Finding one root of a polynomial system

How to improve the complexity?

Pierre Lairez
Inria, France

Felipe’s Fest
Berlin — 19 august 2019
On a problem posed by Steve Smale

By Peter Bürgisser and Felipe Cucker

Abstract

The 17th of the problems proposed by Steve Smale for the 21st century asks for the existence of a deterministic algorithm computing an approximate solution of a system of \( n \) complex polynomials in \( n \) unknowns in time polynomial, on the average, in the size \( N \) of the input system. A partial solution to this problem was given by Carlos Beltrán and Luis Miguel Pardo who exhibited a randomized algorithm doing so. In this paper we further extend this result in several directions. Firstly, we exhibit a linear homotopy algorithm that efficiently implements a nonconstructive idea of Mike Shub. This algorithm is then used in a randomized algorithm, call it \( LV \), à la Beltrán-Pardo. Secondly, we perform a smoothed analysis (in the sense of Spielman and Teng) of algorithm \( LV \) and prove that its smoothed complexity is polynomial in the input size and \( \sigma^{-1} \), where \( \sigma \) controls the size of the random perturbation of the input systems. Thirdly, we perform a condition-based analysis of \( LV \). That is, we give a bound, for each system \( f \), of the expected running time of \( LV \) with input \( f \). In addition to its dependence on \( N \) this bound also depends on the condition of \( f \). Fourthly, and to conclude, we return to Smale’s 17th problem as originally formulated for deterministic algorithms. We exhibit such an algorithm and show that its average complexity is \( N^{O((\log \log N))} \). This is nearly a solution to Smale’s 17th problem.

Contents

1. Introduction 1786
Recognitions 1791
2. Preliminaries 1791
2.1. Setting and notation 1791
2.2. Newton’s method 1793
2.3. Condition numbers 1793
2.4. Gaussian distributions 1794

P.B. was partially supported by DFG grant BU 1371/2-1 and Paderborn Institute for Scientific Computation (PaSCo). F. C. was partially supported by GRF grant CityU 100810.

An extended abstract of this work was presented at STOC 2010 under the title “Solving Polynomial Equations in Smoothed Polynomial Time and a Near Solution to Smale’s 17th Problem.”
Solving polynomial systems in polynomial time?

Can we compute the roots of a polynomial system in polynomial time? **Likely not, deciding feasibility is NP-complete.**

Can we compute the complex roots of \( n \) equations in \( n \) variables in polynomial time? **No, there are too many roots.**

<table>
<thead>
<tr>
<th>Bézout bound vs. input size (( n ) polynomial equations, ( n ) variables, degree ( \delta ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>degree</td>
</tr>
<tr>
<td>input size</td>
</tr>
<tr>
<td>#roots</td>
</tr>
</tbody>
</table>
Finding one root: a purely numerical question

#roots $\gg$ input size  To compute a single root, do we have to pay for #roots?

using exact methods  Having one root is having them all (generically).

using numerical methods  One may approximate one root disregarding the others.

polynomial complexity?  Maybe, but only with numerical methods

This is Smale's question

Now solved, let's ask for more!
Numerical continuation

\( F_t \) a polynomial system depending continuously on \( t \in [0, 1] \)
\( z_0 \) a root of \( F_0 \)

function NumericalContinuation(\( F_t, z_0 \))

\( t \leftarrow 0 \)
\( z \leftarrow z_0 \)

repeat
\( t \leftarrow t + \Delta t \)
\( z \leftarrow \text{Newton}(F_t, z) \)
until \( t \geq 1 \)
return \( z \)
end function

- Solves any generic system
- How to set the step size \( \Delta t \)?
- How to choose the start system \( F_0 \)?
- How to choose a path?
A short history
Average analysis

the complexity is unbounded near singular cases.
⇝ stochastic analysis

global distribution centered Gaussian in the space of all polynomial systems

local distribution non-centered Gaussian

randomized algorithms choosing the continuation path may need randomization

Lairiez (2017) this can be derandomized eliminated for average analysis

\[ x = 0.6044025624180895161178081249104686505290197465315910133226678885000001621027 \]
\[ \text{noise extraction} \]
\[ 0.505290197465315910133226678885000001621027 \]
\[ \text{truncation} \]
\[ 0.6044025624180895161178081249104686 \]
$n$ complex variables
$n$ random equations of degree $\delta$
input size $N$

**input distribution** centered

**# of steps** $\text{poly}(\delta^n)$, with high probability

**starting system** $x_1^\delta = 1, \ldots, x_n^\delta = 1$

**continuation path** $(1 - t)F_0 + tF_1$

**previous best** $\emptyset$
$n$ complex variables

$n$ random equations of degree $\delta$

input size $N$

- **input distribution**: centered
- **# of steps**: $\text{poly}(N)$, with high probability
- **starting system**: not constructive
- **continuation path**: $(1 - t)F_0 + tF_1$
- **previous best**: $\text{poly}(\delta^n)$
$n$ complex variables
$n$ random equations of degree $\delta$
input size $N$

