}$ is a Riemannian geodesic such that $\gamma_{\mathbf{v}}(0) = \mathbf{x}$ and $\dot{\gamma}_{\mathbf{v}}(0) = \mathbf{v}$. It moves a point $\mathbf{x} \in S^{2n+1}$ along a Riemannian geodesic passing through that point with the given initial tangent vector $\mathbf{v} \in T_{\mathbf{x}}S^{2n+1}$ for a step of unit length within the confine of S^{2n+1} . On S^{2n+1} , one can verify that the geodesic with initial tangent vector \mathbf{v} is simply given by

$$\gamma_{\mathbf{v}}(t) = \cos(\|\mathbf{v}\|_2 t) \mathbf{x} + \sin(\|\mathbf{v}\|_2 t) \mathbf{v} / \|\mathbf{v}\|_2.$$

Therefore, in this context, the exponential map is given by

$$\text{Exp}(\mathbf{x}, \mathbf{v}) = \cos(\|\mathbf{v}\|_2) \mathbf{x} + \sin(\|\mathbf{v}\|_2) \mathbf{v} / \|\mathbf{v}\|_2.$$

One can construct the generalized Euler’s method out of a scaled version of the exponential map: We define our **spherical projective Euler’s prediction** $\mathcal{E}_{\text{Exp}} : S^{2n+1} \times \mathbb{R} \rightarrow S^{2n+1}$ by

$$(6) \quad \mathcal{E}_{\text{Exp}}(\mathbf{x}, \Delta t) := \cos(\|\dot{\mathbf{x}}\|_2 \Delta t) \mathbf{x} + \sin(\|\dot{\mathbf{x}}\|_2 \Delta t) \dot{\mathbf{x}} / \|\dot{\mathbf{x}}\|_2$$

where Δt is the step size. It is easy to verify that $\mathcal{E}_{\text{Exp}}(\mathbf{x}, 0) = \mathbf{x}$, $\mathcal{E}_{\text{Exp}}(\mathbf{x}, \Delta t) \in S^{2n+1}$, and the Riemannian distance between \mathbf{x} and $\mathcal{E}_{\text{Exp}}(\mathbf{x}, \Delta t)$ is exactly $\|\dot{\mathbf{x}}\|_2 \cdot \Delta t$ for any $\Delta t \geq 0$, agreeing with our intuition.

8. Spherical projective Newton’s corrector. The prediction $(\mathbf{x}', t_0 + \Delta t)$ produced by the projective Euler’s predictor may not be exactly on or even very close to the projective path defined by $\hat{H} = \mathbf{0}$. If the next prediction step is to start from such an approximation, the error can quickly build up to an unacceptable level. To curb such error accumulation, a corrector is needed to produce a refinement \mathbf{x}'' of the approximate solution \mathbf{x}' of $\hat{H} = \mathbf{0}$ at $t_1 = t_0 + \Delta t$. When a corrector fails to bring the prediction back to the path quickly and reliably, it is usually the case that the step size Δt used in the prediction step is too large, and the prediction should be performed again with a smaller step size.

A natural choice of the corrector is an extension of the Newton’s iteration to the sphere. The extension can be done in the same way the spherical Euler’s method is constructed via the exponential map. Starting from the prediction $\mathbf{x}^{(1)} = \mathbf{x}'$ produced by the spherical Euler’s method, we shall construct an iterative method that produces a sequence of points $\mathbf{x}^{(2)}, \mathbf{x}^{(3)}, \dots$ that hopefully converge to some approximated solution \mathbf{x}'' of $\hat{H} = \mathbf{0}$ at $t = t_1$. For the k -th iteration, using the previous point $\mathbf{x}^{(k-1)}$, the Newton direction $\Delta \mathbf{x}^{(k)}$ is given via the linear system

$$\begin{pmatrix} \hat{H}_{\mathbf{x}}(\mathbf{x}^{(k-1)}, t_1) \\ (\mathbf{x}^{(k-1)})^H \end{pmatrix} \cdot \Delta \mathbf{x}^{(k)} = \begin{pmatrix} -\hat{H}(\mathbf{x}^{(k-1)}, t_1) \\ 0 \end{pmatrix}$$

which came from the “projective Newton’s method” developed in [27]. Considering the vector $\Delta \mathbf{x}^{(k)}$ as a horizontal tangent vector in $\mathcal{H}_{\mathbf{x}}$, the spherical Newton’s iteration is defined as

$$(7) \quad \mathcal{N}_{\text{Exp}}(\mathbf{x}^{(k-1)}) := \cos(\|\Delta \mathbf{x}^{(k)}\|_2) \mathbf{x}^{(k-1)} + \sin(\|\Delta \mathbf{x}^{(k)}\|_2) \Delta \mathbf{x}^{(k)} / \|\Delta \mathbf{x}^{(k)}\|_2.$$

Using this map, we can produce points

$$\mathbf{x}^{(k)} = \mathcal{N}_{\text{Exp}}(\mathbf{x}^{(k-1)})$$

for $k = 1, 2, \dots$ until certain convergence criteria are met. The exact convergence criteria are implementation dependent. The Riemannian distance $d_{S^{2n+1}}(\mathbf{x}^{(k)}, \mathbf{x}^{(k-1)})$ between consecutive points $\mathbf{x}^{(k)}$ and $\mathbf{x}^{(k-1)}$ or, in general, $d_{S^{2n+1}}(\mathbf{x}^{(k)}, \mathbf{x}^{(k-j)})$ for some $j \in \mathbb{N}$ serve as useful stopping criteria, since the shrinking of these distances is usually a good indication of convergence. Here we refer to [20] for a list of the stopping criteria as well as their detailed descriptions. Our preliminary implementation, equipped with these stopping criteria, has shown competitive performance as exhibited in Section 10.

