An *a posteriori* certification algorithm for Newton homotopies

Jonathan D. Hauenstein[*]
Department of Mathematics
North Carolina State University
hauenstein@ncsu.edu

Ian Haywood[†]
Department of Mathematics
North Carolina State University
ithaywoo@ncsu.edu

Alan C. Liddell, Jr.[†]
Department of Mathematics
North Carolina State University
acliddel@ncsu.edu

ABSTRACT

A Newton homotopy is a homotopy that involves changing only the constant terms. They arise naturally, for example, when performing monodromy loops, moving end effectors of robots, and simply when trying to compute a solution to a square system of equations. Previous certified path tracking techniques have focused on using an *a priori* certified tracking scheme which means that the stepsize is constructed so that the result automatically satisfies some conditions. These schemes use pessimistic stepsizes that can be much smaller than those used by heuristic tracking methods. This article designs an *a posteriori* certification scheme that uses the result of a heuristic tracking scheme as input to produce a certificate that the path was indeed tracked correctly, e.g., no path jumpings occurred. By using an *a posteriori* approach, each step can be certified independently and thus certification of the path can be performed in parallel. Examples are presented demonstrating the efficiency of this *a posteriori* certification approach.

Categories and Subject Descriptors

G.4 [**Mathematical Software**]: Verification

General Terms

Algorithms, Verification

Keywords

homotopy continuation, Newton homotopy, alpha theory, certified tracking, alphaCertified

1. INTRODUCTION

Numerical solving methods for polynomial systems based on homotopy continuation, collectively called *numerical algebraic geometry*, are currently being used to solve systems arising in a wide-variety of areas in science and engineering. The fundamental computation for these methods is path tracking which numerically approximates a sequence of points on a one-real-dimensional smooth curve, i.e., a *path*. A *path jumping* occurs when the numerically computed points approximate a point on a different path. In order to computationally *prove* theorems using such methods, path jumpings need to certifiably be avoided. For example, in [21], monodromy groups for Schubert problems are computed via path tracking and the occurrence of even one path jumping could lead to computing the wrong group.

Path tracking related to polynomial systems has been implemented in various software packages, e.g., `Bertini` [3], `HOM4PS-2.0` [19], `PHCpack` [27], and `POLSYS_GLP` [26]. They rely upon using robust, but not certifiable, numerical methods that try to prevent path jumpings, for example, using adaptive precision methods [1, 2, 5]. A path tracker implemented in `NAG4M2` [20] certifiably prevents path jumpings using an *a priori* tracking scheme, which selects the length of the next step so that the resulting point numerically approximates a point on the same path. Robust heuristic path tracking methods are generally much faster than using a certified tracking scheme [7].

Our fundamental idea is to use the data collected using a robust heuristic path tracking scheme as input for an *a posteriori* certification scheme to prove that no path jumpings occurred. Therefore, one is using heuristic computations to generate information about the problem and also maintaining the certainty provided by a certification scheme. One main result is Theorem 4.2 which details our *a posteriori* scheme. Since proving that a path jumping did not occur during a given step along the path is independent of the other steps, such an *a posteriori* certification scheme is naturally parallelizable. Moreover, due to this independence, one can use exact rational arithmetic where the heights of the points are represented based on the local conditioning, i.e., enough to remain in the quadratic convergence basin of Newton's method, rather than on all the previous computations needed to obtain the points. These reduce the computational cost associated with using a certified scheme.

There are additional benefits with using an *a posteriori* scheme. For example, when computing a monodromy group, e.g., for Schubert problems [21], one generates elements of

[*]Supported in part by NSF grant DMS-1262428 and DARPA Young Faculty Award.

[†]Supported by NSF grant DMS-0943855.

ISSAC'14 July 23–25, 2014, Kobe, Japan.

the monodromy group by performing a monodromy loop. If the element associated with a monodromy loop was already contained in the group generated by the previously obtained elements, the element has added no new information about the monodromy group. That is, one can use heuristic path tracking schemes to quickly perform a monodromy loop and determine if the resulting element is already contained in the group generated by the previously computed elements or is a new generator of the monodromy group. If the element adds no new information, one simply moves on to another loop. If it is a new generator, then *a posteriori* certification is performed to prove that no path jumpings occurred during the heuristic path tracking thereby certifiably yielding a new generator of the monodromy group. That is, one saves computational costs by only certifying the monodromy loops which produced new generators of the monodromy group.

In this article, we focus on certifying paths defined by so-called Newton homotopies, which are homotopies of the form

$$H(x,t) = f(x) + t \cdot v \tag{1}$$

where $f : \mathbb{C}^n \to \mathbb{C}^n$ is a polynomial system and $v \in \mathbb{C}^n$ is a given vector. In addition to performing monodromy loops, such homotopies can be used to attempt to compute a solution of $f(x) = 0$ by taking $v = -f(x^*)$ for a given $x^* \in \mathbb{C}^n$ and tracking the path starting at $t = 1$ with $x = x^*$ defined by $H(x,t) \equiv 0$. In fact, if $f = 0$ has the Bézout number of distinct isolated solutions and $x^* \in \mathbb{C}^n$ is generic, then such an approach will yield a solution of $f = 0$.

The *a posteriori* certified approach developed below depends on α-theory (see [9, Ch. 8] for a general overview), which provides sufficient conditions that Newton's method will quadratically converge immediately starting at a given point. The certified tracking methods of [6, 7, 8, 10, 11, 23] depend on using α-theory to compute the size of the next step to prevent path jumpings whereas we will use it to certifiably determine if a path jumping occurred. The software `alphaCertified` [16] implements the necessary α-theoretic routines needed with [15] demonstrating its use to *a posteriori* prove results regarding endpoints of paths. The key distinction is that here we prove results about the whole path, not just the endpoint.

