Complexity of computing zeros of structured polynomial systems

Peter Bürgisser

joint work with

Felipe Cucker and Pierre Lairez

Los Angeles Combinatorics and Complexity Seminar

November 10, 2020

Solving polynomial systems

- Given n homogeneous polynomial equations of degree d,

  $$f_1(z) = 0, \dots, f_n(z) = 0,$$

  in $n+1$ variables. Want to compute solutions in $\mathbb{P}^n(\mathbb{C})$.
- Intensively studied computational question! Not possible to summarize all the contributions here.
- Let's assume the input polynomials f_i are sufficiently generic.
- Well known: algebraic algorithms can compute all d^n solutions with $d^{O(n)}$ arithmetic operations: e.g., Renegar ('89), Lakshman ('91), Giusti, Lecerf, Salvy ('01).
- Can we do better if we just want one (or a few) solutions?
- Numerical homotopy continuation algorithms are capable of this, see impressive software by Breiding and Timme: `https://www.juliahomotopycontinuation.org`

Homotopy continuation algorithms

- Such algorithms have been known for quite some time. Their complexity was investigated in detail by Shub & Smale in the 90ies.
- Smale's 17th problem (1998):

  > *Can a zero of n complex polynomial equations in n unknowns be found approximately, on the average, in polynomial time with a uniform algorithm?*

- Here input polynomials are assumed to be distributed according to a unitary invariant Gaussian distribution (Kostlan, Bombieri, Weyl).
- Smale's question was given a positive answer by Lairez (2017), based on work by Shub & Smale, Shub, Beltran & Pardo, and Bürgisser & Cucker.
- Essentially optimal result for dense model by Lairez ('20): quasilinear expected cost $\mathrm{poly}(n, d)N$, where N is input size in dense representation (number of coefficients).

How meaningful is the dense model?

- Dense representation, polynomials are given by their full list of coefficients: $N := n\binom{d+n}{n}$.

- But: Most systems of interest are structured in some sense; they lie in a set of measure zero in the input space of all polynomials.

- Thus the above probabilistic analysis doesn't tell us anything about the behaviour of the above algorithms on these structured inputs!

The dense setting is only the first step towards understanding the problem!

- Ask about refinement of Smale's question for structured systems.

Unitary invariance

- Unitary invariance is an important feature enabling the probabilistic analysis in the mentioned results.
- The unitary group $U(n+1)$ acts on the space H of homogeneous polynomials of degree d in $n+1$ variables by linear transformation: "orthogonal coordinate transformation".
- The product $\mathcal{U} := U(n+1)^n$ of unitary groups acts on space $\mathcal{H} := H^n$:

$$(f_1, \dots, f_n) \cdot (u_1, \dots, u_n) := (f_1 \circ u_1, \dots, f_n \circ u_n).$$

- Our new result applies to structured settings that respect unitary univariance.
- We will bound the expected running time of algorithms (solvers) over $\mathcal{U}$-orbits, that is, we average over $\mathcal{U}$.

Systems given by black-box evaluation…

- There are many ways to define structured systems.

- One may consider sparse systems, i.e., prescribe the support (occurring monomials) of the polynomials. See Malajovich ('19-'20).
- However, this model is not unitary invariant.

- Here we only assume that the polynomials are given by a black box: i.e., an evaluation routine. (No need to know what the routine actually does.)
- This is a common assumption in optimization: a function (and its gradient) are given by a black-box routine.
- Note: black box for f_i easily gives black box for composition $f_i \circ u_i$.

...and solving them

- Consider system $F = (f_1, \dots, f_n) \in H^n$ s.t. zero set of f_i has no singularities.
- We assign to F a quantity reflecting its "numerical conditioning":

$$\Gamma(F) := \left(\Gamma(f_1)^2 + \cdots + \Gamma(f_n)^2\right)^{\frac{1}{2}} < \infty.$$

- $\Gamma(f_i)$ is essentially the average of Smale's γ quantity $\gamma(f_i, z)$, averaged over the compact hypersurface of zeros of f_i.
- $L(F)$: number of arithmetic operations sufficient to evaluate F.

Theorem (I)

We exhibit an algorithm BBS, which on input $F \in \mathcal{H}$ given as black-box, and $\epsilon > 0$, computes an approximate zero of F with probability at least $1 - \epsilon$.

On input $u \cdot F$, where $u \in \mathcal{U}$ is uniformly random, this algorithm performs at most

$$\operatorname{poly}(n, d) \cdot L(F) \cdot \left(\Gamma(F) \log \Gamma(F) + \log \log \epsilon^{-1}\right)$$

operations on average.

Solving random systems with unitary invariant distribution

- Theorem I applies to any prob. distribution of $F \in \mathcal{H}$ that is unitary invariant.
- On a random input $F \in \mathcal{H}$, the expected number of operations is bounded by
$$\mathrm{poly}(n, \delta) \cdot L \cdot \left(\Gamma \log \Gamma + \log \log \epsilon^{-1}\right)$$
where
$$\Gamma = \mathbb{E}[\Gamma(F)^2]^{\frac{1}{2}}$$
and L denotes an upper bound on $L(F)$.
- We apply this to the class of polynomial systems computed by small algebraic branching programs.
- This class is unitary invariant and, by definition, these systems have small L (evaluation complexity).
- Have a natural invariant Gaussian distribution on this class.
- We managed to effectively upper bound $\Gamma(F)$ in this case.

