An Approach for Certifying Homotopy Continuation Paths: Univariate Case

Juan Xu
Beihang University
Beijing, China
xujuan0505@126.com

Michael Burr*
Clemson University
Clemson, SC
burr2@clemson.edu

Chee Yap†
Courant Institute, NYU
New York, NY
yap@cs.nyu.edu

ABSTRACT

Homotopy continuation is a well-known approach in numerical root-finding. In recent years, certified algorithms for homotopy continuation based on Smale's alpha-theory have been developed. Such approaches enforce very strong requirements at each step, leading to small step sizes. In this paper, we propose an approach that is independent of alpha-theory. It is based on the weaker notion of well-isolated approximations to the roots. We apply it to univariate polynomials and provide experimental evidence of its feasibility.

ACM Reference Format:
Juan Xu, Michael Burr, and Chee Yap. 2018. An Approach for Certifying Homotopy Continuation Paths: Univariate Case. In *Proceedings of ACM International Symposium on Symbolic and Algebraic Computation (ISSAC'18).* ACM, New York, NY, USA, 8 pages.

1 INTRODUCTION

Homotopy continuation is a method for approximating the solutions of a system of polynomials f by tracking and deforming the approximate solutions of an easier system of polynomials g. We call f the target system and g the start system. We are given f and the homotopy method chooses g as well as the homotopy $H(z,t)$ such that $H(z,0) = g$ and $H(z,1) = f$. The recent development of software packages for homotopy continuation has led to significant interest and development of these techniques both within the mathematics community as well as for their applications. Some of the main homotopy continuation packages include `Bertini` [1], `PHCpack` [25], `Hom4PS` [18], and `NAG4M2` [19], see also the references therein. Homotopy continuation algorithms (broadly speaking) consist of three phases: (1) choosing an appropriate start system g and approximating its roots, (2) tracking the roots as g is deformed to the target system f, and (3) applying end-game techniques to analyze singular solutions.

In this paper, we provide a certified version of Phase (2) for univariate polynomials using well-isolated approximations to roots, subdivisions, and interval methods. To this end, we assume that f has only simple roots[1] in order to avoid the end-game issues in Phase (3). Approximating the roots of the start system g in Phase (1) is also not an issue, especially when discussing univariate roots. In this case, it is standard to choose $g(z) = z^n - 1$ when $\deg(f) = n$. We next describe the central problem of Phase (2).

A **solution path** for this homotopy is a continuous function $\alpha(t)$ such that $H(\alpha(t), t) = 0$ for all $t \in [0,1]$. We call $H(z,t)$ a **good homotopy** if the n solution paths $\{\alpha^i : i = 1, \ldots, n\}$ are non-singular and do not diverge to infinity. We restrict attention to good homotopies in this paper.

In general, path tracking consists of the following iterative procedure: to track the ith path, assume we are given z_0^i as an approximation to $z^i(0)$. Starting with $(z_0^i, 0)$, for $j \geq 0$, assume we can generate a pair (z_{j+1}^i, t_{j+1}^i) from (z_j^i, t_j^i) such that $1 \geq t_{j+1}^i > t_j^i$. Intuitively, each z_j^i is an approximation to $\alpha^i(t_j)$. If z_{j+1}^i is not an approximation to $\alpha^i(t_{j+1})$, we say that **path jumping** has occurred. Such errors are the central concern Phase (2). Although the sequence $\{z_j^i : j \geq 0\}$ is explicit, the solution path $\alpha^i(t)$ is only implicit. The problem is how to ensure correctness without direct access to $\alpha^i(t)$. The preceding discussions are quite informal, as the notion "z_j^i is (or is not) an approximation of $\alpha^i(t_j)$" is imprecise. We now turn to ways to make it precise and effective.

The key idea lies[2] in defining what it means for $z_0 \in \mathbb{C}^m$ to identify a unique root z_* of $h : \mathbb{C}^m \to \mathbb{C}^m$. It is tempting to choose z_* to be the root of h closest to z_0, but this idea is not effective. Typically, this identification is done via Newton iteration[3]. In the literature, $z_0 \in \mathbb{C}^m$ typically identifies a root z_* of h in one of two ways: (i) z_0 lies in the Newton basin of z_*, or (ii) z_0 converges quadratically (from the start) towards z_*. Note that Condition (ii) is stricter than Condition (i). By applying Newton iteration to z_0 sufficiently many times, we can (eventually) confirm these conditions, but this is not effective. Smale's α-theory [10] provides a way to check Condition (ii) without applying the Newton operator. In particular, one can effectively evaluate the function[4] α_h such that if this function at z_0 is less than some universal constant (e.g., 0.158), then z_0 satisfies Condition (ii). The certified homotopy methods of Beltran-Leykin

*Partially supported by a grant from the Simons Foundation (#282399 to Michael Burr) and NSF Grant CCF-1527193.
†Partially supported NSF Grants Nos. CCF-1423228 and CCF-1564132.

ISSAC'18, July 2018, New York, New York USA

[1] In practice, due to numerical issues, many homotopy continuation algorithms assume that the system starts at $t = 1$ and ends at time $t = 0$. This allows for more numerical precision near potentially singular solutions of the target system. Since we assume that f has simple roots, reversing the homotopy does not impact the practical efficiency of our approach.
[2] Briefly, we consider the multivariate setting.
[3] This is natural since, until the current paper, the corrector steps of homotopy continuation use Newton iteration.
[4] The α_h function is from Smale's $alpha$-theory. It is defined for the polynomial h and should not be confused with solution paths α^i.

[6] and Bürgisser-Cucker [11] avoid path jumping by ensuring that the z_j^i's satisfy some strong form of this alpha test.

In this paper, we develop a weaker way for z_0 to identify a root z_* that can be used to track a path $\alpha(t)$: we say that $z_0 \in \mathbb{C}$ is a **well-isolated approximation** of h if its distance to the closest root of h is at most one-third the distance to the second closest root of h. To make this effective, assume that such a z_0 comes with a witness radius r_0. That is, we have a disc $\Delta(z_0, r_0)$ centered at z_0 of radius r_0 with the following property: $\Delta(z_0, r_0)$ contains a unique root of $h(z)$, and $\Delta(z_0, 3r_0)$ contains no additional roots of $h(z)$. In this case, we call $\Delta(z_0, r_0)$ a **well-isolator** of a root of h. A key contribution of this paper is to show how, given a well-isolator $\Delta(z_j, r_j)$ for $h_j(z) = H(z, t_j)$, we can compute a time $t_{j+1} \in (t_j, 1]$ and a well-isolator $\Delta(z_{j+1}, r_{j+1})$ for h_{j+1}, such that both well-isolators identify the same solution path $\alpha(t)$. We can use recently developed soft Graeffe-Pellet tests [3, 4] to verify a well-isolator $\Delta(z, r)$.