Since α-theory plays a fundamental role in our certification scheme, we provide the necessary background information in § 2. We summarize path tracking in § 3 and present our continuity test (Theorem 4.2) in § 4. The *a posteriori* certification scheme is described in § 5 which includes a lower bound on stepsize (Theorem 5.1). We briefly summarize the implementation of our certification scheme in the new software package `Cadenza` [13] in § 6 with § 7 providing examples demonstrating its practicality.

2. NEWTON'S METHOD

Let $g : \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial system and suppose that $x^* \in \mathcal{V}(g) = \{x \in \mathbb{C}^n \mid g(x) = 0\}$. If the Jacobian matrix $Dg(x^*)$ is invertible, then x^* is called a nonsingular solution of $g = 0$ and it is well-known that Newton's method quadratically converges to x^* starting at any point in a sufficiently small neighborhood of x^*. In this section, we summarize using α-theory to yield sufficient conditions on a point $x \in \mathbb{C}^n$ such that Newton's method will quadratically converge to some point in $\mathcal{V}(g)$. More details regarding α-theory are provided in [9, Ch. 8].

Consider the *Newton iteration* $N_g : \mathbb{C}^n \to \mathbb{C}^n$ defined by

$$N_g(x) = \begin{cases} x - Dg(x)^{-1}g(x) & \text{if } Dg(x) \text{ is invertible;} \\ x & \text{otherwise.} \end{cases}$$

Given $x_0 \in \mathbb{C}^n$, *Newton's method* defines the sequence

$$x_k = N_g(x_{k-1}) \text{ for } k \geq 1.$$

A point $x_0 \in \mathbb{C}^n$ is an *approximate solution* of $g = 0$ if there exists $\xi \in \mathcal{V}(g)$ such that

$$\|x_k - \xi\| \leq \left(\frac{1}{2}\right)^{2^k - 1} \|x_0 - \xi\|.$$

The point ξ is called the *associated solution* of x_0 and the sequence $\{x_k\}_{k\geq 0}$ quadratically converges immediately to ξ. The key to α-theory is to balance the size of the Newton step with the higher-order derivatives. In particular, if $Dg(x)$ is invertible, define

$$\begin{array}{rcl} \alpha(g,x) & = & \beta(g,x) \cdot \gamma(g,x), \\ \beta(g,x) & = & \|x - N_g(x)\| = \|Dg(x)^{-1}g(x)\|, \\ \gamma(g,x) & = & \sup\limits_{k\geq 2} \left\| \dfrac{Dg(x)^{-1}D^k g(x)}{k!} \right\|^{\frac{1}{k-1}}. \end{array}$$

In γ, $D^k g(x)$ is the k^{th} derivative of g [18, Ch. 5] with Proposition 3 of [24, § I-3] providing an approach for computing an upper bound on $\gamma(g,x)$.

When $Dg(x)$ is not invertible, we can naturally define $\beta(g,x) = 0$ and $\gamma(g,x) = \infty$. The indeterminate form $\alpha(g,x) = 0 \cdot \infty$ is defined based on $g(x)$. That is, if $g(x) = 0$, then $\alpha(g,x) = 0$, otherwise, $\alpha(g,x) = \infty$.

The following results from [9, Ch. 8] describe local information that can be obtained from α-theory.

THEOREM 2.1. *Let $g : \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial system and $x, y \in \mathbb{C}^n$ with $x \neq y$.*

1. *If $\|x - y\| \cdot \gamma(g,x) < 1 - \sqrt{2}/2$ and $Dg(x)^{-1}$ exists, then $Dg(y)^{-1}$ also exists.*
2. *If $g(x) = g(y) = 0$, then $4 \cdot \gamma(g,x) \cdot \|x - y\| \geq 5 - \sqrt{17}$.*

The following summarizes key results from [9, Ch. 8].

THEOREM 2.2. *Let $g : \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial system and $x, y \in \mathbb{C}^n$.*

1. *If x is an approximate solution of $g = 0$ with associated solution ξ, then $\|x - \xi\| \leq 2\beta(g,x)$.*
2. *If $4 \cdot \alpha(g,x) < 13 - 3\sqrt{17}$, then x is an approximate solution of $g = 0$.*
3. *If $u < 1 - \sqrt{2}/2$, $c = 2(\alpha(g,x) + u)/(1 - 4u + 2u^2)^2 < 1$, and $\alpha(g,x) + cu \leq u$, then N_g is a contraction map on*

   $$B = B(u/\gamma(g,x), x) = \{y \mid \|x - y\| \cdot \gamma(g,x) < u\}$$

   with contraction constant c. In particular, there is a unique point ξ in $B \cap \mathcal{V}(g)$ and Newton's method starting at each $y \in B$ converges to ξ.
4. *If $\alpha(g,x) < 0.03$ with $\|x - y\| \cdot \gamma(g,x) < 0.05$, then x and y are approximate solutions of $g = 0$ with the same associated solution.*

Example 2.3. For $g(x) = x^6 - 1$, it is easy to verify

$$\beta(g,x) = \frac{|x^6-1|}{6|x|^5}, \quad \gamma(g,x) = \frac{5}{2|x|}, \quad \alpha(g,x) = \frac{5|x^6-1|}{12|x|^6}$$

for any $x \neq 0$. In particular, $\alpha(g, 1.01) < 0.0242$ so that 1.01 is an approximate solution of $g = 0$ and, for any $y \in \mathbb{C}$ with

$$|y - 1.01| < \frac{0.05}{\gamma(g, 1.01)} = 0.0202,$$

y and 1.01 have the same associated solution, namely $\xi = 1$.

3. PATH TRACKING

The key to using homotopy continuation to numerically solve systems of polynomial equations is to numerically track a solution path. The following provides a basic summary of path tracking for a Newton homotopy (1) with [4, Ch. 2] providing a general overview.