REMARK 2. *Note that the spherical projective Newton’s method proposed here is very similar to the “Projective Newton’s method” introduced in [14] and [27]. One obvious difference is that the spherical projective Newton’s method uses the exponential map. A more important difference is the role it plays here. While the spherical projective Newton’s method is used as the corrector in the predictor-corrector scheme here, [14] and [27] proposed to use Projective Newton’s method alone to track the paths. Section 11 will present detailed comparison between these two approaches.*

9. Numerical condition of paths. While it is a common misconception that numerical algorithms find approximate solutions of a given problem, as pointed out by WILKINSON [35], they actually compute exact solutions to a nearby problem. The distance between the two problems is known as the *backward error*. Whether or not this solution is close to the solution of the original problem depends greatly on the sensitivity of a solution under certain perturbation of the problem. The *condition number* is a numerical measurement of this kind of sensitivity. Simply put, the condition number and errors are related by the inequality

$$\text{Forward error} \leq (\text{Condition number}) \cdot (\text{Backward error}).$$

It is important to note that while the backward error is controlled by the algorithm and computing devices used, the condition number is a property of the problem formulation itself. When the condition number is sufficiently large, one cannot provably control the forward error whenever *any* backward error is present.

We wish to assign such a condition number to the path tracking problem. We found it unlikely that a single number can characterize the condition of such a complex problem, so instead, we will introduce a weaker concept, the *condition of a homotopy path at a point*, in terms of a specific linear equation: Both equations (1) and (5) define the tangent vector of an affine or a projective path at a point in terms of a linear system, which we shall call the **tangent vector problem**. Let us define the **condition number of the path at a point** to be the condition number of the tangent vector problem at that point. If a path has a sufficiently large condition number at a point respect to a given threshold, we say the path is **ill-conditioned** at that point. In general a path is said to be **ill-conditioned** if it is ill-conditioned at any point on the path. The threshold, of course, depends on many factors such as the precision of the floating point arithmetic, the desired accuracy for solutions, and the nature of the problem or its application.

In practice, the effect of the path condition is twofold. First, it is a general experience that large path condition leads to very slow convergence for many numerical algorithms used for path tracking. A quantitative discussion of the computational complexity of the path tracking in relation to the condition number can be found in [14]. Second, when the path condition number is sufficiently large, one cannot obtain approximations of the path tangent vector with any reasonable accuracy which will definitely cast doubts on the validity of the final solutions obtained by the overall homotopy continuation method. In short, the tracking of ill-conditioned paths is slower and less trustworthy. In the following two subsections, we shall first discuss certain basic and well known preconditioning techniques that are very important in the context of our spherical projective path tracking algorithm. We then justify the benefit of our path tracking algorithm with regard to path conditions.

9.1. A dynamic preconditioning technique: row scaling. There is a large set of well known preconditioning techniques ranging from coefficients balancing [20] to randomization [32] that will greatly affect the path condition numbers, so it is crucial that those techniques are applied appropriately.

Here for simplicity, we shall follow a common practice to scale the matrix $\hat{H}_{\mathbf{x}}$ first so that 1 is in between the first singular value, σ_1 , and the n -th singular value, σ_n , of $\hat{H}_{\mathbf{x}}$, that is, $1 \in [\sigma_n, \sigma_1]$, a numerically favorable situation. Besides the scales of the singular values, the spread of the singular values also plays an important role in the path condition number. This spread can partially be controlled via a simple yet important technique of row scaling, illustrated by the following example.

Figure 1 shows the path condition along a single path when our projective path tracking algorithm on S^{2n+1} is used to solve the *barry* problem from the POSSO [2] test suite. With only 3 equations, 3 unknowns, and total degree 20, one expects no numerical difficulties, but the maximum path condition exceeds 10^9 in the figure. While our path tracking algorithm with double-precision floating point arithmetic had no trouble tracking this path, considering the simplicity of the system, this result is certainly surprising and, to a certain extent, unsettling. From Figure 2, it appears that the large difference in the scales of the rows in the Jacobian matrix is the cause of such a big path condition.

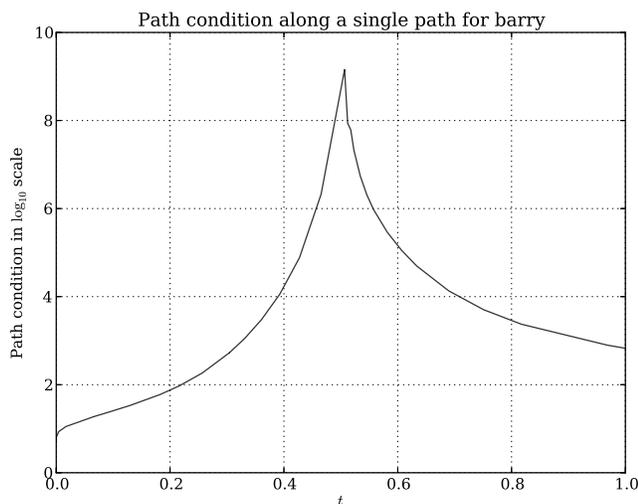


FIG. 1. Path condition, in \log_{10} scale, along a single path tracked for solving the *barry* problem using the spherical projective path tracking algorithm.

Our experiments have shown that this problem of large difference in the scale of rows in Jacobian matrix revealed by the above observation is widespread when homogenization technique is in use. To see why, consider a homogeneous polynomial

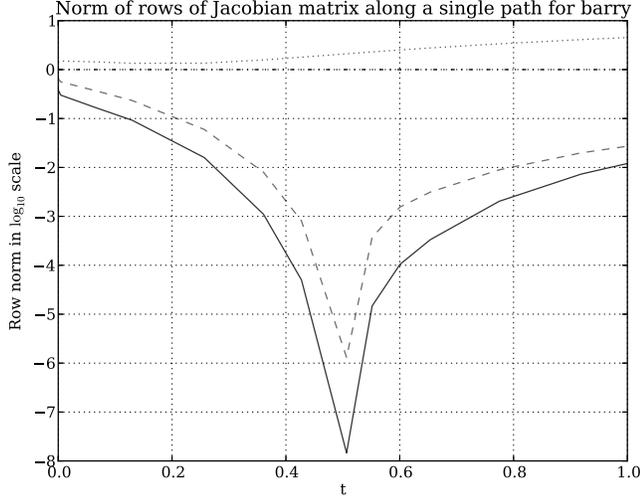


FIG. 2. The 2-norm of the four rows, in \log_{10} scale, of the Jacobian matrix along the same path shown in Figure 1 defined by $\hat{H} = \mathbf{0}$, tracked using spherical projective path tracking algorithm for solving the *barry* problem.

