The Work of Mike Shub in Complexity

Felipe Cucker

City University of Hong Kong

Shubfest, Toronto 2012

Complexity Theory

Goal: Determine the amount of resources (most commonly, computer time) necessary to solve problems with a computer.

Complexity Theory

Goal: Determine the amount of resources (most commonly, computer time) necessary to solve problems with a computer.

This broad goal alternates its focus between two extremes:

Complexity Theory

Goal: Determine the amount of resources (most commonly, computer time) necessary to solve problems with a computer.

This broad goal alternates its focus between two extremes:

(G) To develop a general theory of computational cost (which includes formal models of computation, diverse cost notions, complexity classes built upon them, complete problems in these classes, and —the ultimate desideratum— separations beteeen these complexity classes).

Complexity Theory

Goal: Determine the amount of resources (most commonly, computer time) necessary to solve problems with a computer.

This broad goal alternates its focus between two extremes:

(G) To develop a general theory of computational cost (which includes formal models of computation, diverse cost notions, complexity classes built upon them, complete problems in these classes, and —the ultimate desideratum— separations beteeen these complexity classes).

(P) To analyze (in terms of cost) the behavior of specific algorithms (meant to solve specific problems).

Mike has worked on both ends of this spectrum with contributions that can be grouped in 3 main themes:

Mike has worked on both ends of this spectrum with contributions that can be grouped in 3 main themes:

(1) Zeros of Polynomial Systems.

Mike has worked on both ends of this spectrum with contributions that can be grouped in 3 main themes:

(1) Zeros of Polynomial Systems.

(2) Structural Complexity for Numerical Problems.

Mike has worked on both ends of this spectrum with contributions that can be grouped in 3 main themes:

(1) Zeros of Polynomial Systems.

(2) Structural Complexity for Numerical Problems.

(3) Conditioning of Numerical Problems.

Zeros of Polynomial Systems

- **M.S.**, S. Smale. “Computational complexity. On the geometry of polynomials and a theory of cost.” I. *Ann. Sci. École Norm. Sup.*, 1985. II. *SIAM J. Comput.*, 1986.

One polynomial in one variable.

Zeros of Polynomial Systems

- **M.S.**, S. Smale. "Computational complexity. On the geometry of polynomials and a theory of cost." I. *Ann. Sci. École Norm. Sup.*, 1985. II. *SIAM J. Comput.*, 1986.

One polynomial in one variable.

- **M.S.**, S. Smale. "Complexity of Bézout's Theorem." I, II, III, IV, and V, 1993–1996.

n polynomials in $n+1$ homogeneous variables.

Smale's 17th problem: Can one find an approximate zero of a system (n polynomials in $n+1$ homogeneous variables) in time polynomial on the average?

Smale's 17th problem: Can one find an approximate zero of a system (n polynomials in $n+1$ homogeneous variables) in time polynomial on the average?

approximate zero: a point from which Newton's method converges to a zero, immediately, quadratically fast.

Smale's 17th problem: Can one find an approximate zero of a system (n polynomials in $n+1$ homogeneous variables) in time polynomial on the average?

approximate zero: a point from which Newton's method converges to a zero, immediately, quadratically fast.

polynomial time: number of arithmetic operations bounded by $N^{\mathcal{O}(1)}$ where N is the size of the input system f.

Smale's 17th problem: Can one find an approximate zero of a system (n polynomials in $n+1$ homogeneous variables) in time polynomial on the average?

approximate zero: a point from which Newton's method converges to a zero, immediately, quadratically fast.

polynomial time: number of arithmetic operations bounded by $N^{\mathcal{O}(1)}$ where N is the size of the input system f.

on the average: w.r.t. a Gaussian distribution on the input f.

