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 method in numerical root-finding. Recently, certified algorithms for homotopy continuation based on Smale's alpha-theory have been developed. This approach enforces 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.

CCS CONCEPTS

• **Computing methodologies** → **Hybrid symbolic-numeric methods**; *Equation and inequality solving algorithms*;

KEYWORDS

Homotopy Continuation; Certified Computation; Interval Arithmetic; Well-Isolated Roots

ACM Reference Format:

Juan Xu, Michael Burr, and Chee Yap. 2018. An Approach for Certifying Homotopy Continuation Paths: Univariate Case. In *ISSAC '18: 2018 ACM International Symposium on Symbolic and Algebraic Computation, July 16–19, 2018, New York, NY, USA*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3208976.3209010>

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 while the homotopy method chooses g as well as the homotopy $H(z, t)$ such that $H(z, 0) = g$ and $H(z, 1) = f$. We focus on linear homotopies of the form $H(z, t) = (1 - t)\gamma g(z) + t f(z)$ for some $\gamma \in \mathbb{C} \setminus \{0\}$. 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 [24], Hom4PS [18], and NAG4M2 [19], see also the references therein. Homotopy continuation methods (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 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 $\deg(f) = n$. Next, we describe the central problem of Phase (2).

A **solution path** for a 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 $\{\alpha^i : i = 1, \dots, n\}$ are non-singular and do not diverge to infinity. We restrict our attention to good homotopies in this paper.

In general, path tracking consists of the following iterative procedure: to track the i th path, assume we are given an approximation z_0^i to $\alpha^i(0)$. Starting with $(z_0^i, t_0^i = 0)$, 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 in Phase (2). Although the sequence $\{z_j^i : j \geq 0\}$ is explicit, the solution path $\alpha^i(t)$ is only implicit. The main problem is 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 to $\alpha^i(t_j)$ ” is imprecise. We now turn to ways to make these statements precise and effective.

The key idea, which also applies in multivariate setting, lies in defining what it means for $z_0 \in \mathbb{C}^m$ to identify a unique root z_* of $h : \mathbb{C}^m \rightarrow \mathbb{C}^m$. 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 because, until the current paper, the corrector steps of homotopy continuation use Newton iteration. 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

*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.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ISSAC '18, July 16–19, 2018, New York, NY, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5550-6/18/07...\$15.00

<https://doi.org/10.1145/3208976.3209010>

particular, one can effectively evaluate the function¹ 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 [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 use the recently developed soft Graeffe-Pellet tests [3, 4] to verify a well-isolator $\Delta(z, r)$.

The 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 our purposes. Consequently, we expect our approach takes fewer 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]. We run our tracker on $H_m(x, t) = (1 - t)\gamma(x^2 - 1) + t(x^2 - (1 + m))$ using $\gamma = 1$ or i (the $\gamma = 1$ case was used in the previous work). Overall, our well-isolated tracker uses 4 to 8 times fewer steps than the alpha-based tracker. Hauenstein and Liddle [13] use a specialized² homotopy to track a single root of f . Our performance is comparable to the ModifiedNewton tracker in their Table A.1. Their other trackers use about half as many steps.

α -theory has 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 allows 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 (including 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 an Algebraic Model (e.g., 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 method 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, 21–23], 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. We develop well-isolated approximations into a tool for correct path tracking. Our predictor uses classical 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 describing the interaction between the predictor and corrector. The main algorithm consists of an outer loop around this inner loop. Section 4 presents a slightly abstract view (the “geometric meaning”) of the Update Subroutine, and Section 5 instantiates these with explicit formulas necessary for implementation. Also in Section 5, we address 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 $H(x, 0) = \gamma g(x)$

¹ The α_h function is from Smale’s *alpha*-theory. It is defined for the polynomial h and should not be confused with the solution paths α^i .

² They then exploit the fact that $\frac{\partial H}{\partial t} = -f(x_1)$ is constant. We expect our performance to also improve if we exploit this special case.