Let $f : \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial system, $v \in \mathbb{C}^n$ be a vector, and $H(x,t)$ be the Newton homotopy defined by (1). Let $x^* \in \mathbb{C}^n$ such that $H(x^*, 1) = 0$, i.e., $f(x^*) = -v$. Suppose that there exists a path $x(t)$, continuous on $t \in [0,1]$ with $x(1) = x^*$, such that $H(x(t), t) \equiv 0$ and the Jacobian matrix $Df(x(t))$ is invertible for $0 \leq t \leq 1$. One numerically *tracks* the path $x(t)$ by computing $1 = t_1 > t_2 > \cdots > t_\ell = 0$ and $x_i \in \mathbb{C}^n$ such that x_i is an approximate solution of $H(x, t_i) = 0$ with associated solution $x(t_i)$. A *path jumping* occurs if there is a j such that the associated solution of x_j is not $x(t_j)$.

The path $x(t)$ satisfies the Davidenko differential equation

$$\dot{x} = -Jf(x)^{-1} \cdot v. \tag{2}$$

Path tracking can be performed via a predictor-corrector scheme using (2) to predict a new point along the path and the equation $H(x,t) = 0$ to correct closer to the path. If the prediction is poor, the correction step may produce an approximate solution that does not correspond to a point on the path $x(t)$ resulting in a path jumping. The following sections derive an *a posteriori* test for certifying that a path jumping did not occur, i.e., the path was properly tracked.

4. A RULE FOR PATH CONTINUITY

Certifying that a path was tracked correctly can be broken down into certifying each step. This section certifies one step while the following section presents an algorithm for *a posteriori* certification of path tracking. This certification test for a single step exploits the fact that, for Newton homotopies (1), one has $\gamma(H(\cdot,t), x) = \gamma(f, x)$.

The following presents our universal constants.

Lemma 4.1. *Let $g : \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial system and $x \in \mathbb{C}^n$ such that $Dg(x)^{-1}$ exists and x is an approximate solution of $g = 0$ with associated solution ξ. Define*

$$\alpha_0 = 0.04 \quad \textit{and} \quad u_0 = 0.079. \tag{3}$$

If $\alpha(g,x) \leq \alpha_0$ and

$$B = B(u_0/\gamma(g,x), x) = \{y \mid \|x - y\| \cdot \gamma(g,x) < u_0\}$$

then N_g is a contraction mapping on B. In particular, $\{\xi\} = B \cap \mathcal{V}(g)$ and Newton's method starting at each $y \in B$ converges to ξ.

Proof. This follows from Theorem 2.2(3) since

$$c_0 = 2(\alpha_0 + u_0)/(1 - 4u_0 + 2u_0^2)^2 < 1 \text{ and } \alpha_0 + c_0 u_0 < u_0.$$

□

Using these constants, the following certifies continuity.

Theorem 4.2. *Let $f : \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial system, $v \in \mathbb{C}^n$, $H(x,t) = f(x) + t \cdot v$, and $t_1, t_2 \in \mathbb{C}$ and $x_1, x_2 \in \mathbb{C}^n$ such that $Df(x_i)^{-1}$ exists and x_i is an approximate solution of $H(\cdot, t_i) = 0$ with associated solution ξ_i. If $j \in \{1,2\}$ with*

$$\alpha(H(\cdot,t_j), x_j) + |t_1 - t_2| \cdot \gamma(f, x_j) \cdot \|Df(x_j)^{-1} \cdot v\| \leq \alpha_0, \tag{4}$$

there is a continuous $z : [t_1, t_2] \to \mathbb{C}^n$ such that $Df(z(t))^{-1}$ exists and $H(z(t), t) \equiv 0$ on $[t_1, t_2]$ with $z(t_j) = \xi_j$. Additionally, suppose that $k \neq j$. Then, $z(t_k) = \xi_k$ if

$$\|x_1 - x_2\| \cdot \gamma(f, x_j) < u_0 \tag{5}$$

and $z(t_k) \neq \xi_k$ if

$$\|x_1 - x_2\| > 2\left(\beta(H(\cdot,t_k), x_1) + \beta(H(\cdot,t_k), x_2)\right). \tag{6}$$

Proof. Consider the ball

$$B = B(u_0/\gamma(f,x_j), x_j) = \{x \mid \|x - x_j\| \cdot \gamma(f, x_j) < u_0\}.$$

Since $u_0 < 1 - \sqrt{2}/2$ and $Df(x_j)^{-1}$ exists, Theorem 2.1(1) provides that $Df(x)^{-1}$ exists for all $x \in \overline{B}$.

Additionally, for any $t \in [t_1, t_2]$, we have

$$\begin{aligned} \beta(H(\cdot,t), x_j) &= \|Df(x_j)^{-1}(f(x_j) + t \cdot v)\| \\ &\leq \|Df(x_j)^{-1}(f(x_j) + t_j \cdot v)\| \\ &\quad + |t - t_j| \cdot \|Df(x_j)^{-1} \cdot v\| \\ &= \beta(H(\cdot,t_j), x_j) \\ &\quad + |t - t_j| \cdot \|Df(x_j)^{-1} \cdot v\|. \end{aligned}$$

Since γ is independent of t and $|t - t_j| \leq |t_1 - t_2|$, this yields

$$\begin{aligned} \alpha(H(\cdot,t), x_j) &\leq \alpha(H(\cdot,t_j), x_j) \\ &\quad + |t - t_j| \cdot \gamma(f, x_j) \cdot \|Df(x_j)^{-1} \cdot v\| \\ &\leq \alpha_0. \end{aligned}$$

Therefore, x_j is an approximate solution of $H(\cdot, t) = 0$ for all $t \in [t_1, t_2]$ with, say, associated solution $z(t)$. Clearly, $z(t_j) = \xi_j$ and, by Lemma 4.1, $\{z(t)\} = B \cap \mathcal{V}(H(\cdot, t))$.

