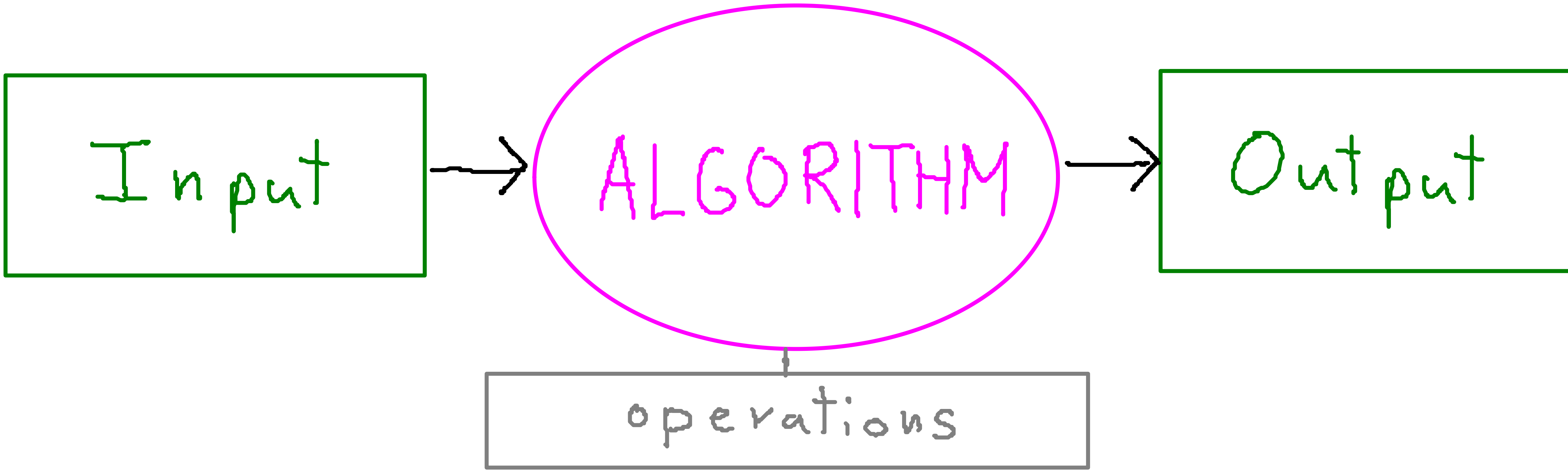
Beyond the Curse of Infinite Average Run-Time in Numerical Real Algebraic Geometry,

can the Adaptive Grid Method do the job?

Josué Tonelli-Cueto

SIAM AG21

Complexity of (Traditional) Algorithms



Worst-case form of complexity estimate:

$$\text{run-time}(\text{ALGORITHM}, \text{Input}) \leq f(\text{size}(\text{Input}))$$

⚠ sometimes size has several parameters
(e.g. #variables, degree...)

Complexity of Numerical Algorithms I

Inexact Input → Numerical ALGORITHM → Inexact Output

inexact operations

⚠ usual form of complexity fails!

ALL INPUTS OF THE SAME SIZE ARE EQUAL,
BUT SOME INPUTS ARE MORE EQUAL
THAN OTHERS

Complexity of Numerical Algorithms II

Condition-based complexity (Turing) (Goldstine, von Neumann)

cond(Input) : measures numerical sensitivity of Input

cond big ⇒ $\boxed{\text{small variations of Input} \longrightarrow \text{big variations of Output}}$

cond small ⇒ $\boxed{\text{'big' variations of Input} \longrightarrow \text{small variations of Output}}$

⚠ cond is a property of the computational problem, <u>not</u> of the algorithm!

Condition-based form of complexity estimates

$$\text{run-time}(\text{ALGORITHM}, \text{Input}) \leq g(\text{size}(\text{Input}), \text{cond}(\text{Input}))$$

Complexity of Numerical Algorithms III

Probabilistic complexity I (Goldstine, von Neumann)
(Smale) (Demmel)

Can we have a complexity estimate
of a numerical algorithm only depending on size?

Yes, if we randomize the Input

How do we randomize the Input?

We choose the probability distribution
depending on the context!

Statistical complexity might have been a better name

Complexity of Numerical Algorithms IV

Probabilistic complexity II (Goldstine, von Neumann) (Smale) (Demmel)

Probabilistic form of complexity estimates

$$\mathbb{P}_{input}\left[\text{run-time}(\text{ALGORITHM}, input) \geq t\right] \leq f(s,t)$$

where $\text{size}(input) \leq s$

... and if we are lucky

$$\mathbb{E}_{input}\left[\text{run-time}(\text{ALGORITHM}, input) \geq t\right] \leq f(s)$$

Complexity of Numerical Algorithms V

Smoothed complexity (Spielman, Teng)

Smoothed form of complexity estimates

$$\sup_{\substack{\text{Input} \\ \text{size(Input)}=s}} \mathbb{P}_{\text{noise}}\left[\text{run-time}(\text{ALGORITHM}, \text{Input} + \sigma\,\text{noise}) \geq t\right] \leq f(s,t)/\sigma$$