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: ($\gamma = 1$)	12	18	20	21	32	36	36	39	41	44	48	48
# of Well-Isolated Steps: ($\gamma = i$)	25	30	33	36	46	49	50	53	53	56	60	62

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

and $H(x, 1) = f(x)$. Given a root $z \in \mathbb{C}$ of g , the **solution path** for z is a continuous function $\alpha : [0, 1] \rightarrow \mathbb{C}$ 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 we clarify the phrase “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 $\alpha(t)$ 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

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

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

Our algorithm proceeds by maintaining closed disks isolating the roots of $H(x, t_0)$ for $t_0 \in [0, 1]$. We denote disks as follows: $\Delta(m, r) \subseteq \mathbb{C}$ denotes the closed complex disk centered at $m \in \mathbb{C}$ of radius $r > 0$. Additionally, for any real $k > 0$, let $k\Delta(m, r) := \Delta(m, kr)$, and, 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 $h \in \mathbb{Q}[x]$, $\Delta(m, r)$ is **well-isolating** (for h) if both $\Delta(m, r)$ and $3\Delta(m, r)$ contain a unique root of h .

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:

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

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

Figure 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 $\Delta(m_0, r_0)$ for start polynomial g
Output: Well-isolator of the valid state σ at time $t = 1$.
 Initialize state $\sigma \leftarrow (m_0, r_0, 0, 2r_0)$
 While (the time of σ is less than 1)
 $\sigma \leftarrow \text{Update}(\sigma)$
 Return the well-isolator σ .

Figure 2: Main Algorithm for path tracking

Our main algorithm is the path tracker shown in Figure 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, called 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 $T \cap \mathbb{C} \times \{t_1\}$ is the disk $\Delta(m_1, r_1, t_1)$ corresponding to the expanded state. If all the tests pass, 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 step size δt_0 . The construction of T and the details of the correctors are discussed in Section 4.

Figure 3: Schematic of 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 σ_0 and expanded state σ_1
Output: Send a reduced state to the Predictor or output σ_1 as a valid state
 If the Bounded(σ_0, σ_1) fails,
 send Reduce($\sigma_0, \varepsilon, \varepsilon'$) to Predictor
 Else if On-Track(σ_0, σ_1) fails,
 send Reduce($\sigma_0, \varepsilon, 1$) to Predictor
 Else if Isolated(σ_1) fails,
 send Reduce($\sigma_0, 1, \varepsilon'$) to Predictor
 Else output σ_1 .

Figure 4: Corrector Module. $\varepsilon, \varepsilon' \in \{\frac{1}{2}, \frac{3}{4}\}$.

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 goals.

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 Figure 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 towards using larger disks and taking larger steps. Hence σ_1 is called an “expanded state”.

SUBROUTINE Predict(σ_0)
Input: Valid state σ_0
Output: Expanded state σ_1
 $m_1 \leftarrow m_0 + G(m_0, t_0) \min\{\delta t_0, 1 - t_0\}$
 $r_1 \leftarrow 2r_0$
 $t_1 \leftarrow \min\{1, t_0 + \delta t_0\}$
 $\delta t_1 \leftarrow 2\delta t_0$.

Figure 5: Predictor Subroutine

By a **transition**, we mean a pair of states written in the form “ $\sigma_0 \rightarrow \sigma_1$ ” where σ_0 is valid, σ_1 is not necessarily valid, but $t_0 < t_1$.

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

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

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 success of the Bounded implies 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 $\hat{T}$ containing T . We may define $\hat{T} := \Delta(m_0, R) \times [t_0, t_1]$ where R is chosen as

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

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 Equation (8) in Section 5 below). In this case, for all $(x, t) \in \Delta(m_0, R) \times [t_0, t_1]$,

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

If the right-hand-side of Inequality (4) is positive, $\frac{\partial H}{\partial x}$ does 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

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

In particular, if Inequality (5) holds then $\frac{\partial H}{\partial x}$ is bounded in T . A formula for $\mathcal{R}_x$ is given in Equation (8).