The continuity of $z(t)$ follows from the Inverse Function Theorem. That is, we know a continuous solution path $\lambda(t)$ of $H(\cdot,t) = 0$ with $\lambda(t_j) = z(t_j) = \xi_j$ exists locally around t_j. Since Df is invertible on $\overline{B}$, we know that $\lambda(t)$ exists for all $t \in [t_1, t_2]$, and is equal to $z(t)$, provided that $\lambda(t) \in B$. Therefore, the only issue that could arise is if there exists $t^* \in [t_1, t_2]$ such that $\lambda(t^*) \in \partial B$. However, if this was the case, $8 \cdot u_0 < 5 - \sqrt{17}$ together with Theorem 2.1(2) yields $B \cap \mathcal{V}(H(\cdot, t^*)) = \emptyset$, which is a contradiction.

If (5) holds, then Lemma 4.1 provides $z(t_k) = \xi_k$.

If (6) holds, then

$$\begin{aligned} \|x_1 - x_2\| &\leq \|x_j - z(t_k)\| + \|z(t_k) - \xi_k\| + \|\xi_k - x_k\| \\ &\leq 2(\beta(H(\cdot,t_k), x_1) + \beta(H(\cdot,t_k), x_2)) \\ &\quad + \|z(t_k) - \xi_k\| \\ &< \|x_1 - x_2\| + \|z(t_k) - \xi_k\|. \end{aligned}$$

Hence, $\|z(t_k) - \xi_k\| > 0$ providing $z(t_k) \neq \xi_k$. □

This theorem provides an approach for certifiably determining whether a path jumping did or did not for a given step. Since this certification can be performed independently of the other steps, the certification procedure described in the following section is naturally parallelizable.

5. A CERTIFICATION PROCEDURE

Suppose that $H(x,t)$ is a Newton homotopy (1), $t_1 < t_2$, $z : [t_1,t_2] \to \mathbb{C}^n$ is a nonsingular solution path, and $x_i \in \mathbb{C}^n$ is an approximate solution of $H(x,t_i) = 0$ with associated solution $z(t_i)$. If (4) and (5) in Theorem 4.2 hold, then we have a certificate regarding the existence of the continuous nonsingular solution path $z(t)$.

We start by first assuming that (4) holds. Following the notation of Theorem 4.2, we know that x_j is an approximate solution of $H(\cdot,t_k) = 0$ with $\alpha(H(\cdot,t_k),x_j) \leq \alpha_0$ and associated solution $z(t_k)$. Thus, if (5) holds, it follows from Lemma 4.1 that both x_1 and x_2 have the same associated solution with respect to $H(\cdot,t_k)$. However, if (5) does not hold, one can use Algorithm 3 of [15] to certifiably determine they have the same associated solution, namely $z(t_k)$, and thus prove that a path jumping did not occur in $[t_1,t_2]$.

If (4) does not hold, the following describes two techniques for certifying continuity. The first is to replace x_j with $N^{\ell}_{H(\cdot,t_j)}(x_j)$ for some $\ell \geq 1$ which is also an approximate solution of $H(\cdot,t_j) = 0$ with associated solution $z(t_j)$. Since $z(t_j)$ is a nonsingular solution of $H(x,t_j) = 0$,

$$\{\gamma(f, N^{\ell}_{H(\cdot,t_j)}(x_j))\}_{\ell\geq 0} \text{ and } \{\|Df(N^{\ell}_{H(\cdot,t_j)}(x_j)))^{-1}\cdot v\|\}_{\ell\geq 0}$$

are bounded with

$$\{\beta(H(\cdot,t_j), N^{\ell}_{H(\cdot,t_j)}(x_j))\}_{\ell\geq 0}$$

quadratically converging to zero. Therefore, by replacing x_j with $N^{\ell}_{H(\cdot,t_j)}(x_j)$, the first term in (4) can be made arbitrarily small while the second term remains bounded. In particular, one can select $\ell \geq 1$ so that

$$\alpha(H(\cdot,t_j), N^{\ell}_{H(\cdot,t_j)}(x_j)) \leq \alpha_0/6.$$

The other technique to satisfy (4) is to reduce the second term by splitting the interval $[t_1,t_2]$ to decrease $|t_1 - t_2|$. Then, testing continuity on the interval $[t_1,t_2]$ is replaced by testing continuity across each subinterval, each of which can be performed independently. Our approach is to split $[t_1,t_2]$ into two subintervals at the midpoint $t_{3/2} = (t_1+t_2)/2$ where the corresponding point $x_{3/2} = (x_1+x_2)/2$ is also taken to be the midpoint. Since $x_{3/2}$ may not be an approximate solution, Newton's method is used to attempt to move the midpoint closer to the path. If this fails, one could use heuristic path tracking methods to obtain a numerical approximation along the path.

The following provides a lower bound on the length of the interval to guarantee that (4) must hold.

Theorem 5.1. *Let $f : \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial system, $v \in \mathbb{C}^n$, $H(x,t) = f(x)+t\cdot v$, and $t_1,t_2 \in \mathbb{C}$ and $x_1,x_2 \in \mathbb{C}^n$ such that $Df(x_i)^{-1}$ exists and x_i is an approximate solution of $H(\cdot,t_i) = 0$ with associated solution ξ_i. Suppose that $j \in \{1,2\}$ such that there is a continuous $z : [t_1,t_2] \to \mathbb{C}^n$ where $Df(z(t))^{-1}$ exists and $H(z(t),t) \equiv 0$ on $[t_1,t_2]$ with $z(t_j) = \xi_j$. Let $K = \lceil \log_2(33\cdot|t_1-t_2|\cdot\mathcal{M})\rceil$ where*

$$\mathcal{M} = \max_{t\in[t_1,t_2]} \gamma(f,z(t))\cdot\|Df(z(t))^{-1}\cdot v\| < \infty$$

and $\Delta s = (t_1-t_2)/2^K$. If $s_m = t_2 + m\cdot\Delta s$ and $y_m \in \mathbb{C}^n$ such that $\alpha(H(\cdot,s_m),y_m) \leq \alpha_0/6$ with associated solution $z(s_m)$ for $m = 0,\ldots,2^K$, then

$$\alpha(H(\cdot,s_m),y_m) + |\Delta s|\cdot\gamma(f,y_m)\cdot\|Df(y_m)^{-1}\cdot v\| \leq \alpha_0. \quad (7)$$