The main motivation for our approach comes from an intuition that using the approximate roots of α-theory is unnecessarily strong for the goal of avoiding path jumping. We show that the weaker notion of well-isolated approximations is sufficient for this purpose. In consequence, we expect our approach to take fewer time steps. This is supported by **Table 1** which compares the number of time steps using α-theory versus our well-approximated root approach. The data for α-based tracking is from Beltrán and Leykin [5, Section 9.1 and Table 3]. Each column correspondsto a homotopy $H_m(x_1, t) = x_1^2 - (1 + mt)$ for some choice of m. We run our tracker using $\gamma = 1$ and $\gamma = \boldsymbol{i}$. Overall, our well-isolated tracker has 4 to 8 times fewer steps than the alpha tracker.

In addition, α-theory may have been employed in previous certified algorithms because it pairs well with Newton iteration, the typical corrector in homotopy continuation. Our approach, however, does not use Newton iteration – this is an intended (possibly surprising) feature of our approach. We design our time steps $\delta t_j := t_{j+1} - t_j$ to be correlated with r_0, and so a larger r_0 would allow us to take larger time steps. Newton iteration, on the other hand, would result in more steps if r_0 were very small.

Since these ideas are new, and the univariate case already has various subtleties, we focus on the univariate case in this paper.

1.1 Literature Survey.

There are various approaches to certify the output of a homotopy continuation algorithm: see [5–7, 11–14, 16, 17, 20], and the references therein. The first rigorously justified algorithm in the Bit Model of computation (this includes Turing Machines) is from Beltran and Leykin [6], with implementation in [5]. The emphasis on a computational model is important: most algorithms are described in the Algebraic Model (in particular the Real RAM Model or BSS Model) where each algebraic operation is treated as a primitive operation. It is well-known that the correct implementation of algebraic algorithms is highly nontrivial. Conversely, the correct implementation of bit model algorithms is comparatively easy.

It is generally understood that certified homotopy algorithms mean that each path is correctly tracked. But there is a weaker notion of simply certifying the output as in [14]. Strictly speaking, this type of certification is not specific to homotopy continuation (since we can use the method to certify the output of any root finding algorithm). This approach amounts to checking that the final root is actually an approximate zero, but it is helpless to recover any lost roots due to path jumping. On the other hand, certifying the paths ensures that no roots are lost.

In [5–7, 11, 13, 22–24], the authors present certified complexity analyses or homotopy continuation algorithms which remain very close to the path, such as near the *alpha*-region of convergence. In [12] the authors use an *a posteriori* approach to certify the correctness of a path by reversing a non-certified homotopy step. In [15–17, 20] the authors present algorithms for path tracking using interval arithmetic. These papers track general curves and are not tuned to studying homotopy continuation-based paths.

1.2 Overview of Paper

In Section 2 we establish some common notation, including clarifying the good homotopy assumption of this work.

In Section 3, we give an overview of our main algorithm, the Well-Isolated Tracker. Couched in the familiar predictor-correct framework of homotopy continuation, we expose the details of our predictor and corrector modules. This is the main locus of our technical contribution. We develop well-isolated approximations into a tool for correct path tracking. Our predictor uses the classic Euler steps (this is absent in current α-theory approaches). Instead of Newton iteration, we use bisection using a recently developed soft Graeffe-Pellet test.

Sections 4 and 5 provide technical details of the Update Subroutine, which is a loop for the interaction between the predictor and corrector. The main algorithm is an outer loop around this inner loop. Section 4 presents a slightly abstract view ("geometric meaning") of the Update Subroutine, and Section 5 instantiates these with explicit formulas necessary for implementation. Also in Section 5, we take care of issues of numerical approximations using interval methods. Because of the intricate predictor-corrector interaction, the correctness of the overall algorithm is nontrivial. We first prove the termination of the inner loop (`Update` Subroutine). The termination of the outer loop (the main algorithm) requires an improved corrector strategy. In Section 6, we present experimental results and conclude in Section 7.

2 BASIC SETUP AND NOTATION

In this paper, we follow many of the standard conventions used in homotopy continuation, e.g., see [2]. Our goal is to find the roots of a given target univariate polynomial $f \in \mathbb{Q}[x]$. Suppose that $g \in \mathbb{Q}[x]$ is the starting polynomial of the same degree as f and whose roots we already know. We connect f and g using a linear homotopy with the standard γ-trick, i.e., for $\gamma \in \mathbb{C}$, $H(x, t) = H_\gamma(x, t) := tf(x) + \gamma(1 - t)g(x)$. We observe that $H(x, 0) = \gamma g(x)$ and $H(x, 1) = f(x)$. Given a root $z \in \mathbb{C}$ of g, the **solution path** for z is a continuous function $\alpha : [0, 1] \to \mathbb{C}$ where $\alpha(0) = z$ and $H(\alpha(t), t) = 0$ for all $t \in [0, 1]$.

Good Homotopy Assumption: *all solution paths are nonsingular and none diverge to infinity.* The results of this paper depend on this assumption, so it is good to clarify a potential ambiguity about the term "non-singularity of solution paths". Non-singularity implies that the solution paths are pairwise disjoint. The algebraic set $\{\alpha(t) : t \in [0, 1]\}$ might be regular, yet its parametrization

Parameter m:	10	40	70	100	1000	2000	3000	4000	5000	10000	20000	30000
# of Alpha Steps (from [5, p. 288]):	184	250	276	292	395	426	446	457	468	499	530	547
# of Well-Isolated Steps: ($\gamma = 1$)	21	32	38	39	63	68	68	75	75	79	87	88
# of Well-Isolated Steps: ($\gamma = i$)	38	47	49	47	76	73	85	89	84	91	96	96

Table 1: Comparison against Alpha Tracking for the Homotopy $H(x,t) = x^2 - (1 + mt)$. The number of steps are identical for both roots of the polynomials.

$\alpha(t)$ might be singular, i.e., $\alpha'(t)$ might not be defined for some t. We interpret the Good Homotopy Assumption to mean that the parametrization $\alpha(t)$ itself is well-defined. To see what this entails, we differentiate $H(\alpha(t), t) = 0$ with respect to t:

$$\alpha'(t)\frac{\partial H}{\partial x}(\alpha(t), t) + \frac{\partial H}{\partial t}(\alpha(t), t) = 0.$$

Thus $\alpha'(t)$ is well-defined *provided* $\frac{\partial H}{\partial x}(\alpha(t), t)$ *does not vanish.* So we assume that the function $\frac{\partial H}{\partial x}(\alpha(t), t) \neq 0$ for all $t \in [0, 1]$, viewed as part of our good homotopy assumption. Note that the non-vanishing of $\alpha'(t)$ for all $t \in [0, 1]$ is the typical case for a random choice of γ. We define the function

$$G(x,t) = -\frac{\frac{\partial H}{\partial t}(x,t)}{\frac{\partial H}{\partial x}(x,t)} = \frac{\gamma g(x) - f(x)}{(1-t)\gamma g'(x) + t f'(x)}. \tag{1}$$

Observe that $\alpha'(t) = G(\alpha(t), t)$ for any solution path $\alpha(t)$.