Smale's 17th problem: Can one find an approximate zero of a system (n polynomials in $n+1$ homogeneous variables) in time polynomial on the average?

approximate zero: a point from which Newton's method converges to a zero, immediately, quadratically fast.

polynomial time: number of arithmetic operations bounded by $N^{\mathcal{O}(1)}$ where N is the size of the input system f.

on the average: w.r.t. a Gaussian distribution on the input f.

$$D := \max\{d_1, \ldots, d_n\} \qquad\qquad N \approx n\binom{D+n}{n}$$

Adaptive linear homotopy

- Given an initial pair (g, ζ) with $g(\zeta) = 0$ and an input f:

Adaptive linear homotopy

- Given an initial pair (g, ζ) with $g(\zeta) = 0$ and an input f:
- Consider the line segment $[g, f]$ connecting g and f. It consists of the systems

$$q_t := (1 - t)g + tf \qquad \text{for } t \in [0, 1].$$

Adaptive linear homotopy

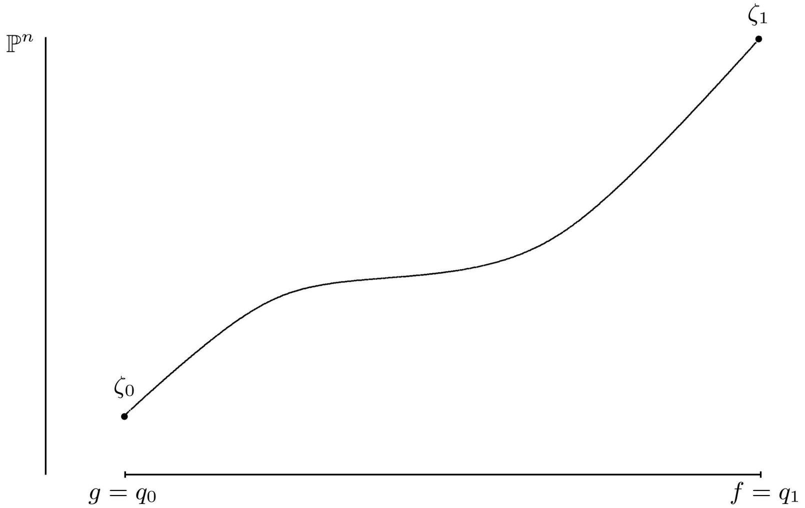
- Given an initial pair (g, ζ) with $g(\zeta) = 0$ and an input f:
- Consider the line segment $[g, f]$ connecting g and f. It consists of the systems

$$q_t := (1-t)g + tf \qquad \text{for } t \in [0,1].$$

- If no q_t has a multiple zero, then there exists a unique lifting of this segment to a curve

$$t \in [0,1] \mapsto (q_t, \zeta_t)$$

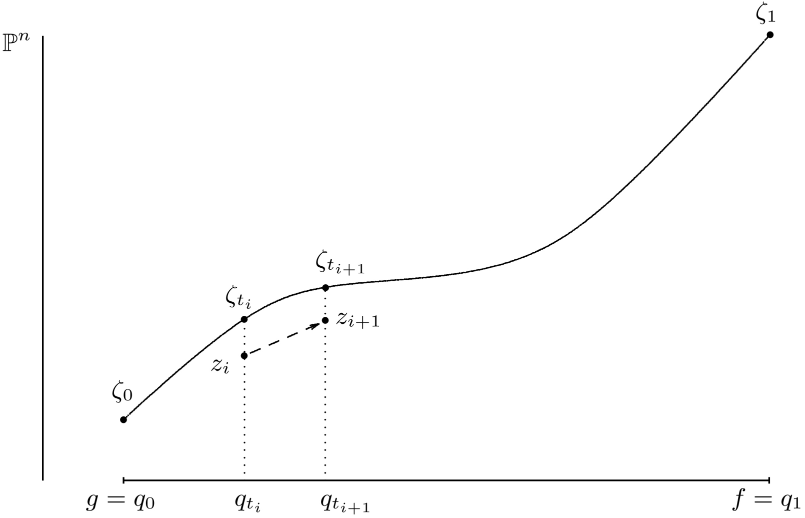
such that $\zeta_0 = \zeta$. Since $q_1 = f$, ζ_1 is a zero of f.



$\mathbb{P}^n$
ζ_1
ζ_0
$g = q_0$
$f = q_1$

The idea is to follow this curve numerically: partition $[0, 1]$ into $t_0 = 0, \ldots, t_k = 1$. Writing $q_i := q_{t_i}$, successively compute approximations z_i of ζ_{t_i} by Newton's method starting with $z_0 := \zeta$. More specifically, compute

$$z_{i+1} := N_{q_{i+1}}(z_i).$$



$\mathbb{P}^n$
ζ_1
$\zeta_{t_{i+1}}$
ζ_{t_i}
z_{i+1}
z_i
ζ_0
$g = q_0$
q_{t_i}
$q_{t_{i+1}}$
$f = q_1$

The Bézout series set up the main properties of this algorithmic scheme and put in place the theoretical tools used today in its study.