**input distribution** centered

**# of steps** $O(n\delta^{3/2}N)$, on average

**starting system** random system, sampled directly with a root

**continuation path** $(1 - t)F_0 + tF_1$

**previous best** $\text{poly}(\delta^n) \rightarrow \text{poly}(N)$
$n$ complex variables

$n$ random equations of degree $\delta$

input size $N$

**input distribution** non-centered, variance $\sigma^2$, really relevant to applications!

**# of steps** $O(n\delta^{3/2} N/\sigma)$, on average

**starting system** idem Beltrán-Pardo

**continuation path** $(1 - t)F_0 + tF_1$

**previous best** $\emptyset$
$n$ complex variables

$n$ random equations of degree $\delta$

input size $N$

**input distribution** centered

**# of steps** $O(n\delta^{3/2} N^{1/2})$, on average

starting system idem Beltrán-Pardo

continuation path $(1 - t)F_0 + tF_1$

**previous best** $\text{poly}(\delta^n) \rightarrow \text{poly}(N) \rightarrow O(n\delta^{3/2} N)$
Lairez (2017)

$n$ complex variables
$n$ random equations of degree $\delta$
input size $N$

**input distribution** centered

**# of steps** $O(n^3 \delta^2)$, on average

**starting system** an analogue of Beltrán-Pardo

**continuation path** $(f_1 \circ u_1^{1-t}, \ldots, f_n \circ u_n^{1-t})$, with $u_i \in U(n + 1)$
(rigid motion of each equations)

**previous best** $\text{poly}(\delta^n) \to \text{poly}(N) \to O(n\delta^{3/2}N) \to O(n\delta^{3/2}N^{1/2})$
Improving the conditioning
How to improve the complexity?

By making **bigger steps!**

\[ z = \text{the current root} \]

\[ \rho(F, z) = \text{inverse of the radius of the bassin of attraction of } z \]

\[ \mu(F, z) = \sup \left[ \text{over } F' \sim F \text{ and } F'(z') = 0 \right] \frac{\text{dist}(z, z')}{\|F - F'\|} \]

**step size heuristic**

\[ \frac{1}{\Delta t} \approx \rho(F, z) \cdot \frac{\Delta z}{\Delta t} \]

\[ \leq \mu(F, z) \cdot \mu(F, z). \]

**average analysis**

Each factor \( \mu \) contributes \( O(N^{1/2}) \) in the average \# of steps.

To go down to \( \text{poly}(n, \delta) \), we must improve both.
Changing the path

**an old idea** Can we choose a path that keeps $\mu(F, z)$ low? i.e. that stays far from singularities?

**yes!** Beltrán, Shub (2009)

...but not applicable for polynomial system solving.

(Pictures by Juan Criado del Rey.)
Rigid continuation algorithm

**input** $f_1, \ldots, f_n$, homogeneous polynomials of degree $\delta$ in $x_0, \ldots, x_n$

1. Pick $x \in \mathbb{P}^n(\mathbb{C})$
2. For $1 \leq i \leq n$,
   a. compute one point $p_i \in \mathbb{P}^n(\mathbb{C})$ such that $f_i(p_i) = 0$
   b. pick $u_i \in U(n+1)$ such that $u_i(x) = p_i$.
3. Perform the numerical continuation with
   $$F_t = (f_1 \circ u_1^{1-t}, \ldots, f_n \circ u_n^{1-t}).$$

**big win** the parameter space has $O(n^3)$ dimensions, the conditioning is poly$(n)$ on average

**total complexity** $O(n^6 \delta^4 N) = N^{1+o(1)}$ operation on average, **quasilinear**
Toward structured systems
Why structured systems?

**structures** sparse
symmetries
low evaluation complexity black box

This includes most practical examples!

**observation** A poly($N$) complexity is far from what we observe in practice.

We want poly($n, \delta$) cost(input)
**input**  \( F \) given as a **black box** function

**question** Can we adapt the rigid continuation algorithm? **Yes!**, but with small probability of failure

**difficulty** Computing \( \gamma \) requires all coefficients, costs \( N \gg \text{cost}(F) \).

**stochastic formulation**

\[
\gamma(f, z) \approx \min_{\rho > 0} \frac{\mathbb{E} \left| f(z + \rho w) - f(z) \right|}{\rho^2 \| d_z f \|},
\]

with \( w \) uniformly distributed in the unit ball.

**Stochastic optimization problem**
Random black box input

**input**  $F$ given as a black box function, randomly distributed

**question** Is the average complexity $\text{poly}(n, \delta) \text{cost}(F)$? Watch arXiv...

**random black boxes** What it is?
A random model for a black box (homogeneous) polynomial:

$$f(x_0, \ldots, x_n) = \text{trace}(A_1(x_0, \ldots, x_n) \cdots A_\delta(x_0, \ldots, x_n)),$$

where the $A_i$ are $r \times r$ matrices with degree 1 entries, coefficients are i.i.d. Gaussian.

**evaluation complexity** $O(r^3 \delta + r^2 n)$
The parameter $r$ reflects the complexity of evaluating $f$.
Polynomially equivalent to Valiant’s determinantal complexity.
Thank you!

Thank you!

Thank you!