Homotopy continuation: the theory of practice

Pierre Lairez
MATHEXP, Université Paris–Saclay, Inria, France

FoCM, July 9, 2026, Vienna, Austria

Inria

université PARIS-SACLAY

European Research Council
Established by the European Commission

Computational complexity theory

“In theoretical computer science and mathematics, computational complexity theory focuses on **classifying computational problems** according to their resource usage.”

Wikipedia

Numerical polynomial system solving

I — A complex polynomial system, n variables, n equations
O — An approximate zero

✓ $N^{1+o(1)}$ average complexity (Lairez, 2020)

- N is the number of coefficients in the dense representation
- BSS model
- regime $\min(\text{dimension}, \text{degree}) \to \infty$

Smale (1981, 1986), Renegar (1987), Shub and Smale (1993a, 1993b, 1993c, 1996, 1994), Smale (1998), Beltrán and Pardo (2008, 2009), Shub (2009), Beltrán and Shub (2009), Beltrán (2011), Beltrán and Pardo (2011), Bürgisser and Cucker (2011), Beltrán and Leykin (2012, 2013), Dedieu, Malajovich, and Shub (2013), Hauenstein and Liddell (2016), Armentano et al. (2016), Lairez (2017, 2020), and Bürgisser, Cucker, and Lairez (2023)

Is it really that easy?

Computational model

BSS $\neq$ Turing, but the theory is robust (Beltrán & Leykin, 2013).

Average complexity results

But what else can we do when worst-case complexity is infinite?

Choice of complexity parameters

N is not the most relevant. Evaluation complexity is interesting Bürgisser, Cucker, and Lairez (2023). Hard to combine with average analysis.

The resolution of complexity is poly(dimension, degree), not $O(1)$!

Artificial construction

We design algorithms not because they are good, but because we know how to analyze them.

The split

Theory

* Smale's α theory
* Integral complexity estimates
* Correct algorithms
* $\mathrm{poly}(n, d)$ resolution

Practice

* Heuristic algorithms
* Certification with interval arithmetic
* Finite precision
* Constant factors matter
* Predictors

How can we understand the runtime of practical methods through complexity?

Today: focus on homotopy continuation

(Not how to design homotopy paths, or average complexity)

Homotopy continuation problems

$F : [0,1] \times \mathbb{C}^n \longrightarrow \mathbb{C}^n$ is a polynomial map
η is a zero of F_1.

ζ_t is continuous, defined by $F_t(\zeta_t) = 0$ and $\zeta_1 = \eta$

In how many continuation steps can we approximate ζ_0?

The Shub–Smale approach

- Control the number of iterations in terms of

$$\gamma(f,z) = \sup_{k\geqslant 2} \left\| \tfrac{1}{k!} \mathrm{D}f(z)^{-1} \mathrm{D}^k f(z) \right\|^{\frac{1}{k-1}}$$

- Difficulties:
  - How to compute γ efficiently?
  - The variation of $\gamma(F_t, z)$ as t varies is hard to control

- Introduce an upper bound $\gamma(F,z) \leqslant \mu(F,z)$ that fixes the issues. The number of continuation steps is at most

$$C \int_0^1 \mu(F_t, \zeta_t) \|\dot{\zeta}_t\| \mathrm{d}t.$$

- Shub and Smale (1996) suggest, but cannot prove, the bound

$$C \int_0^1 \gamma(F_t, \zeta_t) \|\dot{\zeta}_t\| \mathrm{d}t.$$

Newton homotopies

$f : \mathbb{C}^n \to \mathbb{C}^n$ is a polynomial map
$\eta \in \mathbb{C}^n$

ζ_t is continuous, defined by $f(\zeta_t) = tf(\eta)$ and $\zeta_1 = \eta$

Newton homotopies contain the general case!

Choose $f(t, z) = (t, F_t(z))$ and starting point (t, η) to reduce a general homotopy problem to a Newton homotopy with one additional variable.