I won't give details of what these tools are or how they are used in recent work. I will instead limit my exposition to the description of the state-of-the-art in the subject.

The Bézout series set up the main properties of this algorithmic scheme and put in place the theoretical tools used today in its study.

I won't give details of what these tools are or how they are used in recent work. I will instead limit my exposition to the description of the state-of-the-art in the subject.

Two issues neglected in my exposition above:

(1) How to choose the initial pair (g, ζ)?

The Bézout series set up the main properties of this algorithmic scheme and put in place the theoretical tools used today in its study.

I won't give details of what these tools are or how they are used in recent work. I will instead limit my exposition to the description of the state-of-the-art in the subject.

Two issues neglected in my exposition above:

(1) How to choose the initial pair (g, ζ)?

(2) How large should $d(q_{i+1}, q_i)$ be?

How large should $d(q_{i+1}, q_i)$ be?

- We compute t_{i+1} adaptively from t_i such that

$$d(q_{i+1}, q_i) = \frac{0.0085}{D^{3/2}\, \mu_{\mathrm{norm}}^2(q_i, z_i)}.$$

How large should $d(q_{i+1}, q_i)$ be?

- We compute t_{i+1} adaptively from t_i such that

$$d(q_{i+1}, q_i) = \frac{0.0085}{D^{3/2}\, \mu_{\text{norm}}^2(q_i, z_i)}.$$

- Denote by $K(f, g, \zeta)$ the number K of iterations performed to follow the curve.

How large should $d(q_{i+1}, q_i)$ be?

- We compute t_{i+1} adaptively from t_i such that

$$d(q_{i+1}, q_i) = \frac{0.0085}{D^{3/2}\,\mu_{\mathrm{norm}}^2(q_i, z_i)}.$$

- Denote by $K(f, g, \zeta)$ the number K of iterations performed to follow the curve.

“Bézout VI” (**M.S.**, *Found. Comput. Math. 2009*)

For all i, z_i is an approximate zero of q_i. In particular z_K is an approximate zero of f. Moreover,

$$K(f, g, \zeta) \leq 217\, D^{3/2}\, d(f, g) \int_0^1 \mu_{\mathrm{norm}}^2(q_\tau, \zeta_\tau)\, d\tau.$$

Here $\tau \in [0, 1]$ is a ratio of angles and not of Euclidean distances.

This result relates to cost in a clear manner. Each Newton step takes $\mathcal{O}(N)$ arithemetic operations. Therefore, the total number of such operations performed along the homotopy is $\mathcal{O}(N\,K(f,g,\zeta))$.

This result relates to cost in a clear manner. Each Newton step takes $\mathcal{O}(N)$ arithemetic operations. Therefore, the total number of such operations performed along the homotopy is $\mathcal{O}(N\,K(f,g,\zeta))$.

It has been used in the following:

This result relates to cost in a clear manner. Each Newton step takes $\mathcal{O}(N)$ arithemetic operations. Therefore, the total number of such operations performed along the homotopy is $\mathcal{O}(N\,K(f,g,\zeta))$.

It has been used in the following:

(1) a randomized algorithm computing approximate zeros in average randomized polynomial time: $\mathcal{O}(D^{3/2}nN^2)$
[C. Beltán – L.M. Pardo].

This result relates to cost in a clear manner. Each Newton step takes $\mathcal{O}(N)$ arithemetic operations. Therefore, the total number of such operations performed along the homotopy is $\mathcal{O}(N\,K(f,g,\zeta))$.

It has been used in the following:

(1) a randomized algorithm computing approximate zeros in average randomized polynomial time: $\mathcal{O}(D^{3/2}nN^2)$
[C. Beltán – L.M. Pardo].

(2) a deterministic algorithm working in near-polynomial time (average polynomial time for all but a few pairs (n, D) and average time $N^{\mathcal{O}(\log\log N)}$ on those pairs). [P. Bürgisser – F.C.].

Additional remarks:

- Projective Newton method introduced by Mike.

Additional remarks:

• Projective Newton method introduced by Mike.

• Several extensions of Newton method to more general systems (overdetermined, underdetermined, multihomogeneous, ...) studied by Mike, mostly in joint work with Jean-Pierre Dedieu.