Proof. Since $\|y_m - z(t_m)\|\cdot\gamma(f,y_m) \leq 2\alpha(f,y_m) \leq \alpha_0/3$, Lemma 2 and Prop. 3 of [9, Ch. 8] yield

$$\gamma(f,y_m)\cdot\|Df(y_m)^{-1}\cdot v\| \leq 1.1\cdot\mathcal{M}.$$

Therefore, the left side of (7) is bounded above by

$$\alpha_0/6 + 1.1\cdot|t_2-t_1|\cdot\mathcal{M}/2^K \leq \alpha_0/6 + 1.1/33 \leq \alpha_0.$$

□

6. SOFTWARE

The *a posteriori* certification approach presented above has been implemented in the software `Cadenza` [13]. Since `Cadenza` relies upon `alphaCertified` [16] for computing β and bounding α and γ, the coefficients of the Newton homotopy $H(x,t)$ must lie in $\mathbb{Q}[\sqrt{-1}]$. Moreover, the computations can be performed using either exact rational arithmetic or arbitrary precision floating point arithmetic.

As with `alphaCertified`, the internal computations involving rational arithmetic are *certifiable*. Floating point arithmetic can be used to control the bit length growth of rational numbers, but the errors involving the internal computations are not fully controlled. Thus, we prefer to use floating point arithmetic to perform enough subdivisions of the path and related Newton iterations first and then simply verify the results using rational arithmetic. In this way, we avoid expression swell due to long computations using exact rational arithmetic. Since the intervals can be certified independently, *a posteriori* tracking does not use the output of one rational arithmetic computation as the input to another computation. Moreover, the robustness of Newton's method near solutions, e.g., see Items 3 and 4 of Theorem 2.2, allows one to potentially use rational numbers of smaller height, a fact which is exploited in [14, Sec. 5].

7. EXAMPLES

The following demonstrate the *a posteriori* certification approach. In § 7.1, we compare the number of intervals needed for *a posteriori* certification with the number of steps used by the certified tracking approach of [8]. Since this example is univariate, we also consider larger examples arising from kinematics in § 7.2 and § 7.3, and from discretizations of a system of PDEs in § 7.4. For these larger examples, we used `Bertini` [3] to heuristically track the paths producing the input for the certification procedure. All of the computational times reported below are for computations performed using serial processing.

7.1 Certified tracking comparison

For $m > -1$, consider $f(x) = x^2 - 1 - m$ and $v = m$ with

$$H(x,t) = f(x) + vt = x^2 - 1 - m + mt$$

being the Newton homotopy derived from [8, § 9.1]. In this univariate example, we use exact values, namely

$$\begin{array}{rcl} \alpha(H(\cdot,t),x) & = & |x^2-1-m+mt|/|2x|^2, \\ \beta(H(\cdot,t),x) & = & |x^2-1-m+mt|/|2x|, \\ \gamma(f,x) & = & 1/|2x| \end{array}$$

for all $t \in [0,1]$ and $x \neq 0$.

In our first test, we compare the number of intervals needed by our *a posteriori* certification procedure with the number of steps used by the certified tracking procedure of [8] using the values of m listed in [8, Table 3]. The path under consideration is

$$z(t) = \sqrt{1+m-mt} \tag{8}$$

where the number of initial points, with uniform stepsize, along the path depend on the value of m. For $m < 40$, we used $\Delta t = 1/4$ meaning that 5 points along the path were used. For $m < 1000$ and $m <$ 10,000 we took $\Delta = 1/6$ and $\Delta = 1/8$. For larger values of m, we took $\Delta = 1/10$.

Table 1 summarizes the results with Figure 1 comparing the two methods for $m = 30000$ with the local conditioning.

m	Number of *a priori* steps using [8]	Number of *a posteriori* certified intervals
10	184	51
20	217	67
30	237	78
40	250	82
50	260	88
60	269	92
70	276	96
80	282	99
90	288	103
100	292	105
1000	395	162
2000	426	180
3000	446	191
4000	457	197
5000	468	204
10000	499	220
20000	530	238
30000	547	250

Table 1: Comparison for selected values of m

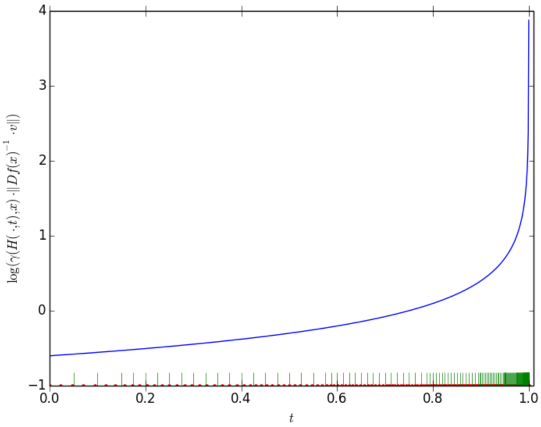

Figure 1: Log plot of local conditioning with respect to time for $m = 30000$ and location of steps via [8] in red with certified intervals marked in green.