4.3 On-Track Test

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

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

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

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

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

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

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 $\sigma_0 \rightarrow \sigma_1$ 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 use the soft-variant of the Graeffe-Pellet test $\widetilde{T}_1^G(\Delta, h(x))$ in [4, Algorithm 5 in Section 3], which returns true only if $\Delta \subseteq \mathbb{C}$ has exactly one root. 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 corrector module is based on the Reduce function, where $\text{Reduce}(\sigma_0, \varepsilon, \varepsilon')$ 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 straightforward since we replace δt_0 with $\varepsilon \delta t_0$. We now reduce 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 $B_1, \dots, B_4$ 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 8: 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( $\Delta(m_0, r_0)$ )
Input: Well-isolator  $\Delta(m_0, r_0)$ 
Output: Well-isolator  $\Delta(m'_0, r'_0)$  of radius  $3r_0/4$ .
  Let  $B$  be circumscribing square of  $\Delta(m_0, r_0)$ 
  Split  $B$  into 4 congruent subboxes  $B_1, \dots, B_4$ 
  For  $j = 1$  to 4:
    Let  $m_j$  be the center of  $B_j$ 
    If  $\widetilde{T}_1^G(\Delta(m_j, \frac{3}{4}r_0), H(x, t_0))$  succeeds
       $m'_0 \leftarrow m_j$  and  $r'_0 \leftarrow \frac{3}{4}r_0$ 
  Return  $\Delta(m'_0, r'_0)$ .

```

Figure 7: The Bisection Subroutine

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

Figure 8: Bisection Geometry. The upper right disk contains the northeast corner, passes the Graeffe-Pellet test and is well-isolated.

In Figure 9, the Predictor and Corrector Modules are merged into an Update Subroutine. Note that the structure of the Corrector Module (Figure 4) is represented by triply-nested If-Then-Else statements. The Else-parts of these statements are the 3 cases of the corrector.