Additional remarks:

• Projective Newton method introduced by Mike.

• Several extensions of Newton method to more general systems (overdetermined, underdetermined, multihomogeneous, . . .) studied by Mike, mostly in joint work with Jean-Pierre Dedieu.

• Back to the roots? [D. Armentano, **M.S.**]

Structural Complexity for Numerical Problems

An algorithm solving a problem provides —through its analysis— an upper bound on the resources necessary to solve this problem.

Structural Complexity for Numerical Problems

An algorithm solving a problem provides —through its analysis— an upper bound on the resources necessary to solve this problem.

To obtain lower bounds one needs instead to consider **all** algorithms solving the problem. Thus, the study of lower bounds demands a formal notion of algorithm at hand.

Structural Complexity for Numerical Problems

An algorithm solving a problem provides —through its analysis— an upper bound on the resources necessary to solve this problem.

To obtain lower bounds one needs instead to consider **all** algorithms solving the problem. Thus, the study of lower bounds demands a formal notion of algorithm at hand.

Classical complexity theory (as studied in Theoretical Computer Science) has the Turing machine for this notion. This is very useful for discrete computations but not so for numerical computations. A "continuous" complexity theory is needed in this context.

Structural Complexity for Numerical Problems

An algorithm solving a problem provides —through its analysis— an upper bound on the resources necessary to solve this problem.

To obtain lower bounds one needs instead to consider **all** algorithms solving the problem. Thus, the study of lower bounds demands a formal notion of algorithm at hand.

Classical complexity theory (as studied in Theoretical Computer Science) has the Turing machine for this notion. This is very useful for discrete computations but not so for numerical computations. A "continuous" complexity theory is needed in this context.

- L. Blum, **M.S.**, S. Smale. "On a theory of computation over the real numbers: NP-completeness, recursive functions and universal machines", *Bull. AMS*, 1989.

- Introduced the BSS-machine.

- Introduced the BSS-machine.
- Natural notions of deterministic cost and nondeterministic cost.

  Cost is, essentially, number of arithmetic operations and comparisons performed.

  Nondeterminism is a theoretical mode of computation that, instead of “finding” or “computing” the solution to a problem, simply “verifies” that a candidate solution is a solution indeeed.

- Introduced the BSS-machine.
- Natural notions of deterministic cost and nondeterministic cost.

  Cost is, essentially, number of arithmetic operations and comparisons performed.

  Nondeterminism is a theoretical mode of computation that, instead of "finding" or "computing" the solution to a problem, simply "verifies" that a candidate solution is a solution indeeed.

- Classes $\mathrm{P}_{\mathbb{R}}$ and $\mathrm{NP}_{\mathbb{R}}$ (and $\mathrm{P}_{\mathbb{C}}$ and $\mathrm{NP}_{\mathbb{C}}$).

- Introduced the BSS-machine.
- Natural notions of deterministic cost and nondeterministic cost.

  Cost is, essentially, number of arithmetic operations and comparisons performed.

  Nondeterminism is a theoretical mode of computation that, instead of "finding" or "computing" the solution to a problem, simply "verifies" that a candidate solution is a solution indeeed.

- Classes $\mathrm{P}_{\mathbb{R}}$ and $\mathrm{NP}_{\mathbb{R}}$ (and $\mathrm{P}_{\mathbb{C}}$ and $\mathrm{NP}_{\mathbb{C}}$).

  A problem in $\mathrm{NP}_{\mathbb{R}}$.

  4FEAS   Given a polynomial f in $\mathbb{R}[X_1, \ldots, X_n]$ of degree 4, does there exist $\xi \in \mathbb{R}^n$ such that $f(\xi) = 0$?

- Introduced the BSS-machine.
- Natural notions of deterministic cost and nondeterministic cost.

  Cost is, essentially, number of arithmetic operations and comparisons performed.

  Nondeterminism is a theoretical mode of computation that, instead of "finding" or "computing" the solution to a problem, simply "verifies" that a candidate solution is a solution indeeed.

- Classes $\mathrm{P}_{\mathbb{R}}$ and $\mathrm{NP}_{\mathbb{R}}$ (and $\mathrm{P}_{\mathbb{C}}$ and $\mathrm{NP}_{\mathbb{C}}$).

  A problem in $\mathrm{NP}_{\mathbb{R}}$.