Our algorithm proceeds by maintaining closed disks containing the roots of $H(x, t_0)$ for $t_0 \in [0, 1]$. We use the following notation for disks: $\Delta(m, r) \subseteq \mathbb{C}$ denotes the closed complex disk centered at $m \in \mathbb{C}$ of radius $r > 0$. Two convenient related notations follow: (i) for any real $k > 0$, let $k\Delta(m, r) := \Delta(m, kr)$, and (ii) for any $t \in [0, 1]$, let $\Delta(m, r, t) := \Delta(m, r) \times \{t\} \subseteq \mathbb{C} \times [0, 1]$. For a polynomial $h \in \mathbb{Q}[x]$, we say that $\Delta(m, r)$ is **well-isolating** (for h) if both $\Delta(m, r)$ and $3\Delta(m, r)$ contain a unique root of h.

In our tests, we appeal to the Graeffe-Pellet test[5] and interval methods for functions. Details on this test and these methods can be found in [4] and [21], respectively.

3 OVERVIEW OF ALGORITHM

In this section, we provide a high-level overview overview of our homotopy algorithm. We frame our algorithm in the standard predictor-corrector framework. Throughout this section, we fix a solution path $\alpha(t)$ from our homotopy.

3.1 Update Subroutine

Our homotopy algorithm maintains and updates a state during its algorithm. A **state** consists of the following data:

$$\sigma_0 = (m_0, r_0, t_0, \delta t_0) \in \mathbb{C} \times \mathbb{R}_{>0} \times [0, 1]^2. \tag{2}$$

This state is a **valid state** relative to a solution path $\alpha(t)$ provided $\Delta(m_0, r_0)$ is well-isolating for the root $\alpha(t_0)$ of the polynomial $H(x, t_0)$. Thus, m_0 is a well-isolated approximation of the root $\alpha(t_0)$. Additionally, δt_0 provides a suggestion for the next time step. In our algorithm, we assume that we begin with a valid initial state for $\gamma g(x) = H(x, 0)$, and, in this section, we focus on the update step that transforms σ_0 to a valid state σ_1 at time $t_1 > t_0$.

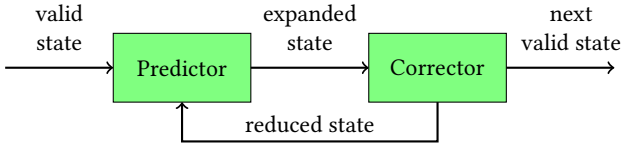

Figure 1: Update Subroutine

Fig. 1 is a schematic for our `Update` Subroutine that converts a valid state into another valid state at the next time instance. It has two modules, a predictor and a corrector. Starting with a valid state σ_0, the predictor produces an "expanded" state σ_1 and sends it to the corrector module. The corrector module either confirms that σ_1 is valid, or sends a "reduced" state back to the predictor. The "expanded" and "reduced" terminology is clarified below.

```
Well-Isolated Tracker Algorithm
Input: Well-isolator Δ(m0, r0) for start polynomial g
Output: Well-isolator of the valid state σ at time t = 1.
    Initialize state σ ← (m0, r0, 0, 2r0)
    While (the time of σ is < 1)
        σ ← Update(σ)
    Return the well-isolator in σ.
```

Figure 2: Main Algorithm for path tracking

Our main algorithm is the path tracker shown in Fig. 2. It is a while-loop around the `Update` Subroutine called the **outer loop**. The loop in `Update` is called the **inner loop**. The tracker terminates when the outer loop reaches a valid state at time $t = 1$. The input $\Delta(m_0, r_0)$, being a well-isolator of the starting polynomial g, defines a unique solution path $\alpha(t)$. Validity of states refers to this path. And the final valid state represents a well-isolated approximation of the root $\alpha(1)$ of the final polynomial f.

3.2 Corrector Module

In this section, we provide an overview of the corrector module of the predictor-corrector loop. The corrector module has three tests, which we call the `Bounded`, `On-Track`, and `Isolated` tests. The first two tests are computed on the "transition region" T between the valid state and the expanded state, i.e., $T \cap \mathbb{C} \times \{t_0\}$ is the starting valid disk $\Delta(m_0, r_0, t_0)$ and $T \cap \mathbb{C} \times \{t_1\}$ equals the disk $\Delta(m_1, r_1, t_1)$ corresponding to the expanded state. If all the three tests pass,

[5]The Graeffe-Pellet test could be used to certify the output of a homotopy continuation, in a similar way as the `alphaCertified` algorithm in [14], but we do not use it in the present paper

the corrector sends σ_1 to the output of the Update Subroutine; otherwise, it reduces the valid disk $\Delta(m_0, r_0)$ and/or the suggested step size δt_0. The construction of T and the details of the correctors are discussed in Section 4.

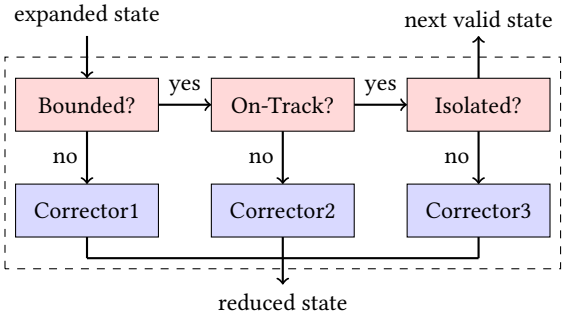

Figure 3: Corrector Module

The Good Homotopy Assumption implies $\frac{\partial H}{\partial x}(\alpha(t), t)$ is never zero for any solution path $\alpha(t)$. However, it is possible that this derivative vanishes for some $(x, t) \in T$. The Bounded test assures that $\frac{\partial H}{\partial x}(x, t)$ never vanishes within T, i.e., $|G(x, t)|$ is bounded.

The On-Track test guarantees that the region T contains the portion of the solution path α between t_0 and t_1, i.e., $\alpha([t_0, t_1]) \subseteq T$. This test assumes that the Bounded test has passed. The On-Track test checks that $\delta t = t_1 - t_0$ is sufficiently small.

The Isolated test guarantees that the disk $3\Delta(m_1, r_1)$ isolates a unique root of $H(x, t_1)$. If both the Bounded and On-Track tests are passed, we know that this unique root is actually $\alpha(t_1)$, and $\alpha(t_1)$ lies in $\Delta(m_1, r_1)$. Thus σ_1 is a valid state relative to the path $\alpha(t)$.