```

SUBROUTINE Update( $\sigma_0$ )
Input: Valid state  $\sigma_0 = (m_0, r_0, t_0, \delta t_0)$ 
Output: Valid state  $\sigma_1 = (m_1, r_1, t_1, \delta t_1)$ , ( $t_0 < t_1 \leq 1$ )
Repeat:
  ▶ Prediction:
   $\sigma_1 \leftarrow \text{Predict}(\sigma_0)$  ◀ (see Figure 5)
  ▶ Bounded Test:
  Compute  $R$  ◀ (see Equation(3))
  If (Inequality (5) holds) then ◀ (see Equation(8))
    ▶ On-track Test:
    Compute  $\mathcal{R}^*$  and  $\delta t^*$  ◀ (see Eqs.(9) and (6))
    If ( $\delta t^* \geq \delta t_0$ ) then
      ▶ Isolated Test:
      If ( $(\tilde{T}_1^G(3\Delta(m_1, r_1), H(x, t_1)))$  succeeds) then
        Return  $\sigma_1$  ◀ ( $\sigma_1$  is valid)
      Else ▶ Corrector3:
         $\Delta(m_1, r_1) \leftarrow \text{Bisect}(\Delta(m_0, r_0))$ 
         $\sigma_0 \leftarrow (m_1, r_1, t_0, \delta t_0)$ 
    Else ▶ Corrector2:
       $\sigma_0 \leftarrow (m_0, r_0, t_0, \frac{1}{2}\delta t_0)$ 
  Else ▶ Corrector1:
     $\Delta(m_1, r_1) \leftarrow \text{Bisect}(\Delta(m_0, r_0))$ ,
     $\sigma_0 \leftarrow (m_1, r_1, t_0, \frac{3}{4}\delta t_0)$ 

```

Figure 9: 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 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 $[0, 1]$) on the distance between the solution path α and any other solution path. Therefore, when r_0 is sufficiently small, the Graeffe-Pellet test always succeeds, by [4, Lemma 4], since $3\Delta(m_1, r_1)$ is 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 are satisfied by the instantiations of $\mathcal{R}_x$ in Section 5.

5 IMPLEMENTATION DETAILS

In this section, we present the details of the computations for 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 numbers (e.g., BigFloats). During the bisection process, the representation of the discs are exact, and remain so because the constants of subdivision are dyadic [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:

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

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 ,

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

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,

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

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

THEOREM 5.1. *The main loop of the homotopy algorithm terminates.*

Proof. By Lemma 4.3, the Update subroutine always terminates, thus it is enough to compute a lower bound on the step size δt_0 taken along a path. For the solution path α , let $\rho : [0, 1] \rightarrow \mathbb{R}$ be the function where $\rho(t)$ is the minimum x -distance from $\alpha(t)$ to a root of $\frac{\partial H}{\partial x}(x, t)$.

We begin by proving that whenever Inequality (5) succeeds, the distance between $\alpha(t_0)$ and m_0 is at most $\frac{1}{2}\rho(t_0)$: 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 $R \geq 3r_0$. 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, $2r_0 \leq \rho(t_0)$.

We now restrict our attention to the compact tubular neighborhood $\mathcal{T} = \bigcup_{t \in [0, 1]} \Delta(\alpha(t), \frac{1}{2}\rho(t), t)$ of α . 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 $\mathcal{T}$. By Inequality (5), it follows that the denominator of Equation (9) is bounded from below by $\frac{1}{4}L$. By the compactness of $\mathcal{T}$, $|P_i(m)|$, $|G(m, t)|$ and $|Q_i(m, t)|$ are bounded from above in $\mathcal{T}$. Since r_0 is bounded from above by the maximum value of $\frac{1}{2}\rho(t)$ and δt_0 is at most 2, $\mathcal{R}^*$ is bounded from above. Let $\bar{\mathcal{R}}^*$ be such an upper bound.

We first give a lower bound on the ratio $\frac{\delta t_0}{r_0}$ for all the states produced in the algorithm, and then show that r_0 and δt_0 are bounded from below. We observe that Predictor and Corrector1 do not change the ratio $\frac{\delta t_0}{r_0}$. Corrector2 decreases this ratio by a factor of $\frac{1}{2}$ when the On-track Test fails and Corrector3 increases this ratio by a factor of $\frac{4}{3} < 2$ when the Isolated Test fails.

By combining Equation (6) and Lemma 4.1, it follows that if $\frac{1}{\bar{\mathcal{R}}^*} \geq \frac{1}{\bar{\mathcal{R}}} \geq \frac{\delta t_0}{r_0}$, then the On-track test succeeds. Since Corrector2 decreases the ratio by a factor of $\frac{1}{2}$, it follows that Corrector2 never decreases $\frac{\delta t_0}{r_0}$ beyond $\frac{1}{2\bar{\mathcal{R}}^*}$. If the initial state is $\frac{\delta t_0}{r_0} = A > 0$ when $t = 0$, then, at every step, $\frac{\delta t_0}{r_0} \geq \min \left\{ \frac{1}{2\bar{\mathcal{R}}^*}, A \right\}$. Therefore, $\delta t_0 \geq r_0 \min \left\{ \frac{1}{2\bar{\mathcal{R}}^*}, A \right\}$. Once we show that r_0 is bounded away from zero, it follows that δt_0 is also bounded away from zero.