  4FEAS Given a polynomial f in $\mathbb{R}[X_1, \ldots, X_n]$ of degree 4, does there exist $\xi \in \mathbb{R}^n$ such that $f(\xi) = 0$?

  A problem in $\mathrm{NP}_{\mathbb{C}}$.

  QUAD Given $f_1, \ldots, f_m$ in $\mathbb{C}[X_1, \ldots, X_n]$ of degree 2, is there a $\xi \in \mathbb{C}^n$ such that $f_1(\xi) = \ldots = f_m(\xi) = 0$?

- Existence of natural $\mathrm{NP}_{\mathbb{R}}$-complete problems.

- Existence of natural $\mathrm{NP}_{\mathbb{R}}$-complete problems.

  A complete problem $\mathcal{P}$ in $\mathrm{NP}_{\mathbb{R}}$ is one such that, if $\mathcal{P} \in \mathrm{P}_{\mathbb{R}}$ then $\mathrm{P}_{\mathbb{R}} = \mathrm{NP}_{\mathbb{R}}$.

- Existence of natural $\mathrm{NP}_{\mathbb{R}}$-complete problems.

  A complete problem $\mathcal{P}$ in $\mathrm{NP}_{\mathbb{R}}$ is one such that, if $\mathcal{P} \in \mathrm{P}_{\mathbb{R}}$ then $\mathrm{P}_{\mathbb{R}} = \mathrm{NP}_{\mathbb{R}}$.

Explanation: All problems in $\mathrm{NP}_{\mathbb{R}}$ "reduce" to $\mathcal{P}$ (negligible overhead cost).

- Existence of natural $\mathrm{NP}_{\mathbb{R}}$-complete problems.

  A complete problem $\mathcal{P}$ in $\mathrm{NP}_{\mathbb{R}}$ is one such that, if $\mathcal{P} \in \mathrm{P}_{\mathbb{R}}$ then $\mathrm{P}_{\mathbb{R}} = \mathrm{NP}_{\mathbb{R}}$.

Explanation: All problems in $\mathrm{NP}_{\mathbb{R}}$ "reduce" to $\mathcal{P}$ (negligible overhead cost).

4FEAS is $\mathrm{NP}_{\mathbb{R}}$-complete

QUAD is $\mathrm{NP}_{\mathbb{C}}$-complete

- Existence of natural $\mathrm{NP}_{\mathbb{R}}$-complete problems.

  A complete problem $\mathcal{P}$ in $\mathrm{NP}_{\mathbb{R}}$ is one such that, if $\mathcal{P} \in \mathrm{P}_{\mathbb{R}}$ then $\mathrm{P}_{\mathbb{R}} = \mathrm{NP}_{\mathbb{R}}$.

Explanation: All problems in $\mathrm{NP}_{\mathbb{R}}$ "reduce" to $\mathcal{P}$ (negligible overhead cost).

4FEAS is $\mathrm{NP}_{\mathbb{R}}$-complete

QUAD is $\mathrm{NP}_{\mathbb{C}}$-complete

These results put focus on the problems 4FEAS and QUAD.

- Existence of natural $\mathrm{NP}_{\mathbb{R}}$-complete problems.

  A complete problem $\mathcal{P}$ in $\mathrm{NP}_{\mathbb{R}}$ is one such that, if $\mathcal{P} \in \mathrm{P}_{\mathbb{R}}$ then $\mathrm{P}_{\mathbb{R}} = \mathrm{NP}_{\mathbb{R}}$.

Explanation: All problems in $\mathrm{NP}_{\mathbb{R}}$ “reduce” to $\mathcal{P}$ (negligible overhead cost).

4FEAS is $\mathrm{NP}_{\mathbb{R}}$-complete

QUAD is $\mathrm{NP}_{\mathbb{C}}$-complete

These results put focus on the problems 4FEAS and QUAD.

Relations of QUAD and Smale’s 17th problem:

decision vs function problem

- Existence of natural $\mathrm{NP}_{\mathbb{R}}$-complete problems.

  A complete problem $\mathcal{P}$ in $\mathrm{NP}_{\mathbb{R}}$ is one such that, if $\mathcal{P} \in \mathrm{P}_{\mathbb{R}}$ then $\mathrm{P}_{\mathbb{R}} = \mathrm{NP}_{\mathbb{R}}$.

Explanation: All problems in $\mathrm{NP}_{\mathbb{R}}$ "reduce" to $\mathcal{P}$ (negligible overhead cost).