$f \in \mathbb{C}[x_0, \dots, x_n]$ of degree d , then each $\frac{\partial f}{\partial x_j}$ for $j = 0, \dots, n$ is homogeneous of degree $d - 1$ unless f is constant with respect to x_j . In either case, we have $\frac{\partial f}{\partial x_j}(\lambda \mathbf{x}) = \lambda^{d-1} \frac{\partial f}{\partial x_j}(\mathbf{x})$. Thus for a fixed t , if we write the Jacobian matrix of $F(\mathbf{x}) = \hat{H}(\mathbf{x}, t)$ with respect to \mathbf{x} at a point \mathbf{x} as the row matrix

$$J(\mathbf{x}) = \begin{pmatrix} J_1 \\ \vdots \\ J_n \end{pmatrix} \quad \text{then} \quad J(\lambda \mathbf{x}) = \begin{pmatrix} \lambda^{d_1-1} J_1 \\ \vdots \\ \lambda^{d_n-1} J_n \end{pmatrix},$$

where $d_i = \deg f_i$. Namely, when the given point \mathbf{x} is scaled by a fixed factor, the rows in J are scaled by different factors determined by the degrees of the corresponding polynomials. Therefore if the original system contains polynomials of very different degrees, the difference in the scales of the rows in J can be very sensitive to the scaling $\mathbf{x} \mapsto \lambda \mathbf{x}$.

To apply this observation to the context of spherical projective path tracking, consider a point on a given projective path $[\mathbf{x}] = \hat{\mathbf{x}}(t_0)$ for some given t_0 , let us pick any point $\mathbf{x}^{\min} \in \mathbb{C}^{n+1}$ that represents $[\mathbf{x}]$ for which the spread of the n singular values of $\hat{H}_{\mathbf{x}}(\mathbf{x}, t_0)$ is minimized among all such points in \mathbb{C}^{n+1} . Since, for simplicity, tracking paths on S^{2n+1} , we are potentially dealing with a sub-optimal scaled version $\frac{1}{\|\mathbf{x}^{\min}\|_2} \cdot \mathbf{x}^{\min}$ of \mathbf{x}^{\min} . This scaling could adversely affect the path condition number.

The problem stated above can be solved quite simply via row scaling, a basic technique in numerical linear algebra: Clearly, the system of linear equations $J\mathbf{v} = \mathbf{b}$

is equivalent to $AJ\mathbf{v} = A\mathbf{b}$ for any nonsingular square matrix A . So to bring all the rows to more or less the same scale, we may take A to be the diagonal matrix with entries $1/\|J_1\|, \dots, 1/\|J_n\|$. In principle any norm can be used here. Our actual implementation uses the ∞ -norm, for the ease of computation. Figure 3 shows the path condition of the same path tracked for solving the `barry` problem with and without the dynamic row-scaling technique. The difference is day and night. In solving a large number of polynomial systems with projective path tracking, this simple technique is helpful, and, in certain cases, essential in improving the path conditions.

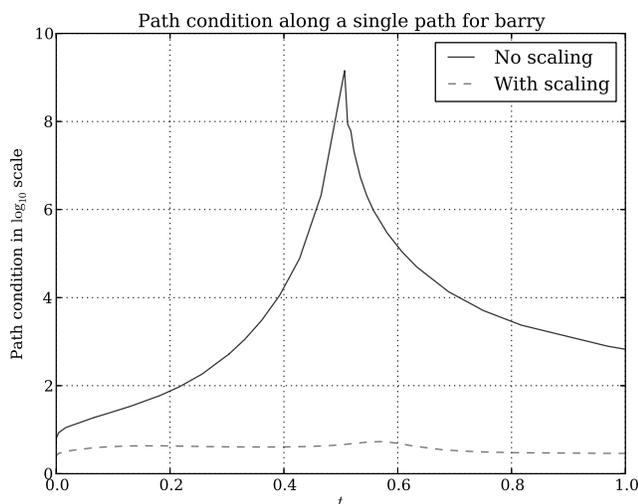


FIG. 3. The comparison between path condition of a single path tracked in solving the `barry` problem using projective path tracking with (dashed) and without (solid) the dynamic row-scaling

However, it is important to note that while row-scaling is useful in improving the path condition, this transformation may conceal the fact that we are near a true singularity of a path, and thus it must be used with caution. In particular, this technique should not be used near the endpoint at $t = 1$ where singularity may appear. Near the endpoint, the so called “endgame” techniques must be used.

9.2. Path condition for spherical projective path tracking. In this subsection, we shall analyze the path condition number in the context of spherical projective path tracking under the assumption that appropriate preconditioning techniques have already been applied. In particular, we assume $1 \in [\sigma_n, \sigma_1]$, where σ_1 and σ_n are the first and n -th singular values of $\hat{H}_{\mathbf{x}}$ respectively.

For a fixed $t \in [0, 1]$, define $F(\mathbf{x}) = \hat{H}(\mathbf{x}, t) = (f_1, \dots, f_n)$, then Equation (5) can

be written as

$$\begin{pmatrix} DF(\mathbf{x}) \\ \mathbf{x}^H \end{pmatrix} \cdot \mathbf{v} = \begin{pmatrix} \mathbf{b} \\ 0 \end{pmatrix}$$

for some $\mathbf{b} \in \mathbb{C}^n$. Here $DF(\mathbf{x})$ is the Jacobian matrix of F . Still let $\{\sigma_1, \dots, \sigma_n\}$ with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0$ be the first n singular values of $DF(\mathbf{x})$. We shall compute the path condition at this point given by the condition number of the matrix $\begin{pmatrix} DF(\mathbf{x}) \\ \mathbf{x}^H \end{pmatrix}$.