Since the Isolated test is only called after the On-track Test succeeds, we know that $\alpha(t_1) \in \Delta(m_1, r_1)$. By [4, Lemma 4], we know that the Isolated Test only fails when there is another root of $H(x, t_1)$ within $4\Delta(m_1, r_1)$. Let $\tau(t)$ be the distance between $\alpha(t)$ and the nearest distinct root of $H(x, t)$. When the Isolated Test fails, the distance between $\alpha(t_1)$ and the other root in $4\Delta(m_1, r_1)$ is at least $3r_1$. In other words, $3r_1 \leq \tau(t_1)$. Since r_1 is twice r_0 , $6r_0 \leq \tau(t_1)$. Since Corrector3 decreases the radius by a factor of $\frac{3}{4}$, Corrector3 can only decrease r_0 to $\frac{\tau(t_1)}{8}$. Since our homotopy is a good homotopy, τ is bounded from below away from zero. Let $\underline{\tau}$ be such a lower bound, then Corrector3 can never reduce r_0 below $\frac{\underline{\tau}}{8}$. Since $\delta t_0 \leq 2$, this implies that Corrector3 never increases the ratio $\frac{\delta t_0}{r_0}$ beyond $\max \left\{ \frac{16}{\underline{\tau}}, A \right\}$. In other words, $\delta t_0 \leq r_0 \max \left\{ \frac{16}{\underline{\tau}}, A \right\}$.

We now show that r_0 and, consequently, δt_0 are bounded away from zero. We observe that r_0 is only reduced when the Bounded or Isolated Tests fail. Above, we computed a lower bound for the Isolated Test, so we focus on the Bounded Test. We observe that the right-hand-side of Inequality (5) is bounded from below by $\frac{3}{4}L$. Moreover, we note that since δt_0 is bounded from above by a constant multiple of r_0 , it follows that R , as defined in Equation (3), is bounded from above on $\mathcal{T}$ by a constant multiple of r_0 . Substituting these bounds into Equation (8), we conclude that $\mathcal{R}_x$ is bounded from above on $\mathcal{T}$ by a constant multiple of r_0 . Hence, if r_0 is sufficiently small, then the Bounded Test succeeds. Since Corrector1 decreases r_0 by a factor of $\frac{3}{4}$ each time, Corrector1 never decreases r_0 below a positive bound. Therefore, r_0 is bounded from below throughout the entire algorithm, so δt_0 is also bounded from below throughout the algorithm and the claim holds. **Q.E.D.**

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. We consider S1 because 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

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

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

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

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

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

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. Using a similar argument as for Theorem 5.1, we conclude that

THEOREM 5.2. *The main loop of the homotopy algorithm with the improved strategy terminates.*

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 Linux OS, but our main focus in the table is the Number of Steps, Step Sizes, and Radii,

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 S0	15/0 0/15	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
mign20	S1 S0	20/0 0/20	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
chrma20	S1 S0	20/0 7/13	2705/123/574.7 6.56e+04/92/1.48e+04	0.125/1.16e-10/0.0017 0.063/7.45e-09/6.76e-05	0.1/0.0031 0.025/0.0063	2.01/0.078/8.7 59.4/0.078/93.6	0.218/0/0.92 5.35/0/8.47
chrma22	S1 S0	21/0 7/14	2683/26/555 3.75e+04/17/1.03e+04	0.13/5.82e-11/0.0018 0.25/1.19e-07/9.71e-05	0.095/0.0029 0.024/0.0059	2.2/0.015/9.5 50.2/0.015/88.6	0.23/0/1.01 4.29/0/7.58
chrnc11	S1 S0	11/0 10/1	1020/49/279.6 7.24+04/41/2.054e+04	0.125/1.91e-06/0.0036 0.063/3.81e-06/4.87e-05	0.091/0.0057 0.18/0.011	0.36/0.031/1.12 41.8/0.016/101	0.062/0/0.219 6.58/0/15.5
kam3_1	S1 S0	9/0 0/9	1095/853/975.3	0.13/2.58e-26/0.0010	3.56/7.89e-16	8.71/0.56/36.6	
cheby20	S1 S0	20/0 2/18	860/34/239.4 505/505/505	0.125/1.86e-09/0.0042 0.0039/0.00195/0.0019	0.025/0.00078 0.0063/0.0031	0.83/0.031/4.66 0.436/0.358/0.794	0.078/0/0.374 0.031/0.031/0.062
cheby40	S1 S0	40/0 2/38	2177/49/482.4 3820/3819/3820	0.13/5.55e-17/0.0021 0.0019/0.00024/0.00026	0.013/0.00021 0.0032/0.00052	11.9/1.15/107 11.7/11.4/23.1	0.312/0/2.7

which 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 (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 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 strategy (S1), both in the certified as well as non-certified code. Our certified S1 runs are 100% successful on our tests. We time-out when T exceeds 100,000. This accounts for the blank lines for strategy 0 in the table.

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 also Newton iteration. We leave the details of the extension to the multivariate case to future work.

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. BertiniTM: 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.*, 2016. ISSAC 2016, July 20-22, Wilfrid Laurier University, 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*, 86:51–96, May-June 2018.
- <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 multi-precision, polynomial rootfinder. *Numerical Algorithms*, 23:127–173, 2000.
- [9] D. A. Bini and L. Robol. Solving secular and polynomial equations: A multiprecision algorithm. *Journal of 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] M. Shub and S. Smale. Complexity of Bezout's Theorem I: Geometric aspects. *J. of Amer. Math. Soc.*, 6(2):459–501, 1993.
- [22] M. Shub and S. Smale. Complexity of Bezout's theorem V: Polynomial time. *Theoretical Computer Science*, 133(1):141–164, 1994.
- [23] 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.
- [24] 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.