4FEAS is $\mathrm{NP}_{\mathbb{R}}$-complete

QUAD is $\mathrm{NP}_{\mathbb{C}}$-complete

These results put focus on the problems 4FEAS and QUAD.

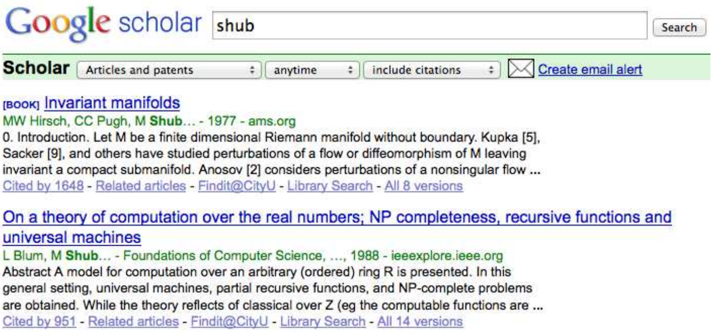
Relations of QUAD and Smale's 17th problem:

decision vs function problem

average-case vs worst-case

The BSS paper has had a tremendous impact in the work of a group of people who made its complexity theory the center of their research.

The BSS paper has had a tremendous impact in the work of a group of people who made its complexity theory the center of their research.



- F.C., **M.S.** “Generalized knapsack problems and fixed degree separations”, *Theoret. Comput. Sci.*, 1996.

• F.C., **M.S.** "Generalized knapsack problems and fixed degree separations", *Theoret. Comput. Sci.*, 1996.

For every $d \geq 1$

$$\mathrm{DTIME}(\mathcal{O}(n^d)) \neq \mathrm{NDTIME}(\mathcal{O}(n^d)).$$

Conditioning of Numerical Problems

$$\varphi : \mathbb{R}^n \to \mathbb{R}^m \qquad\qquad a \in \mathbb{R}^n$$

The condition number of a is the worst-case magnification in $\varphi(a)$ of small relative errors in a:

$$\mathrm{cond}^{\varphi}(a) := \lim_{\delta \to 0} \sup_{\mathrm{RelError}(a) \le \delta} \frac{\mathrm{RelError}(\varphi(a))}{\mathrm{RelError}(a)}.$$

Conditioning of Numerical Problems

$$\varphi : \mathbb{R}^n \to \mathbb{R}^m \qquad\qquad a \in \mathbb{R}^n$$

The condition number of a is the worst-case magnification in $\varphi(a)$ of small relative errors in a:

$$\mathrm{cond}^{\varphi}(a) := \lim_{\delta\to 0} \sup_{\mathrm{RelError}(a)\le\delta} \frac{\mathrm{RelError}(\varphi(a))}{\mathrm{RelError}(a)}.$$

- The condition number plays a key role in finite-precision analyses of algorithms.

Conditioning of Numerical Problems

$$\varphi : \mathbb{R}^n \to \mathbb{R}^m \qquad\qquad a \in \mathbb{R}^n$$

The condition number of a is the worst-case magnification in $\varphi(a)$ of small relative errors in a:

$$\mathrm{cond}^{\varphi}(a) := \lim_{\delta \to 0} \sup_{\mathrm{RelError}(a) \le \delta} \frac{\mathrm{RelError}(\varphi(a))}{\mathrm{RelError}(a)}.$$

- The condition number plays a key role in finite-precision analyses of algorithms.
- For many problems φ the quantity $\mathrm{cond}^{\varphi}(a)$ can be characterized (or approximated) in a more friendly manner.

Conditioning of Numerical Problems

$$\varphi : \mathbb{R}^n \to \mathbb{R}^m \qquad\qquad a \in \mathbb{R}^n$$

The condition number of a is the worst-case magnification in $\varphi(a)$ of small relative errors in a:

$$\mathsf{cond}^{\varphi}(a) := \lim_{\delta\to 0} \sup_{\mathsf{RelError}(a)\leq\delta} \frac{\mathsf{RelError}(\varphi(a))}{\mathsf{RelError}(a)}.$$

- The condition number plays a key role in finite-precision analyses of algorithms.
- For many problems φ the quantity $\mathsf{cond}^{\varphi}(a)$ can be characterized (or approximated) in a more friendly manner.
- These characterizations have allowed, in many cases, to obtain estimates of the expectation $\mathbb{E}(\mathsf{cond}^{\varphi})$ with respect to a measure on $\mathbb{R}^n$.