Since $F(\mathbf{x}) = (f_1, \dots, f_n)$ is a system of homogeneous polynomials and $F(\mathbf{x}) = \mathbf{0}$, with $\mathbf{x} = (x_0, \dots, x_n)$, by the Euler's theorem [34, Theorem 10.2] for homogeneous functions,

$$\sum_{j=0}^n x_j \frac{\partial f_i}{\partial x_j} = d_i \cdot f_i(\mathbf{x}) = 0$$

for each $i = 1, \dots, n$ where $d_i = \deg f_i$. So we have $DF(\mathbf{x}) \cdot \mathbf{x} = \mathbf{0}$, or $\mathbf{x} \in \ker DF(\mathbf{x})$. Thus there are n right singular vectors $\{\mathbf{v}^1, \dots, \mathbf{v}^n\}$ such that $\{\mathbf{v}^1, \dots, \mathbf{v}^n, \mathbf{x}\}$ form an orthonormal basis of \mathbb{C}^{n+1} with respect to the complex inner product and from the singular value decomposition of $DF(\mathbf{x})$ we have

$$U^H DF(\mathbf{x}) \begin{pmatrix} \mathbf{v}^1 & \dots & \mathbf{v}^n & \mathbf{x} \end{pmatrix} = \begin{pmatrix} \sigma_1 & & & 0 \\ & \ddots & & \vdots \\ & & \sigma_n & 0 \end{pmatrix}$$

for some unitary $n \times n$ matrix U . It follows that

$$\begin{pmatrix} U^H & \\ & 1 \end{pmatrix} \begin{pmatrix} DF(\mathbf{x}) \\ \mathbf{x}^H \end{pmatrix} \begin{pmatrix} \mathbf{v}^1 & \dots & \mathbf{v}^n & \mathbf{x} \end{pmatrix} = \begin{pmatrix} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_n & \\ & & & 1 \end{pmatrix}.$$

Simply put, the matrix $\begin{pmatrix} DF(\mathbf{x}) \\ \mathbf{x}^H \end{pmatrix}$ has singular values $\sigma_1, \dots, \sigma_n$, and 1. But $1 \in [\sigma_n, \sigma_1]$ by assumption, so the maximum and the minimum singular value of the matrix $\begin{pmatrix} DF(\mathbf{x}) \\ \mathbf{x}^H \end{pmatrix}$ are still σ_1 and σ_n respectively, and thus its condition number is

$$(8) \quad \text{cond} \begin{pmatrix} DF(\mathbf{x}) \\ \mathbf{x}^H \end{pmatrix} = \frac{\sigma_1}{\sigma_n}.$$

Namely, the condition number of the path at this point is only determined by the singular values of $DF(\mathbf{x}) = \hat{H}_{\mathbf{x}}(\mathbf{x}, t)$. We can therefore conclude that our choice of the horizontal lift of the projective path based on (5) *does not pollute the path condition* in the sense that it does not make it any worse.

10. Numerical results: A case study of the 5-body central configuration problem. One desirable fact of the path tracking on S^{2n+1} proposed in this article is that all points in S^{2n+1} have coordinates with norm 1. This eliminates the problem of having large coordinates, a common problem in affine path tracking. Figure 4 shows the path condition along an actual affine path provided by the polyhedral homotopy method [22] for solving the 5-body central configuration problem with the specific formulation described in [21]. (It is a system of 20 equations, 20 unknowns, and a total degree of $1,787,822,080 = 4^{10} \cdot 3^{10}$.) Evidently, the path condition grows rapidly in the “middle” of the path to as high as 10^{18} , and floating point arithmetic with much higher precision must be used in order to track this path with confidence. Upon closer inspection, this particular path has large components at the point where the path condition is high. This represents a typical case when a path is close to escaping \mathbb{C}^n . Naturally, performing path tracking in S^{2n+1} solves the problem.

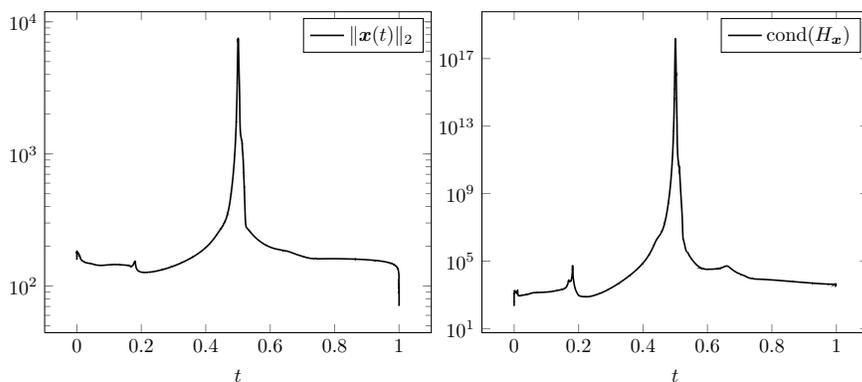


FIG. 4. The norm $\|\mathbf{x}(t)\|_2$ (on the left) and the path condition number (on the right) of a single path defined by polyhedral homotopy in \mathbb{C}^n .

Figure 5 exhibits the side-by-side comparison between the path condition along the affine path and the path condition along the horizontal lift of the associated projective path when the spherical projective path tracking we proposed is used. The difference is quite clear, the bad path condition, caused largely by large coordinates is completely eliminated: the path condition keeps a constant of roughly 10^2 along the entire path, and standard double precision floating point arithmetic is more than sufficient to handle the new path on S^{2n+1} .

The improved path condition certainly leads to higher confidence in the end point we obtain. A pleasant side effect is that the time required to track this particular path is also significantly reduced as shown in Table 1.