In our second test, we consider both the performance of our *a posteriori* procedure as the endpoint at $t = 0$ approaches a singularity as well as the ability to identify a discontinuity. In particular, we take $m = -1 + 10^{-k}$ for $k = 1, \ldots, 10$ with uniform stepsize $\Delta t = 1/64$. For $[0, 13/64)$ and $(7/8, 1]$ we took points along $z(t)$ defined by (8) and, for $[13/64, 7/8]$, we took points along $-z(t)$. In each of our tests, `Cadenza` correctly located the discontinuities with Table 2 listing the number of intervals used by [8] and the *a posteriori* certification approach. This table shows the impact of the singularity was minimal for the selected values of k for the *a posteriori* method. Figure 2 compares these two methods for $k = 10$ near $t = 0$.

k	Number of *a priori* steps using [8]	Number of *a posteriori* certified intervals
1	176	64
2	287	68
3	390	70
4	492	71
5	593	71
6	695	71
7	798	71
8	901	71
9	1003	71
10	1108	71

Table 2: Summary for $m = -1 + 10^{-k}$

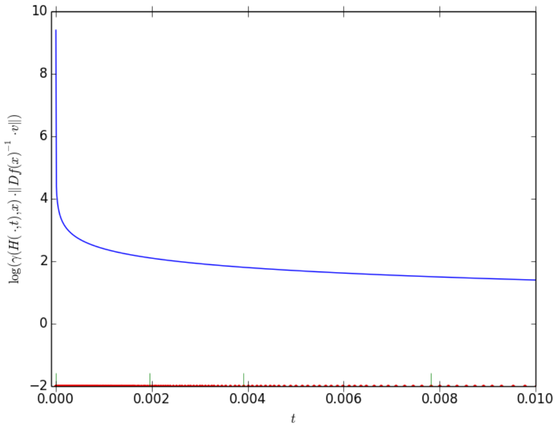

Figure 2: Log plot of local conditioning with respect to time for $k = 10$ near $t = 0$ and location of steps via [8] in red with certified intervals marked in green.

7.2 Moving a 6R linkage

Consider a polynomial from two linear and ten quadratic polynomials in 12 variables arising from the inverse kinematics of a general six-revolute, serial-link robot described in [31]. A Newton homotopy arises after fixing the parameters of a specific robot and considering the endpoint of this robot moving along a line segment. In particular, we use our *a posteriori* certification approach to prove that the specified robot can reach each point of a line segment using a smooth and continuous motion.

The twelve variables correspond to the coordinates of four vectors $\mathbf{z}_2, \ldots, \mathbf{z}_5 \in \mathbb{R}^3$ that will be vectors along the joint axes. The fixed parameters, whose values are listed in Table 3, are the vectors $\mathbf{z}_1, \mathbf{z}_6 \in \mathbb{R}^3$ for the fixed joint axes, the link lengths $a_1, \ldots, a_5 \in \mathbb{R}_{>0}$, and the link offset distances $d_2, \ldots, d_5 \in \mathbb{R}_{>0}$. The vectors $\mathbf{p} \in \mathbb{R}^3$, corresponding to the position of the endpoint, and $\mathbf{c} = (c_1, \ldots, c_5) \in [-1,1]^5$, corresponding to the cosines of the twist angles, will change with Table 3 showing the two ends of the segments, namely $\mathbf{p}^0$ and $\mathbf{p}^1$, and $\mathbf{c}^0$ and $\mathbf{c}^1$, respectively. With these selections, the resulting Newton homotopy can be formulated to have rational coefficients.

$\mathbf{z}_1 = (0,0,1)$	$\mathbf{z}_6 = (-27,-65,72)/100$
$(a_1,\ldots,a_5) = (2,10,4,5,7)$	$(d_2,\ldots,d_5) = (6,5,3,4)/10$
$\mathbf{c}^0 = (-51,-78,10,57,-23)/100$	$\mathbf{c}^1 = (-33,-98,64,97,-13)/100$
$\mathbf{p}^0 = (-3,5,4)$	$\mathbf{p}^1 = (2,3,1)$

Table 3: Parameters for 6R robot

Using dot product and cross product notation, the polynomial system consists of the following polynomials:

$$\begin{array}{ll} \mathbf{z}_i \cdot \mathbf{z}_i - 1 = 0, & i = 2, \ldots, 5 \\ \mathbf{z}_i \cdot \mathbf{z}_{i+1} - c_i = 0 & i = 1, \ldots, 5 \end{array}$$
$$a_1 \mathbf{z}_1 \times \mathbf{z}_2 + \sum_{i=2}^{5} (a_j \mathbf{z}_j \times \mathbf{z}_{j+1} + d_j \mathbf{z}_j) - \mathbf{p} = 0$$

with Newton homotopy arising by taking $\mathbf{p} = (1-t)\mathbf{p}^0 + t\mathbf{p}^1$ and $\mathbf{c} = (1-t)\mathbf{c}^0 + t\mathbf{c}^1$. In particular, this is a Newton homotopy for which each $\mathbf{z}_i$ is restricted to the unit sphere in $\mathbb{R}^3$.

We start, at $t = 1$, with the solution (rounded to 4 digits):

$$\begin{array}{l} \mathbf{z}_2 = (0.9439, -0.0082, -0.3300) \\ \mathbf{z}_3 = (-0.8677, -0.0850, 0.4897) \\ \mathbf{z}_4 = (-0.5247, 0.6911, 0.4971) \\ \mathbf{z}_5 = (-0.5926, 0.7584, 0.2716) \end{array}$$

and used Bertini to track the homotopy path. Using the default settings, it tracks the path taking 16 steps in roughly one-hundredth of a second. Using Cadenza, it took 2.2 seconds to certify the continuity of this path using 51 intervals.

7.3 A monodromy loop

We next consider performing a monodromy loop for a polynomial system consisting of 17 quadratics in 18 variables describing a 12-bar spherical linkage. The monodromy loops here will be setup to be used in coordination with decomposing an algebraic set into irreducible components [25]. Monodromy loops can also be used to compute Galois groups [21]. By using an *a posteriori* approach, as discussed in § 1, only the monodromy loops yielding new information need to be certified following the use of a fast heuristic approach.