In how many continuation steps can we approximate ζ_0?

The technical difficulties vanish, Hauenstein and Liddell (2016) prove the bound

$$C \int_0^1 \gamma(f, \zeta_t) \|\dot{\zeta}_t\| \mathrm{d}t.$$

The continuation algorithm (Hauenstein & Liddell, 2016)

def $\mathit{continuation}(f, \zeta)$**:**
  $t \leftarrow 1$
  $z \leftarrow \zeta$
  $v \leftarrow f(\zeta)$
  while $t > 0$**:**
    $\varepsilon \leftarrow \|\mathrm{D}f(z)^{-1} f(z)\| \gamma(f, z)$
    $t \leftarrow \max\left(t - \frac{1}{5\varepsilon}, 0\right)$
    $z \leftarrow \mathit{Newton}(f - tv, z)$
  return z

! Same spirit as Shub (2009) and Beltrán (2011), but much simpler because of the Newton homotopy setting.

A practical continuation algorithm

def *continuation*(f, ζ)**:**
    $t \leftarrow 1$
    $z \leftarrow \zeta$
    $v \leftarrow f(\zeta)$
    while $t > 0$**:**
        $t \leftarrow t - \delta$ [trial and error]
        $z \leftarrow$ *predictor(t)*   [extrapolation based on previous values]
        $z \leftarrow \mathit{Newton}^k(f - tv, z)$
        if the Newton iteration does not converge:
            restart the iteration with a smaller δ
    return z

! We test only necessary conditions for convergence

A practical certified algorithm (Guillemot & Lairez, 2024)

Validated Numerics for Algebraic Path Tracking

Alexandre Guillemot
Inria, Université Paris-Saclay, Palaiseau, France

Pierre Lairez
Inria, Université Paris-Saclay, Palaiseau, France

ABSTRACT

Using validated numerical methods, interval arithmetic and Taylor models, we propose a certified predictor-corrector loop for tracking zeros of polynomial systems with a parameter. We provide a Rust implementation which shows tremendous improvement over existing software for certified path tracking.

CCS CONCEPTS

• **Mathematics of computing** → **Interval arithmetic**.

KEYWORDS

numerical algebraic geometry, certified path tracking

ACM Reference Format:
Alexandre Guillemot and Pierre Lairez. 2024. Validated Numerics for Algebraic Path Tracking. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '24), July 16–19, 2024, Raleigh, NC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3666000.3669673

1 INTRODUCTION

Path tracking, or homotopy continuation, is the backbone of numerical algebraic geometry [29]. Given a polynomial map $(t,x) \mapsto F_t(x)$, from $\mathbb{C} \times \mathbb{C}^n$ to $\mathbb{C}^n$, and a regular zero $\zeta \in \mathbb{C}^n$ of F_0, we want to compute for $t \in [0,1]$ the continuation ζ_t, which is, as-

precision, which should be as low as possible for performance, but large enough to ensure appropriate convergence properties and termination.

The use of Taylor models enables predictors (Section 6), such as the tangent predictor, or the cubic Hermite predictor, as observed by van der Hoeven [31], leading to tremendous improvement over previous methods based on Smale's α-theory. We provide a Rust implementation that we compare with existing software for path tracking, both certified and noncertified (Section 7). We find that the number of iterations performed using the Hermite predictor is in the same order of magnitude as noncertified approaches.

Related work. The method of path tracking received vast attention in the context of numerical multivariate polynomial system solving. It is the method of choice for most state-of-the-art software: PHCpack [32], Bertini [2] or HomotopyContinuation.jl [6], for example. Despite their effectiveness (or perhaps *because* of their effectiveness), and despite recent work which brings robustness to unprecedented levels [30], this software does not guarantee correctness, that is the consistency of the result with the definition of the continuation. We obtain zeros of the target system which we may certify independently, but we are not certain that the paths were tracked correctly, without swapping for example.