At a high level, the corrector module is described by the following pseudo-code: In this code, the Reduce function serves as the corrector function, and details on this function are provided in the next section. The parameters ε and ε' are often set to $\frac{1}{2}$ or $\frac{3}{4}$.

```
CORRECTOR MODULE
Input:  Valid state σ0 and expanded state σ1
Output: Send a reduced state to the Predictor
  or output σ1 as a valid state
    If the Bounded(σ0, σ1) fails,
      send Reduce(σ0, ε, ε') to Predictor
    Else if On-Track(σ0, σ1) fails,
      send Reduce(σ0, ε, 1) to Predictor
    Else if Isolated(σ1) fails,
      send Reduce(σ0, 1, ε') to Predictor
    Else output σ1.
```

Figure 4: Corrector Module

4 DETAILS OF THE MODULES

In this section, we describe the details of the predictor and corrector modules. The predictor is designed to be optimistic and encourages larger disks and step sizes. The corrector reduces a valid state by reducing r_0 or δt_0. The adaptivity of our algorithm is based on a balance between these two opposing tendencies.

4.1 Predictor Module

For a given valid state σ_0, the predictor module suggests a new state σ_1 at time $\min\{t_0 + \delta t_0, 1\}$ by approximating the path $\alpha(t)$ with (an approximation to) the tangent line to the curve at t_0. In particular, we observe that, on small scales, the path $\alpha(t)$ starting at t_0 can be approximated by the tangent line to the curve $\alpha'(t_0)(t-t_0)+\alpha(t_0)$ at t_0. Since we do not know the root $\alpha(t_0)$, we approximate $\alpha(t_0)$ with m_0 and $\alpha'(t_0)$ with $G(m_0, t_0)$. This approximation is the point m_1 (known as the Euler step) in Program 5: Note that we define r_1 and δt_1 to be twice their previous values. These choices are optimistic and are designed to bias the algorithm using larger disks and taking larger steps. Hence σ_1 is called an "expanded state".

```
SUBROUTINE Predict(σ0)
Input:  Valid state σ0
Output: Valid state σ1
    m1 ← m0 + G(m0, t0) min {δt0, 1 − t0}
    r1 ← 2r0
    t1 ← min {1, t0 + δt0}
    δt1 ← 2δt0.
```

Figure 5: Predictor Subroutine

By a **transition**, we mean a pair of states written in the form "$\sigma_0 \to \sigma_1$" where σ_0 is valid and σ_1 (not necessarily valid) follows σ_0 in time. We next address the the tests and the Reduce function in the corrector module.

4.2 Bounded Test

Given σ_0 and an expanded state σ_1, we consider the convex hull of $\Delta(m_0, r_0, t_0)$ and $\Delta(m_1, r_1, t_1)$, denoted

$$T := \texttt{Chull}(\Delta(m_0, r_0, t_0), \Delta(m_1, r_1, t_1)).$$

So T is frustum in the three-dimensional space $\mathbb{C} \times [0, 1]$ illustrated in Figure 6. Since the function $G(x, t)$ is a rational function, its value is bounded if and only if the denominator never vanishes. The Bounded test is passed if the denominator $\frac{\partial H}{\partial x}(x, t)$ does not vanish in the frustum T.

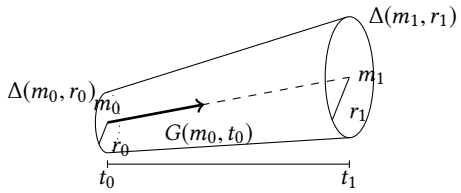

Figure 6: The frustum T is the convex hull of $\Delta(m_0, r_0, t_0)$ and $\Delta(m_1, r_1, t_1)$.

To verify that $\frac{\partial H}{\partial x}$ does not vanish on T, it is easier to show that it does not vanish on a cylinder $\widehat{T}$ containing T. We may define

$\widehat{T} := \Delta(m_0, R) \times [t_0, t_1]$ where R is chosen as

$$R := 3r_0 + |G(m_0, t_0)|\delta t_0. \tag{3}$$

Taylor series has linearly many terms in its degree. Suppose that we can find an $\mathcal{R}_x$ so that the following holds

$$\left(\frac{\partial H}{\partial x}\right)(\Delta(m_0, R) \times [t_0, t_1]) \subseteq \frac{\partial H}{\partial x}(m_0, t_0) + \Delta(0, \mathcal{R}_x)$$

for some positive real number $\mathcal{R}_x$ (see Eqn. (8) in Section 5 below). In this case, for all $(x, t) \in \Delta(m_0, R) \times [t_0, t_1]$,

$$\left|\frac{\partial H}{\partial x}(x, t)\right| \geq \left|\frac{\partial H}{\partial x}(m_0, t_0)\right| - \mathcal{R}_x. \tag{4}$$

If the right hand side of (4) is positive, $\frac{\partial H}{\partial x}$ will not vanish on T. For our correctness proof below, we need the slightly stronger condition that the right hand side of (4) is larger than $\frac{1}{4}\frac{\partial H}{\partial x}(m_0, t_0)$, which is equivalent to

$$\mathcal{R}_x \leq \frac{3}{4}\left|\frac{\partial H}{\partial x}(m_0, t_0)\right|. \tag{5}$$

In particular, the `Bounded` test is true if Inequality (5) holds.

4.3 On-Track Test

Suppose that the bounded condition is satisfied for transition $\sigma_0 \to \sigma_1$, so that G is bounded on T. We can then define

$$\max_{(x,t)\in T} |G(x, t) - G(m_0, t_0)|$$

to be the **relative variation** of G on T. Let $\mathcal{R}^*$ be any upper bound on the relative variation on T (see Eqn. (9) in Section 5 below), and define

$$\delta t^* := \frac{r_0}{\mathcal{R}^*}. \tag{6}$$

This quantity describes how much a solution path α starting within $\Delta(m_0, r_0)$ can bend away from $G(m_0, t_0)$. This is made precise in the following lemma:

Lemma 4.1 (On-Track Condition). *If $\delta t^* \geq \delta t_0$ and σ_0 is a valid state for $\alpha(t_0)$, then α remains within T.*

Proof. Let $m(t) := m_0 + G(m_0, t_0)(t - t_0)$ be the path connecting the centers of $\Delta(m_0, r_0) \times \{t_0\}$ and $\Delta(m_1, r_1) \times \{t_1\}$ and $\gamma(t) := \alpha(t) - m(t)$. Observe that by our choice of R, for all $t \in [t_0, t_1)$, $T \cap \{t\}$ is contained within the relative interior of $\widehat{T} \cap \{t\}$. Since $\Delta(m_0, r_0)$ is valid for $\alpha(t_0)$, we know that $|\gamma(t_0)| \leq r_0$. Suppose, for contradiction, that α leaves T. By our observation, there is some time $t' \in (t_0, t_1)$ so that $\alpha(t') \notin T$ but $\alpha(t) \in \widehat{T}$ for all $t \in [t_0, t']$. We observe that $|\gamma(t')| > r_0\left(1 + \frac{t'-t_0}{\delta t_0}\right)$. Therefore, by the reverse triangle inequality, it follows that

$$\frac{|\gamma(t') - \gamma(t_0)|}{t' - t_0} \geq \frac{|\gamma(t')| - |\gamma(t)|}{t' - t_0} > \frac{r_0}{\delta t_0}. \tag{7}$$

However, by the mean value theorem for paths, the quantities in Inequality (7) are bounded above by $|\gamma'(t'')| \leq \mathcal{R}^* = \frac{r_0}{\delta t^*}$ for some $t'' \in (t_0, t')$. This contradicts the given inequality. **Q.E.D.**

Therefore, we define the `On-Track` test to be true if the inequality in Lemma 4.1 holds.