Here we isolated one particular path to illustrate the benefit of spherical projective path tracking. Table 2 shows the total time required to track all paths that lead to real regular solutions (since only this type of solutions are of physical interest in

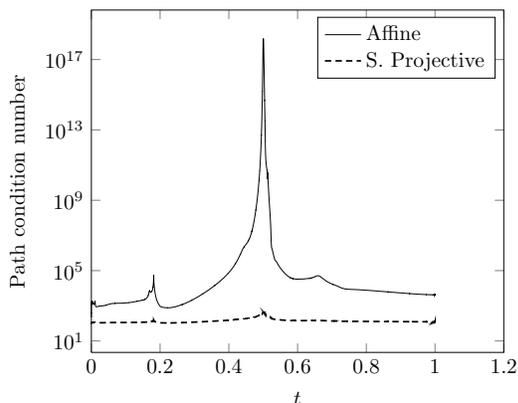


FIG. 5. The path condition number of a single path defined by polyhedral homotopy in the affine space \mathbb{C}^n (solid line) and that of the horizontal lift of the associated projective path on S^{2n+1} (dotted line).

TABLE 1

The amount of time it takes to track this particular path using different methods. These results are obtained on a computer with Intel Xeon E5620 CPU running at 2.40GHz and 16Gb of RAM.

Path tracking method	Time required for this path
Affine path tracking in \mathbb{C}^n	2455ms
Projective path tracking on S^{2n+1}	260ms

the context of the original problem). We can see that the spherical projective path tracking is strongly competitive in terms of the overall time consumption.

TABLE 2

The amount of time it takes to track all paths that leads to regular real solutions. Those paths that escapes \mathbb{C}^n or converge to singular solutions in the end are ignored as they require the use of “singular endgame” which is outside the scope of this article.

Path tracking method	Time required to obtain all real solutions
Affine path tracking in \mathbb{C}^n	16,433s \approx 4.5hrs
Projective path tracking on S^{2n+1}	14,767s \approx 4.1hrs

11. Numerical results: a comparison to path tracking with projective Newton’s iterations. A very similar path tracking algorithm using Projective Newton’s iterations alone was introduced in [14] and [27]. It was since used as the basis for many complexity analysis related to solving polynomial systems by the homotopy continuation method (A list of references is provided in the introduction). The analysis of the complexity of Euler’s other similar methods can be found in earlier works such

as [17] and [18]. In order to solve specific real-world problems, the focus of our work is quite different: Here the predictor and corrector scheme is chosen from a consideration of the efficiency in actual numerical computation where the exact wall-clock time taken is of the ultimate importance.

From numerical ODE, the algorithms that use Projective Newton's iterations alone with no predictor, as proposed in [14] and [27], can be considered as a special case of the predictor-corrector scheme where the predictor simply does nothing at all. Such a predictor would be of zero-th order accuracy. It is well known in the context of numerical ODE that Euler's method is a first order predictor which has great advantage over the zero-th order predictor in terms of both efficiency and reliability. To justify the addition of the spherical Euler's predictor with data from actual numerical computation, we present the following examples.

In all the tables and figures in this section, "P. Newton" stands for the path tracking method with projective Newton's iterations alone (as introduced in [14] and [27]) and "S.P. E/N" stands for the combination of spherical projective Euler's method and spherical projective Newton's iterations (as proposed in this paper). The two methods are each used to solve the `eco14`[24] system (14 equations, 14 unknowns, with total degree of 1,062,882) 1000 times with randomly generated coefficients and liftings to account for the randomized nature of the Polyhedral Homotopy method. Table 3 shows the number of steps it takes for each method. For the method that uses P. Newton alone, a "step" is defined to be a series of P. Newton iterations (that converges or fails to converge) at a t value. For the S.P. E/N combination, a step is simply a single S.P. Euler's prediction followed by a series of S.P. Newton's iterations (that converges or fails to converge). Notice that P. Newton method uses 7.61 times more steps than S.P. E/N method on average. Moreover, P. Newton method performs less consistently in the sense that the standard deviation is more than 100 times greater than the standard deviation obtained using the S.P. E/N method.

TABLE 3

The minimum, maximum, and mean number of steps it takes to track all solutions paths for the `eco14`[24] system using the two different methods (1000 runs each, which represent 1000 different set of paths). Last row shows the standard deviation of the two samples.

	P. Newton	S.P. E/N	Ratio
Min. number of steps	2,121,996.00	388,424.00	5.46
Max. number of steps	67,144,909.00	525,088.00	127.87
Mean number of steps	3,538,142.09	464,697.71	7.61
Std. deviation	2,082,192.75	19,504.55	106.75

Even though a spherical projective Euler's predictor introduces additional costs over the method that uses projective Newton iterations alone, such a first order

predictor offers much better predictions and resulting in a much lower overall running time. Table 4 shows the actual time, in seconds, spent on tracking homotopy paths using each method. The average time consumption for P. Newton method is 5.974 times that of the S.P. E/N method, and the standard deviation for P. Newton method is 71.206 times that of the S.P. E/N method. In other words, as far as this particular system is concerned, the 1000 random runs shows that the S.P. E/N method is more efficient and far more consistent.

TABLE 4

The minimum, maximum, and mean time it takes to track all solutions paths for the eco14[24] system using the two methods. Data are collect over 1000 runs using each method (1000 runs each, which represent 1000 different set of paths). Last row shows the standard deviation of the two samples.

	P. Newton	S.P. E/N	Ratio
Min. time	103.087s	22.654s	4.551
Max. time	2539.260s	31.422s	80.812
Mean time	160.561s	26.879s	5.974
Std. deviation	78.111s	1.096s	71.206

The difference in consistency is even more visible using the histograms as shown in Figure 6. While the running time, over 1000 different runs, for the S.P. E/N method spans a very narrow range (22 to 32 seconds), the histogram for the P. Newton method shows a “long tail” and spans a much wider range (100 to more than 2500 seconds).