There are countless certified path tracking algorithms based on

- Uses interval arithmetic and Moore's criterion instead of γ.
- Can use polynomial predictors.

Questions

- How does Smale's γ compare to Moore's criterion?

  Step-size computations rely on criteria to isolate a zero of a polynomial system; how do they compare?

- What is the cost of certification?

- Can we quantify the benefit of predictors?

Equivalence of root isolation criteria

Let $B \subseteq \mathbb{C}^n$ be the unit ball.
Four ways to quantify the deviation of $f(z)$ from its linearization on a ball $z + rB$:

$$
\begin{aligned}
\Delta_0(z,r) &= \sup_{u \in rB} \left\| \mathrm{D}f(z)^{-1} \left(f(z+u) - f(z)\right) - u \right\| \\
\Delta_1(z,r) &= \sup_{u \in rB} \left\| \mathrm{id} - \mathrm{D}f(z)^{-1} \circ \mathrm{D}f(z+u) \right\| \cdot r \\
\Delta_2(z,r) &= \sup_{u \in rB} \left\| \mathrm{d}f(z)^{-1} \circ \mathrm{d}^2 f(z+u) \right\| \cdot \tfrac{1}{2} r^2 \\
\Delta_\infty(z,r) &= \sum_{k \geqslant 2} \left\| \mathrm{D}f(z)^{-1} \circ \mathrm{D}^k f(z) \right\| \frac{r^k}{k!} \\
\gamma_i(f,z) &= \sup_{r>0} \quad \min \left\{ \frac{1}{r}, \frac{\Delta_i(z,r)}{r^2} \right\} \\
\alpha_i(z) &= \| \mathrm{D}f(z)^{-1} \cdot f(z) \| \gamma_i(z).
\end{aligned}
$$

Root isolation

Theorem (Rouché)
If $\alpha_0(f, z) < \frac{1}{4}$, then f has a unique zero in the ball $z + \frac{1}{2\gamma_0(f,z)}B$.

Theorem (Moore)
If $\alpha_1(f, z) < \frac{1}{4}$, then f has a unique zero in the ball $z + \frac{1}{2\gamma_1(f,z)}B$, and the quasi-Newton iteration converges.

Theorem (Kantorovich)
If $\alpha_2(f, z) < \frac{1}{4}$, then f has a unique zero in the ball $z + \frac{1}{2\gamma_2(f,z)}B$ and the Newton iteration converges.

Theorem (Smale)
If $\alpha_\infty(f, z) < \frac{1}{4}$, then f has a unique zero in the ball $z + \frac{1}{2\gamma_\infty(f,z)}B$, and the Newton iteration converges.

Theorem
$\gamma_0 \approx \gamma_1 \approx \gamma_2 \approx \gamma_\infty \approx \gamma_{\text{Smale}}$ (up to multiplicative constants)

Conclusion on Smale vs Moore

- ✓ Smale's and Moore's criteria are equally powerful.
- γ_{Smale} (or γ_∞) and γ_1 are both hard to compute (even up to a constant factor).
- We can compute an upper bound for γ_1 very efficiently using interval arithmetic...
- ... but there is an overestimation amplified by the conditioning of the Jacobian $\mathrm{D}f(z)^{-1}$...
- ... so the criterion used in (Guillemot & Lairez, 2024) is qualitatively worse than Smale's.

The role of predictors

- Predictors aim to make larger steps than $\frac{1}{\gamma}$.

- Taylor predictors are limited by the radius of convergence of ζ_t.

- **Theorem**

  $\gamma_{\text{Smale}}(\zeta) \leqslant 6\,\gamma_{\text{Smale}}(f, \zeta)$, where $\gamma_{\text{Smale}}(\zeta) = \sup_{k \geqslant 2} \left| \frac{1}{k!} \frac{\zeta_t^{(k)}}{\zeta_t'} \right|$,
  and this is tight in the worst case.