4.4 Isolated Test

The Isolated Test verifies that $\Delta(m_1, r_1)$ is a well-isolator. If the transition $\sigma_0 \to \sigma_1$ satisfies both the `Bounded` and `On-Track` tests, then the solution path α must pass through $\Delta(m_1, r_1)$. Therefore $\Delta(m_1, r_1)$ is a well-isolator if and only if $3\Delta(m_1, r_1)$ contains exactly one root of $H(x, t_1)$. We certify this by using the soft-variant of the Graeffe-Pellet test from [4]. In particular, if $\widetilde{T}_1^G(3\Delta(m_1, r_1), H(x, t_1))$ succeeds, then $3\Delta(m_1, r_1)$ contains exactly one root of $H(x, t_1)$. Moreover, by [4, Lemma 4], if there are no roots other than $\alpha(t_1)$ in $4\Delta(m_1, r_1)$, then the Graeffe-Pellet test succeeds.

Thus, we define the `Isolated` test to be true if the Graeffe-Pellet test $\widetilde{T}_1^G(3\Delta(m_1, r_1), H(x, t_1))$ succeeds.

4.5 Corrector

The correctors within the corrector module are based on the `Reduce` function, where $\texttt{Reduce}(\sigma_0, \varepsilon, \varepsilon')$ reduces the step size by a factor of ε and replaces the isolating disk $\Delta(m_0, r_0)$ with a new disk which is a subset of $\Delta(m_0, r_0)$ of radius at most $\varepsilon' r_0$. Reducing the step size is straight-forward since we may replace δt_0 with $\varepsilon \delta t_0$. We now focus on the reduction of the radius, by following the approach in [4, Algorithm 8].

Since σ_0 is known to be isolating, it is enough to cover $\Delta(m_0, r_0)$ by smaller disks and confirm that one of these disks is isolating. In particular, let B be the circumscribing square of $\Delta(m_0, r_0)$, treating $\mathbb{C}$ as $\mathbb{R}^2$. We then split B into four equal boxes $B_1, \ldots, B_4$ by bisecting each of the sides of B. For each B_i, let m_i be the midpoint of B_i and consider $\Delta\left(m_i, \frac{3}{4}r_0\right)$. We show that for one of these disks the Graeffe-Pellet test succeeds as follows, see also Figure 7: The disk $\Delta\left(m_i, \frac{\sqrt{2}}{2}r_0\right)$ contains the box B_i and $\Delta\left(m_i, \frac{3}{2}r_0\right)$ is contained within $3\Delta(m_0, r_0)$. If $\alpha(t_0) \in B_i$, then the Graeffe-Pellet test succeeds on $\Delta\left(m_i, \frac{3}{4}r_0\right)$ by [4, Lemma 4]. Moreover, the disk $3\Delta\left(m_i, \frac{3}{4}r_0\right)$ is contained in $3\Delta(m_0, r_0)$, so $\Delta\left(m_i, \frac{3}{4}r_0\right)$ is valid. This construction is made explicit in the `Bisect` subroutine. We summarize this argument in the following lemma:

Lemma 4.2. *The* `Bisect` *subroutine returns a valid state for $\alpha(t)$.*

```
Subroutine Bisect(Δ(m_0, r_0))
Input:  Well-isolator Δ(m_0, r_0)
Output: Well-isolator Δ(m_0', r_0') of radius 3r_0/4.
  Let B be circumscribing square of Δ(m_0, r_0)
    Split B into 4 congruent subboxes B_1, ..., B_4
  For j = 1 to 4:
    Let m_j be the center of B_j
    If T̃_1^G(Δ(m_j, 3/4 r_0, H(x, t_0)) succeeds
      m_0' ← m_j and r_0' ← 3/4 r_0
      Return Δ(m_0', r_0').
```

4.6 Correctness of the Algorithm

The main issue of correctness is halting: this is proved in two parts. Here, we prove that the inner loop halts (i.e., `Update` halts). The

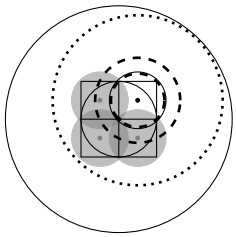

Figure 7: Bisection Geometry. The upper right disk contains the northeast corner, passes the Graeffe-Pellet test and is well-isolated.

halting of the outer loop (i.e., the main tracker) is more intricate, and awaits some additional development in Section 5.

In Program 8, we merged the Predictor and Corrector Modules into an `Update` Subroutine; note that we "inverted" the linear structure of the Corrector Module (Fig. 4) into an equivalent triply-nested If-Then-Else statements to fit into standard programming constructs. The Else-parts of these statements are the 3 cases of the corrector.

```
SUBROUTINE Update(σ0)
Input:  Valid state σ0 = (m0, r0, t0, δt0)
Output: Valid state σ1 = (m1, r1, t1, δt1), (t0 < t1 ≤ 1)
  Repeat:
    ▷ Prediction:
    σ1 ← Predict(σ0)                    ◁ (see Program 5)
    ▷ Bounded Test:
    Compute R                           ◁ (see Eqn.(3))
    If (Inequality (5) holds) then      ◁ (see Eqn.(8))
      ▷ On-track Test:
      Compute R* and δt*                ◁ (see Eqs.(9) and (6))
      If (δt* < δt0) then
        ▷ Isolated Test:
        If (T̃1^G(3Δ(m1, r1), H(x, t1)) succeeds) then
          Return σ1                     ◁ (σ1 is valid)
        Else ▷ Corrector3:
          Δ(m1, r1) ← Bisect(Δ(m0, r0))
          σ0 ← (m1, r1, t0, δt0)
      Else ▷ Corrector2:
        σ0 ← (m0, r0, t0, (1/2)δt0)
    Else ▷ Corrector1:
      Δ(m1, r1) ← Bisect(Δ(m0, r0)),
      σ0 ← (m1, r1, t0, (1/2)δt0)
```

Figure 8: Update Subroutine

LEMMA 4.3. *Suppose that $\mathcal{R}_x$ converges to 0 as r_0 and δt_0 decrease to zero. Then, the* `Update` *subroutine terminates and is correct.*

Proof. By following the three tests, we see that when the update step terminates, the result is a valid step. It remains to show that the algorithm will terminate. If the subroutine does not terminate, then at least one of the following three steps steps occurs infinitely often: Inequality (5) fails, $\delta t^* < \delta t_0$, or the Graeffe-Pellet test.