The 12-bar linkage is obtained by locking the scissors of a collapsible cube having 12 scissor linkages presented in [30]. Such a linkage is presented in [29, Fig. 3] and we follow the setup presented there. In particular, we take the length of the side of the cube to be 2 and we fix the center of the cube at the origin. To remove the trivial rotation of the cube, we also fix two adjacent vertices at $\mathbf{P}_7 = (-1,1,-1)$ and $\mathbf{P}_8 = (-1,-1,-1)$. The coordinates of the remaining six vertices $\mathbf{P}_1, \ldots, \mathbf{P}_6$ form the 18 variables with the 17 polynomial constraints arising from maintaining relative distances:

$$\|\mathbf{P}_i - \mathbf{P}_j\|^2 - 4 = 0,$$
$$(i,j) \in \left\{ \begin{array}{c} (1,2),(3,4),(5,6),(1,5),(2,6),(3,7), \\ (4,8),(1,3),(2,4),(5,7),(6,8) \end{array} \right\}$$
$$\|\mathbf{P}_i\|^2 - 3 = 0, \quad i = 1, \ldots, 6.$$

Since this polynomial system f is underdetermined, each irreducible component (in $\mathbb{C}^{18}$) is positive-dimensional with the nondegenerate components being one-dimensional. The irreducible decomposition is presented in [12] which, in particular, shows that the solution set consists of 8 irreducible curves, six of degree 4 and two of degree 6. Let $\mathcal{C}$ be one of these degree 6 curves. (We note that since the degree 6 curves are complex conjugates of each other, the following computation holds for either of them.)

We will perform a monodromy loop on the 6 points of a witness point set for $\mathcal{C}$ as follows. For a constant $c \in \mathbb{C}$, we consider the linear polynomial $\mathcal{L}_c = \sum_{j=1}^{6} \mathbf{R}_i \cdot \mathbf{P}_i - c$ where

$$\begin{array}{lll} \mathbf{R}_1 = (1,4,1), & \mathbf{R}_2 = (-2,1,-5), & \mathbf{R}_3 = (5,4,-1) \\ \mathbf{R}_4 = (5,5,2), & \mathbf{R}_5 = (-2,1,-2), & \mathbf{R}_6 = (3,3,-1). \end{array}$$

We start with the witness point set $\mathcal{C} \cap \mathcal{V}(\mathcal{L}_1)$ consisting of 6 points. Then, we perform a loop by moving the constant term c from 1 to $5 - 12\sqrt{-1}$ to $-11 - 2\sqrt{-1}$ to 1 where each of the three is moved in a straight line. The result of this loop is three points return to themselves while the other three points form a 3-cycle. Using Bertini, the total time to track all of these 18 paths is under a second. The resulting data from Bertini was used as the input for Cadenza. The certification of these paths used between 23 and 123 intervals with times per path ranging from one second to nine seconds. The total certification time, using serial processing, for all 18 paths was 78 seconds.

7.4 Discretizing a system of PDEs

The last example is a collection of polynomial systems arising from the discretization of the Lotka-Volterra population model with diffusion [22, 28]:

$$\begin{array}{l} -\Delta u = u(1-v), \\ -\Delta v = v(u-1). \end{array}$$

The functions u and v are defined on the square $[0,1]^2$ with Δ being the Laplacian. For $N \geq 2$, we discretized the system similar to that performed in [17, § 9.4] and [4, § 17.1.2] via a central difference scheme at points $(x_i, y_j) = (\frac{i}{N+1}, \frac{j}{N+1})$ for $0 \leq i,j \leq N+1$ with variables $u_{i,j} \approx u(x_i, y_j)$ and $v_{i,j} \approx v(x_i, y_j)$. The Newton homotopy is constructed based on changing the boundary conditions. In particular, the homotopy consists of $2N^2$ quadratic equations and $8N$ linear equations which define the changing boundary conditions. For $1 \leq i,j \leq N$, the quadratic equations are:

$$\begin{array}{r} u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} + \dfrac{u_{i,j}(1-v_{i,j})}{(N+1)^2} = 0 \\ v_{i+1,j} + v_{i-1,j} + v_{i,j+1} + v_{i,j-1} - 4v_{i,j} + \dfrac{v_{i,j}(u_{i,j}-1)}{(N+1)^2} = 0 \end{array}$$

and linear equations are:

$$\begin{array}{r} u_{i,0} - \left(t + (1-t) \cdot \dfrac{i}{N+1}\right) = 0 \\ v_{i,0} - \left(t + (1-t) \cdot \dfrac{N+1-i}{N+1}\right) = 0 \\ u_{i,N+1} - \left(t + (1-t) \cdot b_u\left(\dfrac{i}{N+1}\right)\right) = 0 \\ v_{i,N+1} - \left(t + (1-t) \cdot b_v\left(\dfrac{i}{N+1}\right)\right) = 0 \\ u_{0,j} - t = 0 \\ v_{0,j} - 1 = 0 \\ u_{N+1,j} - 1 = 0 \\ v_{N+1,j} - t = 0 \end{array}$$

where $b_u(x) = \frac{120}{101}(x-x^3/6+x^5/120)$ and $b_v(x) = b_u(1-x)$. All of the starting boundary points, at $t = 1$, are 1 for which there is a unique positive real solution, namely $u \equiv 1$ and $v \equiv 1$, and this is the starting solution for the Newton homotopy. The ending boundary conditions are rational approximations of the boundary conditions in [4, § 17.1.2].

For each $2 \leq N \leq 8$, we used Bertini to track the corresponding path, which required at most 15 steps and completed in under a second. Using those points, we certified the path tracking using Cadenza. The results are summarized in Table 4 which reports the computational time using serial processing.

N	# variables	Number of *a posteriori* certified intervals	time (sec)
2	24	23	3
3	42	23	11
4	64	23	33
5	90	37	164
6	120	41	464
7	154	51	1014
8	192	56	2311

Table 4: Summary of *a posteriori* certification for the discretized Lotka-Volterra systems.

8. CONCLUSION

Heuristic path tracking schemes using high-order prediction approaches with adaptive precision robustly and efficiently track solution paths. For a Newton homotopy, we use the results of this heuristic path tracking as input for an *a posteriori* procedure that certifies the desired solution path was indeed tracked properly. Since each step can be certified independently, an *a posteriori* procedure is naturally parallelizable with a goal of making the time associated with using a parallelized *a posteriori* certified scheme roughly the same as the time it takes to heuristically track the path.