- So it is hard to *prove* that predictors improve the step size, even if it is obvious in practice.

- Taylor predictors can compensate for interval overestimation!

References I

Armentano, D., Beltrán, C., Bürgisser, P., Cucker, F., & Shub, M. (2016). Condition length and complexity for the solution of polynomial systems. *Found. Comput. Math.*, *16*(6), 1401–1422.

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

Beltrán, C., & Leykin, A. (2012). Certified numerical homotopy tracking. *Exp. Math.*, *21*(1), 69–83.

Beltrán, C., & Leykin, A. (2013). Robust certified numerical homotopy tracking. *Found. Comput. Math.*, *13*(2), 253–295.

Beltrán, C., & Pardo, L. M. (2008). On Smale's 17th problem: A probabilistic positive solution. *Found. Comput. Math.*, *8*(1), 1–43.

Beltrán, C., & Pardo, L. M. (2009). Smale's 17th problem: Average polynomial time to compute affine and projective solutions. *J. Amer. Math. Soc.*, *22*(2), 363–385.

Beltrán, C., & Pardo, L. M. (2011). Fast linear homotopy to find approximate zeros of polynomial systems. *Found. Comput. Math.*, *11*(1), 95–129.

Beltrán, C., & Shub, M. (2009). Complexity of Bezout's theorem. VII. Distance estimates in the condition metric. *Found. Comput. Math.*, *9*(2), 179–195.

References II

Bürgisser, P., & Cucker, F. (2011). On a problem posed by Steve Smale. *Ann. of Math. (2), 174*(3), 1785–1836.

Bürgisser, P., Cucker, F., & Lairez, P. (2023). Rigid continuation paths II. Structured polynomial systems. *Forum Math. Pi*, *11*, e12.

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

Guillemot, A., & Lairez, P. (2024). Validated numerics for algebraic path tracking. *Proc. ISSAC 2024*, 36–45.

Hauenstein, J. D., & Liddell, A. C. (2016). Certified predictor–corrector tracking for Newton homotopies. *J. Symb. Comput.*, *74*, 239–254.

Lairez, P. (2017). A deterministic algorithm to compute approximate roots of polynomial systems in polynomial average time. *Found. Comput. Math.*, *17*(5), 1265–1292.

Lairez, P. (2020). Rigid continuation paths I. Quasilinear average complexity for solving polynomial systems. *J. Am. Math. Soc.*, *33*(2), 487–526.

Renegar, J. (1987). On the efficiency of Newton's method in approximating all zeros of a system of complex polynomials. *Math. Oper. Res.*, *12*(1), 121–148.

References III

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

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

Shub, M., & Smale, S. (1993b). Complexity of Bezout's theorem. II. Volumes and probabilities. In *Computational algebraic geometry (Nice, 1992)* (pp. 267–285, Vol. 109). Birkhäuser.

Shub, M., & Smale, S. (1993c). Complexity of Bezout's theorem. III. Condition number and packing. *J. Complexity*, *9*(1), 4–14.

Shub, M., & Smale, S. (1994). Complexity of Bezout's theorem. V. Polynomial time. *Theoret. Comput. Sci.*, *133*(1), 141–164.

Shub, M., & Smale, S. (1996). Complexity of Bezout's theorem. IV. Probability of success; extensions. *SIAM J. Numer. Anal.*, *33*(1), 128–148.

Smale, S. (1981). The fundamental theorem of algebra and complexity theory. *Bull. Am. Math. Soc.*, *4*(1), 1–36.

Smale, S. (1986). Newton's method estimates from data at one point. In *The merging of disciplines: New directions in pure, applied, and computational mathematics (Laramie, Wyo., 1985)* (pp. 185–196). Springer.

References IV

Smale, S. (1998). Mathematical problems for the next century. *Math. Intell.*, *20*(2), 7–15.