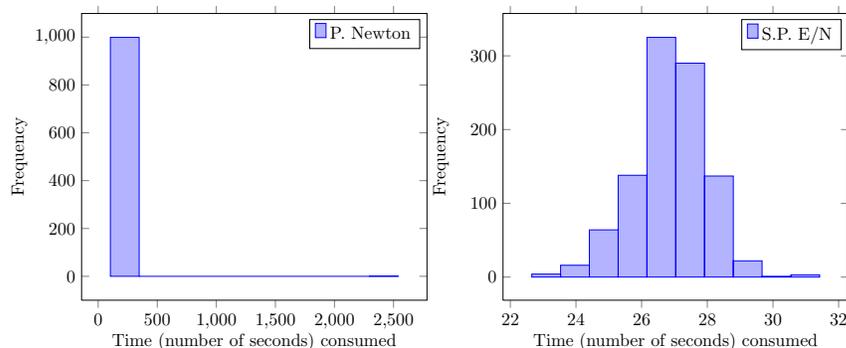


FIG. 6. *A comparison between the histograms representing the distribution of the time consumed to track all paths for the eco14[24] system using P. Newton method alone (left) and the S.P. E/N method(right). The timing data are collect over 1000 runs using each method. The size of the rectangles represents the frequency within the 1000 runs. E.g., the tallest rectangle in the histogram on the right represent that over 300 runs using S.P. E/N method consumed between 26 and 27 seconds.*

The same difference can be observed in solving a list of polynomial systems. Table 5 shows the difference in running time between the two methods when applied to

That is, we can decompose $T_{\mathbf{x}}S^{2n+1}$ into the direct sum of two subspaces that are orthogonal with respect to the inner product given by $g_{S^{2n+1}}$ at \mathbf{x} . The subspace $\mathcal{H}_{\mathbf{x}}$ is then described by the following formula, as listed in Section 6:

PROPOSITION (Proposition 2 of Section 6) Via the isomorphism $T_{\mathbf{x}}\mathbb{C}^{n+1} \cong \mathbb{C}^{n+1}$, $\mathcal{H}_{\mathbf{x}}$ is given by the subspace

$$\mathcal{H}_{\mathbf{x}} = \{\mathbf{v} \in \mathbb{C}^{n+1} \mid \langle \mathbf{x}, \mathbf{v} \rangle_{\mathbb{C}} = \mathbf{x}^H \mathbf{v} = 0\}.$$

Proof. At \mathbf{x} , the tangent space $T_{\mathbf{x}}S^{2n+1}$ of S^{2n+1} , as an embedded submanifold of \mathbb{R}^{2n+2} , can be identified with the linear subspace $\{\mathbf{x}\}^{\perp} \subseteq T_{\mathbf{x}}\mathbb{R}^{2n+2} \cong \mathbb{R}^{2n+2}$. The space $\mathcal{V}_{\mathbf{x}}$ is simply the tangent space $T_{\mathbf{x}}[\mathbf{x}]$ of the orbit of \mathbf{x} under the action of S^1 . It is clear that near \mathbf{x} , the one (real) dimensional submanifold $[\mathbf{x}]$ is parametrized by $\gamma(\theta) = e^{i\theta} \mathbf{x}$ with $\gamma(0) = \mathbf{x}$. So

$$\dot{\gamma}(0) = ie^{i0} \mathbf{x} = i\mathbf{x}.$$

As a vector in $T_{\mathbf{x}}\mathbb{R}^{2n+2}$, it is a generator of the one dimensional vector space $T_{\mathbf{x}}[\mathbf{x}]$. Thus

$$\mathcal{V}_{\mathbf{x}} = T_{\mathbf{x}}[\mathbf{x}] = \text{span}\{i\mathbf{x}\} \text{ and } T_{\mathbf{x}}S^{2n+1} = \{\mathbf{x}\}^{\perp}.$$

Since $\mathcal{H}_{\mathbf{x}}$ is defined to be the orthogonal compliment of $\mathcal{V}_{\mathbf{x}}$ in $T_{\mathbf{x}}S^{2n+1}$. So it is simply $\{i\mathbf{x}\}^{\perp} \cap \{\mathbf{x}\}^{\perp}$ in $T_{\mathbf{x}}\mathbb{R}^{2n+2} \cong \mathbb{R}^{2n+2}$, i.e., it is the set of vector \mathbf{v} such that

$$\langle i\mathbf{x}, \mathbf{v} \rangle_{\mathbb{R}} = 0$$

$$\langle \mathbf{x}, \mathbf{v} \rangle_{\mathbb{R}} = 0$$

which is equivalent to the complex equation $\langle \mathbf{x}, \mathbf{v} \rangle_{\mathbb{C}} = 0$ based on the observation from Equation (9). Therefore the horizontal space can be characterized as

$$\mathcal{H}_{\mathbf{x}} = \{\mathbf{v} \in T_{\mathbf{x}}\mathbb{R}^{2n+2} : \langle \mathbf{x}, \mathbf{v} \rangle_{\mathbb{C}} = 0\}.$$

□

B. The distance formula for S^{2n+1} . In the Newton's corrector, the distance between two points on S^{2n+1} is used as an important criterion for the convergence test. In this section we shall state the distance formula for two points on S^{2n+1} .

It is clear that the distance between two points $\mathbf{x} = (a, b)$ and $\mathbf{x}' = (1, 0)$ on the unit circle $S^1 \subset \mathbb{R}^2$ must be the length of the shorter arch between the two points on the unit circle. This length is given by the angle between $\mathbf{x} = (a, b)$ and the horizontal axis on which \mathbf{x}' lies:

$$d_{S^1}(\mathbf{x}, \mathbf{x}') = \cos^{-1}(a) = \cos^{-1} \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}^{\top} \begin{pmatrix} a \\ b \end{pmatrix} \right) = \cos^{-1}(\langle \mathbf{x}, \mathbf{x}' \rangle_{\mathbb{R}}).$$