Inequality (5) does not fail infinitely often. If it is does, both r_0 and δt_0 are approaching 0. Thus $\mathcal{R}_x$ is approaching zero. Since $\frac{\partial H}{\partial x}$ is never zero on α, there exists a closed tubular neighborhood around α where $\frac{\partial H}{\partial x}$ is never zero. We restrict our attention to within this tube. By compactness, $\frac{\partial H}{\partial x}$ is bounded away from zero on this tubular neighborhood, so, for all r_0 and δt_0 sufficiently small, $\mathcal{R}_x$ is sufficiently small so that Inequality (5) succeeds.

We show that the Graeffe-Pellet test does not fail infinitely often. Since the solution paths are disjoint, there is a global lower bound (i.e., on $[t_0, t_1]$) on the distance between the solution path α and any other solution path. Therefore, when r_0 is sufficiently small, the Graeffe-Pellet test will always succeed, by [4, Lemma 4], since $3\Delta(m_1, r_1)$ will be sufficiently far away from other roots of $H(x, t_1)$.

The test $\delta t^* < \delta t_0$ does not fail infinitely either: when Inequality (5) succeeds, δt^* is bounded away from zero, so δt_0 will need to be reduced only finitely many times for $\delta t^* \geq \delta t_0$ to succeed. **Q.E.D.**

The preconditions of this lemma is easily satisfied by the instantiations of $\mathcal{R}_x$ in Section 5.

5 IMPLEMENTATION DETAILS

In this section, we present the details of the computations our algorithms' modules. We base our calculations on the standard centered form in interval arithmetic. Our computational model assumes that numerical values are approximated by dyadic number (BigFloats). During the bisection process, the representation of the disc are exact, and remain so because the constants of subdivision are carefully chosen [3, 4].

A formula for $\mathcal{R}_x$ from Section 4.2 can be constructed from the standard centered form as follows: We define $Q(x,t) = \frac{\partial H}{\partial x}(x,t)$ and $Q_i(x,t)$ to be the i-th Taylor coefficient of $Q(x,t)$ with respect to x, i.e, $Q_i(x,t) = \frac{\partial^{i+1} H}{i!\partial x^{i+1}}(x,t)$. Similarly, we define $P(x) = \frac{\partial H}{\partial t}(x,t)$ and P_i to be the i-th Taylor coefficient of $P(x)$ with respect to t, i.e., $P_i(x) = \frac{\partial^{i+1} H}{i!\partial t \partial x^{i+1}}(x)$. We can then compute the radius of the standard centered form on $\Delta(m_0, R) \times (t_1 - t_0)[-1, 1]$ as:

$$\mathcal{R}_x = \sum_{i=1}^{n-1} |Q_i(m_0, t_0)| R^i + \sum_{i=0}^{n-1} (i+1)|P_{i+1}(m_0)| R^i \delta t_0. \tag{8}$$

We fix a closed tubular neighborhood of α, and we observe that each of these functions can be bounded from above within this neighborhood. Moreover, as r_0 and δt_0 approach zero, R also approaches zero. Therefore, when r_0, δt_0, and R are small enough to be within the fixed tubular neighborhood of α, we see that $\mathcal{R}_x$ approaches zero as R and δt_0 approach 0.

In a similar manner, we can derive a formula for $\mathcal{R}^*$. We observe that since $G(x,t) = -\frac{P(x,t)}{Q(x,t)}$, for any x and t,

$$G(x,t) - G(m_0, t_0) = -\frac{P(x) + G(m_0, t_0) Q(x,t)}{Q(x,t)}.$$

By maximizing the absolute value of the numerator and minimizing the absolute value of the denominator of the right-hand-side over $\Delta(m_0, R) \times (t_1 - t_0)[-1, 1]$, we have an upper bound on the relative variation on G. We have seen a lower bound for the denominator in Inequality (4). On the other hand, we observe that at $x = m_0$

and $t = t_0$, the numerator vanishes, so an upper bound on the numerator is the width of the standard centered form applied to the numerator. In other words,

$$\mathcal{R}_t = \sum_{i=1}^{n} |P_i(m_0) + G(m_0, t_0)Q_i(m_0, t_0)|\, R^i + |G(m_0, t_0)| \sum_{i=0}^{n-1} (i+1)|P_{i+1}(m_0)|R^i \delta t_0.$$

Therefore,

$$\mathcal{R}^* = \mathcal{R}_t \Big/ \left(\left| \frac{\partial H}{\partial x}(m_0, t_0) \right| - \mathcal{R}_x \right) \tag{9}$$

is an upper bound on the relative variation of G on T.

5.1 Improved Update Strategy

The `Update` routine as given above implements what we will call the basic strategy, designated S0. We now describe an improved strategy, designated S1. The motivations for considering S1 are two-fold: First, it is unclear how to prove its termination. Moreover, our experiments (see Table 2) show that it may be very slow). Experimentally, we found that when the `Update` subroutine calls Corrector 1, it sometimes needs to only reduce one of r_0 or δt_0. However, Corrector 1 reduces both parameters. We therefore decouple the radius r_0 from the speed δt_0, by first substituting the formula for R into the formula for $\mathcal{R}_x$, see Eqns. (3) and (8). Then, by observing that

$$R^i = (3r_0 + |G(m_0, t_0)|\delta t_0)^i \le (6r_0)^i + (2|G(m_0, t_0)\delta t_0)^i,$$

we separate the r_0 terms from the δt_0 terms and note that δt_0 is at most 2 (otherwise, the previous step terminated). Therefore, if

$$\sum_{i=1}^{n-1} (|Q_i(m_0, t_0)| + 2(i+1)|P_{i+1}(m_0)|)6^i r_0^i \le \frac{3}{8} \left| \frac{\partial H}{\partial x}(m_0, t_0) \right| \tag{10}$$

is false, we perform the `Bisect` subroutine. Alternatively, if

$$\sum_{i=1}^{n-1} (|Q_i(m_0, t_0)| + (i+1)|P_{i+1}(m_0)|)2^{i+1}|G(m_0, t_0)|^{i+1}\delta t_0^{i+1} + |P_1(m_0)|\delta t_0 \le \frac{3}{8} \left| \frac{\partial H}{\partial x}(m_0, t_0) \right| \tag{11}$$

is false, we reduce δt_0 by half. We observe if both of these conditions are true, then Inequality (5) succeeds. Our experiments show that the refined algorithm is much more efficient than the original one.