The examples presented in § 7 demonstrate that this *a posteriori* certification procedure is applicable for moderately sized systems which, for example, can be used to prove results about monodromy groups and continuous motion of an end effector. Based on the efficiency and practicality of this *a posteriori* certification procedure, we are working on generalizing this to other types of homotopies. For Newton homotopies, (4) guaranteed the existence of a smooth path along the interval under consideration. For general homotopies, one aims to develop a similar statement by estimating how the relevant data changes as t varies.

One key idea used in *a posteriori* certification is that both the startpoint and endpoint for an interval can be used to prove that path jumping did not occur. This is in contrast to an *a priori* certified tracking procedure in which one only has access to the startpoint and aims to approximate another point along the path. One could utilize (4) from Theorem 4.2 to develop an *a priori* certified tracking procedure for Newton homotopies. A refinement of such an *a priori* tracking approach is developed in [14] which also includes the use of an Euler predictor, i.e., a first-order prediction method, together with additional complexity results.

9. REFERENCES

[1] D.J. Bates, J.D. Hauenstein, and A.J. Sommese. Efficient path tracking methods. *Numer. Algorithms*, 58(4), 451–459, 2011.

[2] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Adaptive multiprecision path tracking. *SIAM J. Numer. Anal.*, 46(2), 722–746, 2008.

[3] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Bertini: software for numerical algebraic geometry. Available at bertini.nd.edu.

[4] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. *Numerically Solving Polynomial Systems with Bertini.* Volume 25 of *Software, Environments, and Tools*, SIAM, Philadelphia, 2013.

[5] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Stepsize control for path tracking. *Contemp. Math.*, 496, 21–31, 2009.

[6] C. Beltran. A continuation method to solve polynomial systems, and its complexity. *Numer. Math.*, 117(1), 89–113, 2011.

[7] C. Beltran and A. Leykin. Certified numerical homotopy tracking. *Exp. Math.*, 21(1), 69–83, 2012.

[8] C. Beltran and A. Leykin. Robust certified numerical homotopy tracking. *Found. Comput. Math.*, 13(2), 253–295, 2013.

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

[10] P. Bürguisser and F. Cucker. On a problem posed by Steve Smale. *Ann. Math.*, 174, 1785–1836, 2011.

[11] J.P. Dedieu, G. Malajovich, and M. Shub. Adaptive step-size selection for homotopy methods to solve polynomial equations. *IMA J. Numer. Anal.*, 33(1), 1–29, 2013.

[12] J.D. Hauenstein. Numerically computing real points on algebraic sets. *Acta Appl. Math.*, 125(1), 105–119, 2013.

[13] J.D. Hauenstein, I. Haywood, and A.C. Liddell, Jr. Cadenza: certifying homotopy paths for polynomial systems. Available at www.math.ncsu.edu/~jdhauens/cadenza.

[14] J.D. Hauenstein and A.C. Liddell, Jr. Certified predictor-corrector tracking for Newton homotopies. Preprint available at www.math.ncsu.edu/~jdhauens/preprints.

[15] J.D. Hauenstein and F. Sottile. Algorithm 921: alphaCertified: certifying solutions to polynomial systems. *ACM Trans. Math. Softw.*, 38(4), 28, 2012.

[16] J.D. Hauenstein and F. Sottile. alphaCertified: software for certifying solutions to polynomial systems. Available at www.math.tamu.edu/~sottile/research/stories/alphaCertified.

[17] J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Regeneration homotopies for solving systems of polynomials. *Math. Comput.*, 80, 345–377, 2010.

[18] S. Lang. *Real Analysis*, 2$^{\text{nd}}$ edition. Addison-Wesley, Reading, MA, 1983.

[19] 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(2-3), 109–133, 2008.

[20] A. Leykin. Numerical algebraic geometry. *J. Softw. Algebra Geom.*, 3, 5–10, 2011.

[21] A. Leykin and F. Sottile. Galois groups of Schubert problems via homotopy computation. *Math. Comp.*, 78(267), 1749–1765, 2009.

[22] A.J. Lotka. Undamped oscillations derived from the laws of mass action. *J. Amer. Chem. Soc.*, 42, 1595–1599, 1920.

[23] M. Shub. Complexity of Bézout's theorem. VI: Geodesics in the condition (number) metric. *Found. Comput. Math.*, 9(2), 171–178, 2009.

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

[25] A.J. Sommese, J. Verschelde, and C.W. Wampler. Using monodromy to decompose solution sets of polynomial systems into irreducible components. In *NATO Sci. Ser. II Math. Phys. Chem.*, 36, Kluwer Acad. Publ., Dordrecht, 2001, pp. 297–315.

[26] H.-J. Su, J.M. McCarthy, M. Sosonkina, and L.T. Watson. Algorithm 857: POLSYS_GLP – a parallel general linear product homotopy code for solving polynomial systems of equations. *ACM Trans. Math. Software*, 42(4), 561–579, 2006.

[27] J. Verschelde. Algorithm 795: PHCpack: A general-purpose sovler for polynomial systems by homotopy continuation. *ACM Trans. Math. Software*, 25(2), 251–276, 1999.

[28] V. Volterra. Variazionie fluttuazioni del numero d'individui in specie animali convivent. *Mem. Acad. Lincei.*, 2, 31–113, 1926.

[29] C.W. Wampler, J.D. Hauenstein, and A.J. Sommese. Mechanism mobility and a local dimension test. *Mech. Mach. Theory*, 46(9), 1193–1206, 2011.

[30] C.W. Wampler, B. Larson, and A. Edrman. A new mobility formula for spatial mechanisms. In *Proc. DETC/Mechanisms & Robotics Conf., Sept. 4–7, Las Vegas, NV (CDROM)*, 2007.

[31] C.W. Wampler and A.P. Morgan. Solving the kinematics of general 6R manipulators using polynomial continuation. *Robotics: Applied Mathematics and Computational Aspects*, K. Warwick, ed., Clarendon Press, Oxford, 1993, pp. 57–69.