The same formula works in general for S^{2n+1} . For two distinct points $\mathbf{x}, \mathbf{x}' \in S^{2n+1} \subset \mathbb{C}^{n+1} \approx \mathbb{R}^{2n+2}$, there exists a unique 2-dimensional linear subspace of \mathbb{R}^{2n+2} that contains both \mathbf{x} and \mathbf{x}' . This subspace intersects S^{2n+1} on a circle, the **great circle** through \mathbf{x} and \mathbf{x}' . It is intuitively clear that the distance between them must be the length of the shorter arc joining the two on the great circle (geodesic) that passes through both of them. An indirect proof of this fact can be found in [19, Proposition 5.13]. There exists an orthogonal change of coordinates after which the two points \mathbf{x} and \mathbf{x}' together with the great circle passing through them lie flat in $\mathbb{R}^2 \subset \mathbb{R}^{2n+2}$. Indeed, this change of coordinates is given explicitly by the QR decomposition: there exists a $(2n+2) \times (2n+2)$ real orthogonal matrix Q such that

$$Q \begin{pmatrix} \mathbf{x} & \mathbf{x}' \end{pmatrix} = \begin{pmatrix} 1 & a \\ 0 & b \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix}$$

for some $a, b \in \mathbb{R}$. So the orthogonal transformation Q maps the great circle through those two points to the unit circle $S^1 \subset \mathbb{R}^2 \subset \mathbb{R}^{2n+2}$, and we can thus compute the distance as we did in the previous case. Since Q , being orthogonal, preserves dot product, it follows that

$$d_{S^{2n+1}}(\mathbf{x}, \mathbf{x}') = \cos^{-1}(a) = \cos^{-1}((Q\mathbf{x})^\top(Q\mathbf{x}')) = \cos^{-1}(\langle \mathbf{x}, \mathbf{x}' \rangle_{\mathbb{R}}).$$

In this case, the distance is still given by the arccosine of the real inner product of the two points as vectors in \mathbb{R}^{2n+2} .

C. Properties of horizontal lift. In this section we will restate and prove the basic properties of the horizontal lift of a given projective path and a starting point.

PROPOSITION (Proposition 1 of Section 5) Fix the projective path $\hat{\gamma} \subset \mathbb{C}\mathbb{P}^n$ with its smooth parametrization $\hat{\mathbf{x}} : [0, 1] \rightarrow \mathbb{C}\mathbb{P}^n$ and a starting point $\mathbf{x}_0 \in \pi^{-1}(\hat{\mathbf{x}}(0))$. Let Γ be the set of all smoothly parametrized curves $\mathbf{x} : [0, 1] \rightarrow S^{2n+1}$ such that $\pi \circ \mathbf{x} = \hat{\mathbf{x}}$ and $\mathbf{x}(0) = \mathbf{x}_0$. Also let $\mathbf{x}^{\mathcal{H}}$ be the unique horizontal lift of $\hat{\gamma}$. Then

- i. $\|\dot{\mathbf{x}}^{\mathcal{H}}(t)\| \leq \|\dot{\mathbf{x}}(t)\|$ for all $\mathbf{x}(t) \in \Gamma$.
- ii. For any $\mathbf{x}(t) \in \Gamma$ and $t_0 \in (0, 1)$, there is a sufficiently small $\epsilon \in \mathbb{R}^+$ such that $\int_{t_0}^{t_0+\epsilon} \|\dot{\mathbf{x}}^{\mathcal{H}}(t)\| dt \leq \int_{t_0}^{t_0+\epsilon} \|\dot{\mathbf{x}}(t)\| dt$.
- iii. $\int_0^1 \|\dot{\mathbf{x}}^{\mathcal{H}}(t)\| dt = \int_0^1 \|\dot{\mathbf{x}}(t)\| dt$.

Proof.

- i. Fix any $t \in (0, 1)$ and $\mathbf{x}(t) \in \Gamma$. By the decomposition $T_{\mathbf{x}(t)}S^{2n+1} = \mathcal{H}_{\mathbf{x}(t)} \oplus \mathcal{V}_{\mathbf{x}(t)}$, $\dot{\mathbf{x}}(t)$ can be written uniquely as $\mathbf{h} + \mathbf{v}$ with $\mathbf{h} \in \mathcal{H}_{\mathbf{x}(t)}$ and $\mathbf{v} \in \mathcal{V}_{\mathbf{x}(t)}$. Since $\pi \circ \mathbf{x} = \pi \circ \mathbf{x}^{\mathcal{H}} = \hat{\mathbf{x}}$ by assumption, thus $\pi_*\dot{\mathbf{x}}(t) = \pi_*\dot{\mathbf{x}}^{\mathcal{H}}(t) = \dot{\hat{\mathbf{x}}}(t)$. But $\mathbf{x}^{\mathcal{H}}$ is horizontal, so

$$\|\dot{\mathbf{x}}^{\mathcal{H}}(t)\|^2 = \|\mathbf{h}\|^2 \leq \|\mathbf{h}\|^2 + \|\mathbf{v}\|^2 = \|\mathbf{h} + \mathbf{v}\|^2 = \|\dot{\mathbf{x}}(t)\|^2.$$

- ii. This follows immediately from the previous part.
 iii. Since $\mathbf{x}^{\mathcal{H}}(t)$ is horizontal and π_* acts as an isometry on the horizontal space,

$$\begin{aligned} \int_0^1 \|\dot{\mathbf{x}}^{\mathcal{H}}(t)\| dt &= \int_0^1 \sqrt{g_{S^{2n+1}}(\dot{\mathbf{x}}^{\mathcal{H}}(t), \dot{\mathbf{x}}^{\mathcal{H}}(t))} dt \\ &= \int_0^1 \sqrt{g_{\mathbb{C}\mathbb{P}^n}(\pi_*(\dot{\mathbf{x}}^{\mathcal{H}}(t)), \pi_*(\dot{\mathbf{x}}^{\mathcal{H}}(t)))} dt \\ &= \int_0^1 \sqrt{g_{\mathbb{C}\mathbb{P}^n}(\dot{\mathbf{x}}(t), \dot{\mathbf{x}}(t))} dt \\ &= \int_0^1 \|\dot{\mathbf{x}}(t)\| dt. \end{aligned}$$