... and if we are lucky

$$\sup_{\substack{\text{Input} \\ \text{size(Input)}=s}} \mathbb{E}_{\text{noise}}\left[\text{run-time}(\text{ALGORITHM}, \text{Input} + \sigma\,\text{noise})\right] \leq f(s)/\sigma$$

- As $\sigma \to \infty$, we recover the average form
- As $\sigma \to 0$, we recover the worst-case form

Computing the Homology of Algebraic Sets I

Input

$f \in \mathcal{H}_d[q]$ q-tuple of homogeneous real polynomials in variables $X_0, \dots, X_n$

Output

$H_\bullet(Z_{\mathbb{P}}(f))$ Homology of projective real zero set

Size

$$N = \sum_{i=1}^{q} \binom{n+d_i}{n}$$

Total number of coefficients (including the zero ones)

$$D = \max d_i$$

Maximum degree

Desired complexity $\mathrm{poly}(q,D)^{\mathrm{poly}(n)} N$

Computing the Homology of Algebraic Sets II

Condition number (local version)

$$\kappa(f,x) := \frac{\|f\|_W}{\sqrt{\|f(x)\|^2 + \sigma_q(\Delta^{-1/2} D_x f)}}$$

where

$$\|f\|_W := \sqrt{\sum_{i=1}^{q} \sum_{|\alpha|=d_i} \binom{d_i}{\alpha}^{-1/2} f_{i,\alpha}^2} \quad \text{(Weyl norm)}$$

σ_q qth singular value

$\Delta := \mathrm{diag}(d_1, \ldots, d_q)$

$D_x f : T_x \mathbb{P}^n \to \mathbb{R}^q$ tangent map (up to signs!)

Computing the Homology of Algebraic Sets II

Condition number (global version)

$$\kappa(g) := \max_{x \in \mathbb{P}^n} \kappa(g, x)$$

Condition Number Theorem:

$$\kappa(g) = \frac{\|g\|_W}{\mathrm{dist}_W(g, \Sigma)}$$

where Σ is the discriminant variety

Cor. If $\frac{\|\tilde{g} - g\|_W}{\|g\|_W} \leq \frac{1}{\kappa(g)}$, then $H_\bullet(Z_{\mathbb{P}}(g)) = H_\bullet(Z_{\mathbb{P}}(\tilde{g}))$

Topology does not change!

Computing the Homology of Algebraic Sets III

Grid Method
(A bird's view)

(Cucker, Krick, Malajovich, Wschebor)
(Cucker, Krick, Shub)

1. Cover with a sufficiently fine grid $\mathbb{P}^n$
2. Select points in the grid sufficiently near to $\mathcal{Z}_{\mathbb{P}}(f)$ (meaning evaluation of f 'small')
3. Use TDA to compute $H_\bullet(\mathcal{Z}(f))$ out of the selected points (the sample)

Advantages:
- Numerically stable
- Parallelizable

For semialgebraic: (Bürgisser, Cucker, Lairez), (Bürgisser, Cucker, Tonelli-Cueto)

Computing the Homology of Algebraic Sets IV

Complexity of the Grid Method

Condition-based complexity estimate

$$\text{run-time}(\text{GRID}, f) \leq (n D \kappa(f))^{\mathcal{O}(n^2)}$$

Probabilistic complexity estimate

$$\mathbb{P}_f\left[\text{run-time}(\text{GRID}\ f) \geq (nD)^{\mathcal{O}(n^3)}\, t^{10n(n+1)}\right] \leq 1/t$$

where $f_i := \sum \binom{d_i}{\alpha}^{1/2} f_{i,\alpha} X^\alpha$ with i.i.d. $f_{i,\alpha} \sim N(0,1)$

also for smoothed & more general distributions (Ergür, Paouris, Rojas)

... unfortunately

$$\mathbb{E}_f\left[\text{run-time}(\text{GRID}\ f)\right] = \infty$$

Computing the Homology of Algebraic Sets V

the Curse

$$\mathbb{E}_f\, \mathcal{K}(f) = \infty$$

... so $\mathcal{K}(f)^{\mathcal{O}(n^2)}$ is really non-finite

Obs. $\mathbb{E}_f \log^\ell \mathcal{K}(f) < \infty$ for all $\ell \geq 1$

Are there numerical algorithms with condition-based complexity $(nD \log \mathcal{K}(f))^{\mathrm{poly}(n)}$?

This would be amazing!

Computing the Homology of Algebraic Sets VI

Adaptive Grid Method

- Like the grid method, but having a non-uniform grid
- Local mesh around x controlled by $\mathcal{K}(g,x)$
- More details in my MEGA21 talk (only the zero dimensional case)

For obtaining sample (not homology), the condition-based complexity is:

$$(nD)^{\mathcal{O}(n^2)} \mathbb{E}_{x \in \mathbb{P}^n} \mathcal{K}(g,x)^n$$

Computing the Homology of Algebraic Sets VII

where...

$$\mathbb{E}_f \mathbb{E}_{x \in \mathbb{P}^n} \kappa(f, x)^n \leq (nD)^{\mathcal{O}(n^3)}$$

is FINITE!

Unfortunately, postprocessing the sample brings expected complexity back to ∞,

can we scape this situation?

Better analysis?

New algorithms?

Only time will say...

Eskerrik asko

zure arretagatik!