Conditioning of Numerical Problems

$$\varphi : \mathbb{R}^n \to \mathbb{R}^m \qquad\qquad a \in \mathbb{R}^n$$

The condition number of a is the worst-case magnification in $\varphi(a)$ of small relative errors in a:

$$\mathrm{cond}^{\varphi}(a) := \lim_{\delta\to 0} \sup_{\mathrm{RelError}(a)\le\delta} \frac{\mathrm{RelError}(\varphi(a))}{\mathrm{RelError}(a)}.$$

- The condition number plays a key role in finite-precision analyses of algorithms.
- For many problems φ the quantity $\mathrm{cond}^{\varphi}(a)$ can be characterized (or approximated) in a more friendly manner.
- These characterizations have allowed, in many cases, to obtain estimates of the expectation $\mathbb{E}(\mathrm{cond}^{\varphi})$ with respect to a measure on $\mathbb{R}^n$.
- Condition numbers have also been used in estimates for the speed of convergence of iterative algorithms (complexity!).

Mike's first work in conditioning studies a notion of condition number obtained by replacing "worst-case perturbation" by "average perturbation." This is relevant for finite-precision analyses.

• N. Weiss, G. Wasikowski, H. Wozniakowski, **M.S.** "Average condition number for solving linear equations." *Linear Algebra Appl.*, 1986.

Mike's first work in conditioning studies a notion of condition number obtained by replacing "worst-case perturbation" by "average perturbation." This is relevant for finite-precision analyses.

- N. Weiss, G. Wasikowski, H. Wozniakowski, **M.S.** "Average condition number for solving linear equations." *Linear Algebra Appl.*, 1986.

Then attention turned to the relationship between condition and complexity. This relationship pervades the Bézout series.

For each of the zeros $\zeta_1, \ldots, \zeta_{\mathcal{D}}$ of a system f we have that f $\mu_{\mathrm{norm}}(f, \zeta_i)$ is a condition number in the sense above!

For each of the zeros $\zeta_1, \ldots, \zeta_{\mathcal{D}}$ of a system f we have that f $\mu_{\text{norm}}(f, \zeta_i)$ is a condition number in the sense above!

The problem is, the map system $\to$ zero is multivalued.
What should we define as the condition of input f?

For each of the zeros $\zeta_1, \ldots, \zeta_{\mathcal{D}}$ of a system f we have that f $\mu_{\text{norm}}(f, \zeta_i)$ is a condition number in the sense above!

The problem is, the map system $\to$ zero is multivalued. What should we define as the condition of input f?

In the Bézout series the answer to this problem is

$$\mu_{\max}(f) := \max_{i \leq \mathcal{D}} \mu_{\text{norm}}(f, \zeta_i).$$

For each of the zeros $\zeta_1, \dots, \zeta_{\mathcal{D}}$ of a system f we have that f $\mu_{\text{norm}}(f, \zeta_i)$ is a condition number in the sense above!

The problem is, the map system $\to$ zero is multivalued.
What should we define as the condition of input f?

In the Bézout series the answer to this problem is

$$\mu_{\max}(f) := \max_{i \le \mathcal{D}} \mu_{\text{norm}}(f, \zeta_i).$$

The main result in Bézout VI allows one to use instead

$$\mu_{\text{av}}(f) := \sqrt{\frac{1}{\mathcal{D}} \sum_{i \le \mathcal{D}} \mu^2_{\text{norm}}(f, \zeta_i)}.$$

This fact is, as we already pointed out, at the core of the recent advances towards a final solution to Smale's 17th problem.

A Unifying Theory?

c) For a decision problem $\mathbb{R}^\infty, \mathbb{R}^\infty_{yes}$ we say a problem instance $x \in \mathbb{R}^k$ is ill posed if $\forall \delta > 0$ $\exists\, y_1, y_2 \in \mathbb{R}^k$ with $\|x - y_i\| \le \delta\|x\| + \delta$ $\quad i = 1, 2$ and $y_1 \in \mathbb{R}^k_{yes}$ while $y_2 \in \mathbb{R}^k_{no}$. We denote the set of ill-posed problems by Ill and Ill_k for inputs of size k if confusion is likely.