Algebraic branching programs

- Algebraic branching programs (ABPs), introduced by Nisan ('91), play an important role in algebraic complexity theory: notably in Valiant's VP versus VNP theory.
- ABPs provide an elegant graphical way of formalizing computations with polynomials, but the most concise way to express the model is using matrices.
- Fix $r_0, \ldots, r_d \in \mathbb{N}_{>0}$ with $r_0 = r_d$ and let

  $$A_i(z) = A_{i0}\, z_0 + \cdots + A_{in}\, z_n$$

  with complex matrices A_{ij} of format $r_{i-1} \times r_i$, $1 \le i \le d$.
- The trace of iterated matrix multiplication

  $$f(z) = \mathrm{tr}\,(A_1(z) \cdots A_d(z))\,.$$

  is the polynomial defined by the corresponding ABP.
- By associativity of matrix multiplication, can evaluate $f(z)$ with a total of $O(ndr^3)$ arithmetic operations, where $r := \max_i r_i$.

Gaussian algebraic branching programs

- We assume now that the A_{ij} are independent complex standard Gaussian matrices and focus on the distribution of the (highly structured) random polynomial

$$f(z) = \operatorname{tr}\left(A_1(z)\cdots A_d(z)\right).$$

- Important: the distribution of f is unitarily invariant
- The support S of this distribution is a low dimensional algebraic subvariety of the space H of d-dimensional forms: $\dim S \leq d(n+1)^2$.

Theorem (II)

We have

$$\mathbb{E}\left[\Gamma(f)^2\right] \;\leq\; \tfrac{3}{4}d^3(d+n)\log d,$$

provided $r_1, \ldots, r_{d-1} \geq 2$. *Otherwise,* $\Gamma(f) = \infty$ *almost surely.*

The proof is a technical tour de force ...

Solving systems given by Gaussian ABPs

Corollary (II)

If $f_1, \dots, f_n$ are given by independent Gaussian random ABPs of degree at most d, format $r_1, \dots, r_{d-1} \geq 2$, and evaluation complexity at most L, then algorithm BBS computes a zero of $f_1 = 0, \dots, f_n = 0$ with probability at least $1 - \epsilon$ in

$$\mathrm{poly}(n, d) \cdot L \cdot \log\log \epsilon^{-1}$$

operations on average.

Our result may be interpreted as a first step towards providing an affirmative answer to a refined version of Smale's 17th question, concerned with structured systems of polynomial equations.

Rigid Homotopy Continuation

- General framework due to Pierre Lairez, J. AMS 2019, substantially improving Shub & Smale and basis of our work.
- Fix $F = (f_1, \dots, f_n) \in \mathcal{H}$ s.t. zero set of f_i has no singularities. Consider the compact "solution variety"

$$\mathcal{V} := \{((u_1, \dots, u_n), z) \in \mathcal{U} \times \mathbb{P}^n \mid f_1(u_1(z)) = 0, \dots, f_n(u_n(z)) = 0\}.$$

- $\mathcal{V}$ is invariant under action of $\mathcal{U}$, have $\mathcal{U}$-invariant prob. distribution.
- It is possible to efficiently compute a sample $(u, z) \in \mathcal{V}$.
- Connect u to identity I by a geodesic path $[0, 1] \to \mathcal{U}$, $t \mapsto u_t$, s.t. $u_0 = u$, $u_1 = I$. Continue the zero z by a path $[0, 1] \to \mathbb{P}^n$, $t \mapsto z_t$ s.t. $z_0 = z$ (almost surely possible).

Stepsizes via Monte Carlo sampling

- Implement numerical continuation via Newton iteration.
- Appropriate step sizes are given by Smale's parameter $\gamma(f, z)$.
- Previous algorithm estimated stepsize in terms of condition number, which leads to large step sizes and is much too wasteful!
- Definition of $\gamma(f, z)$ involves norm of higher order derivatives

  $$\tfrac{1}{k!}\left\| D_z^k f \right\|_F = \| f(z+\bullet)_k \|_W,$$

  where $p = f(z+\bullet)_k$ is the homogeneous component of degree k of the shifted polynomial $x \mapsto f(z+x)$.
- Trick from algebraic complexity: p is easy to compute in black box model.
- Estimate $\gamma(f, z)$ using Monte Carlo random samplling based on

  $$\|p\|_W^2 = \tbinom{n+1+k}{k}\, \mathbb{E}_w[|p(w)|^2],$$

  with w chosen uniformly at random in euclidean unit ball B of $\mathbb{C}^{n+1}$.
- The rigid continuation algorithm is Monte-Carlo: it fails with controlled error probability.

Thank you for your attention!