5.2 Termination of the Algorithm

Theorem 5.1. *The main loop of the homotopy algorithm with the improved strategy terminates.*

Proof. Since the `Update` subroutine always terminates, by Lemma 4.3, it is enough to compute a lower bound on the step size δt taken along a path. For the solution path α, let $\rho : [0, 1] \to \mathbb{R}$ be the function where $\rho(t_0)$ is the minimum distance from $\alpha(t_0)$ to a root of $\frac{\partial H}{\partial x}(x, t_0)$.

We begin by proving that whenever Inequality (5) succeeds, the distance between $\alpha(t_0)$ and m_0 is at most $\frac{1}{2}\rho(t_0)$ as follows: When Inequality (5) succeeds, there are no roots of $\frac{\partial H}{\partial x}(x, t_0)$ contained in $\Delta(m_0, R)$ with radius $R \ge 3r_0$. Since the distance between m_0 and $\alpha(t_0)$ is at most r_0, we see that there are no roots of $\frac{\partial H}{\partial x}(x, t_0)$ within $\Delta(\alpha(t_0), 2r_0) \subseteq \Delta(m_0, R)$. Therefore, $2r_0 \le \rho(t_0)$.

For the rest of this proof, we restrict our attention to the closed tubular neighborhood $\mathcal{T} = \bigcup_{t\in[0,1]} \Delta\left(\alpha(t), \frac{1}{2}\rho(t), t\right)$ of α. We observe that we can restrict our attention to this compact set. Since $\frac{\partial H}{\partial x}$ does not vanish on $\mathcal{T}$, there exists a positive lower bound L on $\frac{\partial H}{\partial x}$ on T. By Inequality (5), it follows that the denominator of Eqn. (9) is bounded from below by $\frac{1}{4}L$. On the other hand, on the compact set $\mathcal{T}$, there is a finite upper bound for $|P_i(m)|$, $|G(m, t)|$ and $|Q_i(m, t)|$ for $(m, t) \in \mathcal{T}$.

We begin by finding a lower bound on r_0. We first observe that in Inequality (10), the term on the right can be bounded from below and the coefficients on the left can be bounded from above. Let $\hat{r}$ be the largest value for r_0 so that the corresponding inequality with the upper and lower bounds is true. Therefore, Inequality (10) succeeds whenever $r_0 \le \hat{r}$.

Additionally, let $\tau : [0, 1] \to \mathbb{R}$ be the function where $\tau(t_0)$ is the minimum distance from $\alpha(t_0)$ to a distinct root of $H(x, t_0)$. Following the argument above since $\Delta(m_0, r_0)$ is well-isolating, the distance between m_0 and $\alpha(t_0)$ is at least $\frac{1}{8}\tau(t_0)$ by [4, Lemma 4]. By compactness, this is bounded from below at some time $\hat{t}$. Therefore, the Graeffe-Pellet test succeeds whenever $r_0 \le \frac{1}{8}\tau(\hat{t})$. Since at every step the radius is decreased by at most a factor of $\frac{3}{4}$, we know that $\frac{3}{4}\min\left\{\hat{r}, \frac{1}{8}\tau(\hat{t})\right\}$ is a lower bound for r_0.

Next, we find a lower bound on δt_0. Substituting the bounds for $|P_i(m)|$, $|G(m, t)|$ and $|Q_i(m, t)|$ into the formula for $\mathcal{R}_t$, we see that $\mathcal{R}_t$ is bounded from above, and let U be such an upper bound. Therefore, the largest $\mathcal{R}^*$ that will ever be throughout the homotopy is $4\frac{U}{L}$. Since, in addition, we have a lower bound on r_0, we can construct a positive lower bound $\widehat{\delta t^*}$ on δt^* throughout the algorithm.

Applying the argument from Inequality (10) from above to Inequality (11), we conclude that we can find $\widehat{\delta t}$, the largest value for δt_0 for which the corresponding inequality with the upper and lower bounds is true. Therefore, Inequality (11) succeeds whenever $\delta t_0 \le \widehat{\delta t}$. Since at every step δt_0 is decreased by at most a factor of $\frac{1}{2}$, we know that $\frac{1}{2}\min\left\{\widehat{\delta t^*}, \widehat{\delta t}\right\}$ is a lower bound for δt_0 throughout the algorithm. **Q.E.D.**

6 EXPERIMENTAL RESULTS

In Table 2 we report on some experiments on a suite of polynomials (mostly taken from the MPSolve database [8, 9]). Our software can be downloaded from our SVN repository https://cs.nyu.edu/exact/ under `progs/homotopyPath`. The implementation is in `C++` using the `Boost mpfi` and `mpfr` libraries. The experiments were carried out on a PC with an Intel Core i5-4210U CPU at 1.70 GHz and 4GB RAM. We use the Cygwin platform on a Windows 7 OS. The timings below could be greatly improved on a native Windows implementation, or a more powerful CPU. But our main focus in the table is statistics on the Number of Steps (also Step Sizes and Radii), and these are machine-independent. These tables are reproducible as targets of Makefiles in our download folders.

For each polynomial in our test suite, we run our tracker and count the number of successful or failing paths, and for the successful paths obtain the Number of Time Steps (T), Step Size (δt), Radius

Table 2: Experimental Results

PolyID	Strategy (S0 or S1)	No.Paths (succ/fail)	No.Steps (T) (max/min/avg)	Step Size (δt) (max/min/avg)	Radius (r) (max/min)	Time(certified) (max/min/avg) secs.	Time(non-certified) (max/min/avg) secs.
wilk15	S1	15/0	1992/154/790.3	0.125/1.8e-15/0.0013	0.53/0.0083	1.06/0.078/6.93	0.125/0.015/0.77
	S0	0/15					
mign20	S1	20/0	929/174/272.2	0.0625/8.97e-44/0.0037	0.1/6.62e-25	40.8/0.12/81.2	11.9/0.015/23.6
	S0	0/20					
chrma20	S1	20/0	2705/123/574.7	0.125/1.16e-10/0.0017	0.1/0.0031	2.01/0.078/8.7	0.218/0/0.92
	S0	7/13	6.56e+04/92/1.48e+04	0.063/7.45e-09/6.76e-05	0.025/0.0063	59.4/0.078/93.6	5.35/0/8.47
chrma22	S1	21/0	2683/26/555	0.13/5.82e-11/0.0018	0.095/0.0029	2.2/0.015/9.5	0.23/0/1.01
	S0	7/14	3.75e+04/17/1.03e+04	0.25/1.19e-07/9.71e-05	0.024/0.0059	50.2/0.015/88.6	4.29/0/7.58
chrmc11	S1	11/0	1020/49/279.6	0.125/1.91e-06/0.0036	0.091/0.0057	0.36/0.031/1.12	0.062/0/0.219
	S0	10/1	7.24+04/41/2.054e+04	0.063/3.81e-06/4.87e-05	0.18/0.011	41.8/0.016/101	6.58/0/15.5
kam3_1	S1	9/0	1095/853/975.3	0.13/2.58e-26/0.0010	3.56/7.89e-16	8.71/0.56/36.6	
	S0	0/9					
cheby20	S1	20/0	860/34/239.4	0.125/1.86e-09/0.0042	0.025/0.00078	0.83/0.031/4.66	0.078/0/0.374
	S0	2/18	505/505/505	0.0039/0.00195/0.0019	0.0063/0.0031	0.436/0.358/0.794	0.031/0.031/0.062
cheby40	S1	40/0	2177/49/482.4	0.13/5.55e-17/0.0021	0.013/0.00021	11.9/1.15/107	0.312/0/2.7
	S0	2/38	3820/3819/3820	0.0019/0.00024/0.00026	0.0032/0.00052	11.7/11.4/23.1	