□

REFERENCES

- [1] E. ALLGOWER AND K. GEORG. *Introduction to numerical continuation methods*, volume 45. Society for Industrial and Applied Mathematics, 2003.
- [2] G. ATTARDI AND C. TRAVERSO. *The PoSSo library for polynomial system solving*. Proc. of AIHENP95, 1995.
- [3] D. J. BATES, J. D. HAUENSTEIN, A. J. SOMMESE, AND C. W. WAMPLER. *Adaptive multiprecision path tracking*. SIAM Journal on Numerical Analysis, 46:2(2008), pp. 722–746.
- [4] D. J. BATES, J. D. HAUENSTEIN, A. J. SOMMESE, AND C. W. WAMPLER. *Numerically Solving Polynomial Systems with Bertini*. Society for Industrial and Applied Mathematics, 2013.
- [5] C. BELTRÁN. *A continuation method to solve polynomial systems and its complexity*. Numerische Mathematik, 117:1(2011), pp. 89–113.
- [6] C. BELTRÁN AND A. LEYKIN. *Certified numerical homotopy tracking*. Experimental Mathematics, 21:1(2012), pp. 69–83.
- [7] C. BELTRÁN AND A. LEYKIN. *Robust certified numerical homotopy tracking*. Foundations of Computational Mathematics, 13:2(2013), pp. 253–295.
- [8] C. BELTRÁN AND L. PARDO. *On Smale’s 17th problem: A probabilistic positive solution*. Foundations of Computational Mathematics, 8:1(2008), pp. 1–43.
- [9] C. BELTRÁN AND L. PARDO. *Smale’s 17th problem: Average polynomial time to compute affine and projective solutions*. Journal of the American Mathematical Society, 22:2(2009), pp. 363–385.
- [10] C. BELTRÁN AND L. PARDO. *Fast linear homotopy to find approximate zeros of polynomial systems*. Foundations of Computational Mathematics, 11:1(2011), pp. 95–129.
- [11] C. BELTRÁN AND M. SHUB. *The complexity and geometry of numerically solving polynomial systems*. In: Contemporary Math. AMS available on arXiv.
- [12] C. BELTRÁN AND M. SHUB. *Complexity of Bézout’s theorem VII: Distance estimates in the condition metric*. Foundations of Computational Mathematics, 9:2(2009), pp. 179–195.
- [13] G. BJÖRCK AND R. FRÖBERG. *A faster way to count the solutions of inhomogeneous systems of algebraic equations, with applications to cyclic- n -roots*. Journal of Symbolic Computation, 12:3(1991), pp. 329–336.
- [14] L. BLUM, F. CUCKER, M. SHUB, AND S. SMALE. *Complexity and real computation*. Springer-Verlag, 1998.
- [15] J.-P. DEDIEU, G. MALAJOVICH, AND M. SHUB. *Adaptive step-size selection for homotopy methods to solve polynomial equations*. IMA Journal of Numerical Analysis, 33:1(2013), pp. 1–29.

- [16] T. GUNJI, S. KIM, M. KOJIMA, A. TAKEDA, K. FUJISAWA, AND T. MIZUTANI. *PHoM—a polyhedral homotopy continuation method for polynomial systems*. Computing, 73:1(2004), pp. 57–77.
- [17] M.-H. KIM. *Computational complexity of the Euler type algorithms for the roots of complex polynomials*. PhD thesis, City University of New York Graduate Center, 1985.
- [18] M.-H. KIM. *On approximate zeros and root finding algorithms for a complex polynomial*. Mathematics of Computation, 51:184(1988), pp. 707–719.
- [19] J. M. LEE. *Riemannian manifolds: An introduction to curvature*, volume 176. Springer-Verlag, 1997.
- [20] 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*. Computing, 83:2(2008), pp. 109–133.
- [21] T. L. LEE AND M. SANTOPRETE. *Central configurations of the five-body problem with equal masses*. Celestial Mechanics and Dynamical Astronomy, 104:4(2009), pp. 369–381.
- [22] T. Y. LI. *Numerical solution of polynomial systems by homotopy continuation methods*. In: P. G. Ciarlet, editor, Handbook of Numerical Analysis, volume 11, pages 209–304. North-Holland, 2003.
- [23] A. P. MORGAN. *A transformation to avoid solutions at infinity for polynomial systems*. Applied mathematics and computation, 18:1(1986), pp. 77–86.
- [24] A. P. MORGAN. *Solving polynomial systems using continuation for engineering and scientific problems*, volume 57 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics, 2009.
- [25] M. SHUB. *Some remarks on Bézout’s theorem and complexity theory*. In: From topology to computation: proceedings of the Smalefest, pages 443–455. Springer, 1993.
- [26] M. SHUB. *Complexity of Bézout’s theorem VI: Geodesics in the condition (number) metric*. Foundations of Computational Mathematics, 9:2(2009), pp. 171–178.
- [27] M. SHUB AND S. SMALE. *Complexity of Bézout’s theorem I: Geometric aspects*. Journal of the AMS, 6:2(1993), pp. 459–501.
- [28] M. SHUB AND S. SMALE. *Complexity of Bézout’s theorem II: Volumes and probabilities*. In: F. Eyssette and A. Galligo, editors, Computational Algebraic Geometry, volume 109 of Progress in Mathematics, pages 267–285. Birkhuser Boston, 1993.
- [29] M. SHUB AND S. SMALE. *Complexity of Bézout’s theorem III: Condition number and packing*. Journal of Complexity, 9:1(1993), pp. 4–14.
- [30] M. SHUB AND S. SMALE. *Complexity of Bézout’s theorem V: Polynomial time*. Theoretical Computer Science, 133:1(1994), pp. 141–164.
- [31] M. SHUB AND S. SMALE. *Complexity of Bézout’s theorem IV: Probability of success; extensions*. SIAM Journal on Numerical Analysis, 33:1(1996), pp. 128–148.
- [32] A. J. SOMMESE AND C. W. WAMPLER. *The Numerical solution of systems of polynomials arising in engineering and science*. World Scientific Pub Co Inc, 2005.
- [33] J. VERSHELDE. *Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation*. ACM Transactions on Mathematical Software (TOMS), 25:2(1999), pp. 251–276.
- [34] R. WALKER. *Algebraic curves*. Springer-Verlag, 1978.
- [35] J. WILKINSON. *The algebraic eigenvalue problem*. Oxford University Press, USA, 1988.