(r) as well as computing time (τ). Since there are multiple paths, each of these statistics are represented by three numbers: max/min/avg. The impact of certification on running time is indicated by another statistic, namely the computing time for a non-certified version (τ_{nc})

We implemented a certified and non-certified version of our Well-Isolated Tracker: the certified version uses interval arithmetic, but the non-certified version accepts the usual round-off errors. Each polynomial in our test suite is run four times, using our basic strategy (S0) and improved stratey (S1), both in the certified as well as non-certified code. Our certified S1 runs are always 100% successful. We time out our S0 runs when T exceeds 100,000, counted as failure. This accounts for the blank lines for strategy 0 of the polynomials `wilk15, mign20, kam3_1`.

7 CONCLUSION

Our fundamental contribution is the introduction of well-isolated approximation as a new tool for certified path tracking in homotopy continuation methods. It is the first method that is independent of α-theory, and rather unexpectedly, also devoid of Newton iteration.

We develop an predictor-corrector strategy where the predictor's bias for taking larger time steps is restrained by the corrector in a calibrated way. Our basic strategy had to be refined in a non-trivial way in order to achieve our final algorithm for which it is possible to prove termination. A natural direction for future work is to extend this to multivariate root tracking.

REFERENCES

[1] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler. Bertini$^{\text{TM}}$: Software for numerical algebraic geometry. https://bertini.nd.edu, 2013.

[2] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler. *Numerically Solving Polynomial Systems with Bertini.* Software, Environments, Tools. Society for Industrial and Applied Mathematics, 2013.

[3] R. Becker, M. Sagraloff, V. Sharma, J. Xu, and C. Yap. Complexity analysis of root clustering for a complex polynomial. In *41st Int'l Symp. Symbolic and Alge. Comp.*, pages 71–78, 2016. July 19-22, Waterloo, Canada.

[4] R. Becker, M. Sagraloff, V. Sharma, and C. Yap. A near-optimal subdivision algorithm for complex root isolation based on Pellet test and Newton iteration. *J. Symbolic Computation*, Sept. 2017. https://doi.org/10.1016/j.jsc.2017.03.009. Also: arXiv:1509.06231v3 [cs.NA], 51 Pages.

[5] C. Beltrán and A. Leykin. Certified numerical homotopy tracking. *Experimental Mathematics*, 21(1):69–83, 2012.

[6] C. Beltrán and A. Leykin. Robust certified numerical homotopy tracking. *Foundations of Computational Mathematics. The Journal of the Society for the Foundations of Computational Mathematics*, 13(2):253–295, 2013.

[7] C. Beltrán and L. M. Pardo. On Smale's 17th problem: a probabilistic positive solution. *Foundations of Computational Mathematics. The Journal of the Society for the Foundations of Computational Mathematics*, 8(1):1–43, 2008.

[8] D. A. Bini and G. Fiorentino. Design, analysis, and implementation of a multiprecision , polynomial rootfinder. *Numerical Algorithms*, 23:127–173, 2000.

[9] D. A. Bini and L. Robol. Solving secular and polynomial equations: A multiprecision algorithm. *J. Computational and Applied Mathematics*, 272:276–292, 2014.

[10] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation.* Springer, 1998.

[11] P. Bürgisser and F. Cucker. On a problem posed by Steve Smale. *Annals of Mathematics. Second Series*, 174(3):1785–1836, 2011.

[12] J. D. Hauenstein, I. Haywood, and A. C. Liddell, Jr. An *a posteriori* certification algorithm for Newton homotopies. In *ISSAC 2014—Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, pages 248–255. ACM, New York, 2014.

[13] J. D. Hauenstein and A. C. Liddell, Jr. Certified predictor-corrector tracking for Newton homotopies. *Journal of Symbolic Computation*, 74:239–254, 2016.

[14] J. D. Hauenstein and F. Sottile. Algorithm 921: alphaCertified: certifying solutions to polynomial systems. *Association for Computing Machinery. Transactions on Mathematical Software*, 38(4):28:1–28:20, 2012.

[15] J. V. D. Hoeven. Reliable homotopy continuation. Technical Report hal-00589948, HAL archives-ouvertes, 2015.

[16] R. B. Kearfott and Z. Xing. An interval step control for continuation methods. *SIAM Journal on Numerical Analysis*, 31(3):892–914, 1994.

[17] M. Kim. *Hybrid interval marching/branch and bound method for parametrized nonlinear systems.* PhD thesis, University of Louisiana at Lafayette, 2004.

[18] 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.

[19] A. Leykin and R. Krone. NAG4M2 – numericalalgebraicgeometry. http://people.math.gatech.edu/~aleykin3/NAG4M2/, 2013.

[20] B. Martin, A. Goldsztejn, L. Granvilliers, and C. Jermann. Certified parallelotope continuation for one-manifolds. *SIAM Journal on Numerical Analysis*, 51(6):3373–3401, 2013.

[21] H. Ratschek and J. Rokne. *Computer Methods for the Range of Functions.* Horwood Publishing Limited, Chichester, West Sussex, UK, 1984.

[22] M. Shub and S. Smale. Complexity of Bezout's Theorem I: Geometric aspects. *J. of Amer. Math. Soc.*, 6(2):459–501, 1993.

[23] M. Shub and S. Smale. Complexity of Bezout's theorem V: Polynomial time. *Theoretical Computer Science*, 133(1):141–164, 1994.

[24] M. Shub and S. Smale. Complexity of Bezout's theorem. IV. probability of success and extensions. *SIAM Journal on Numerical Analysis*, 33(1):128–148, 1996.

[25] J. Verschelde. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software*, 25(2):251–276, 1999.