Condition Numbers & Probability for explaining Algorithms

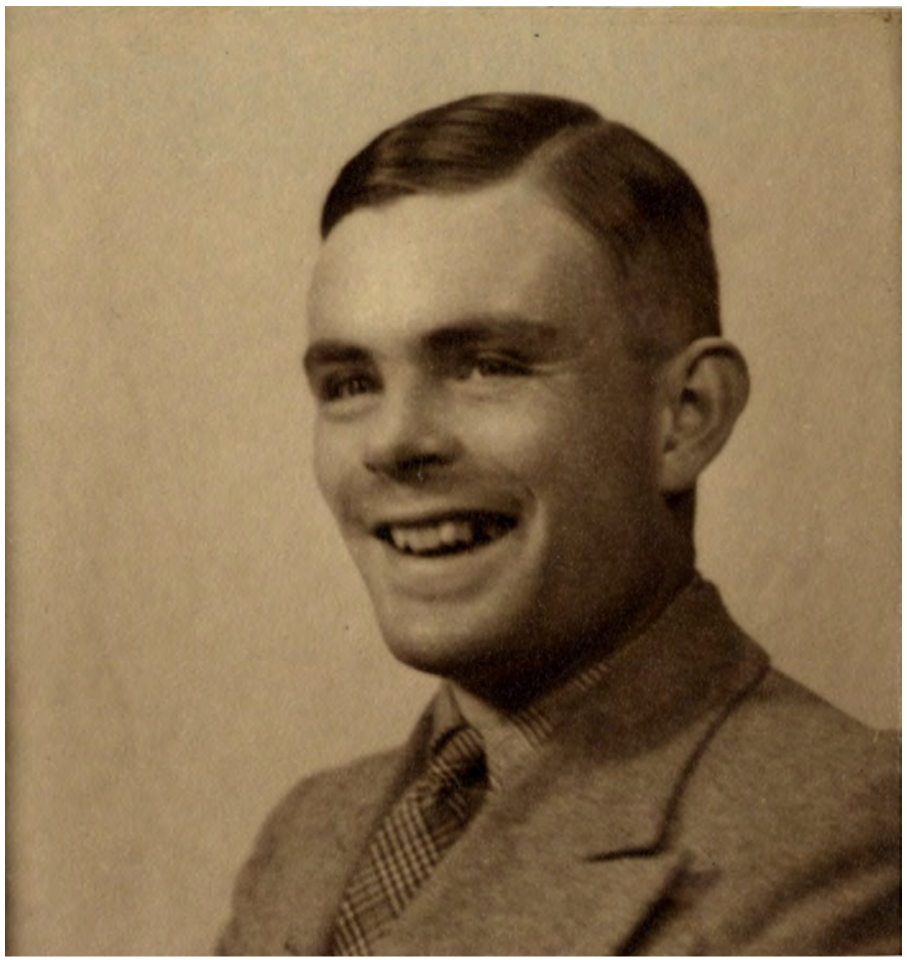
Josué Tonelli-Cueto

The University of Texas at San Antonio

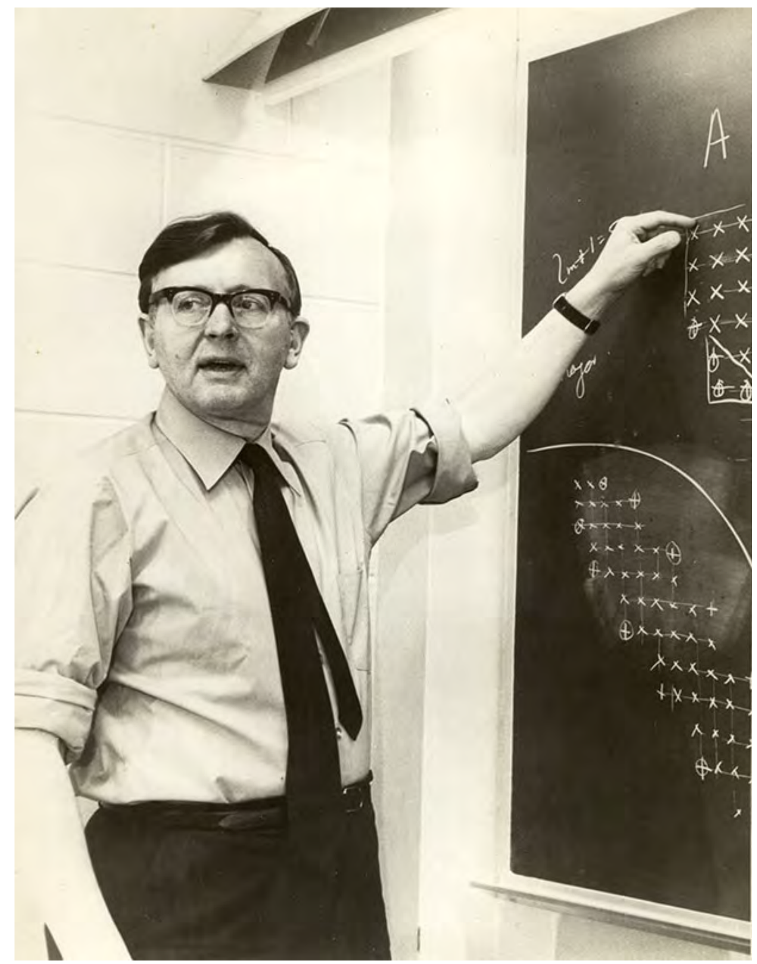
CodEx Seminar

A Foundational Myth

Turing vs. Wilkinson



Source: King's College [ATM/K/7/11]

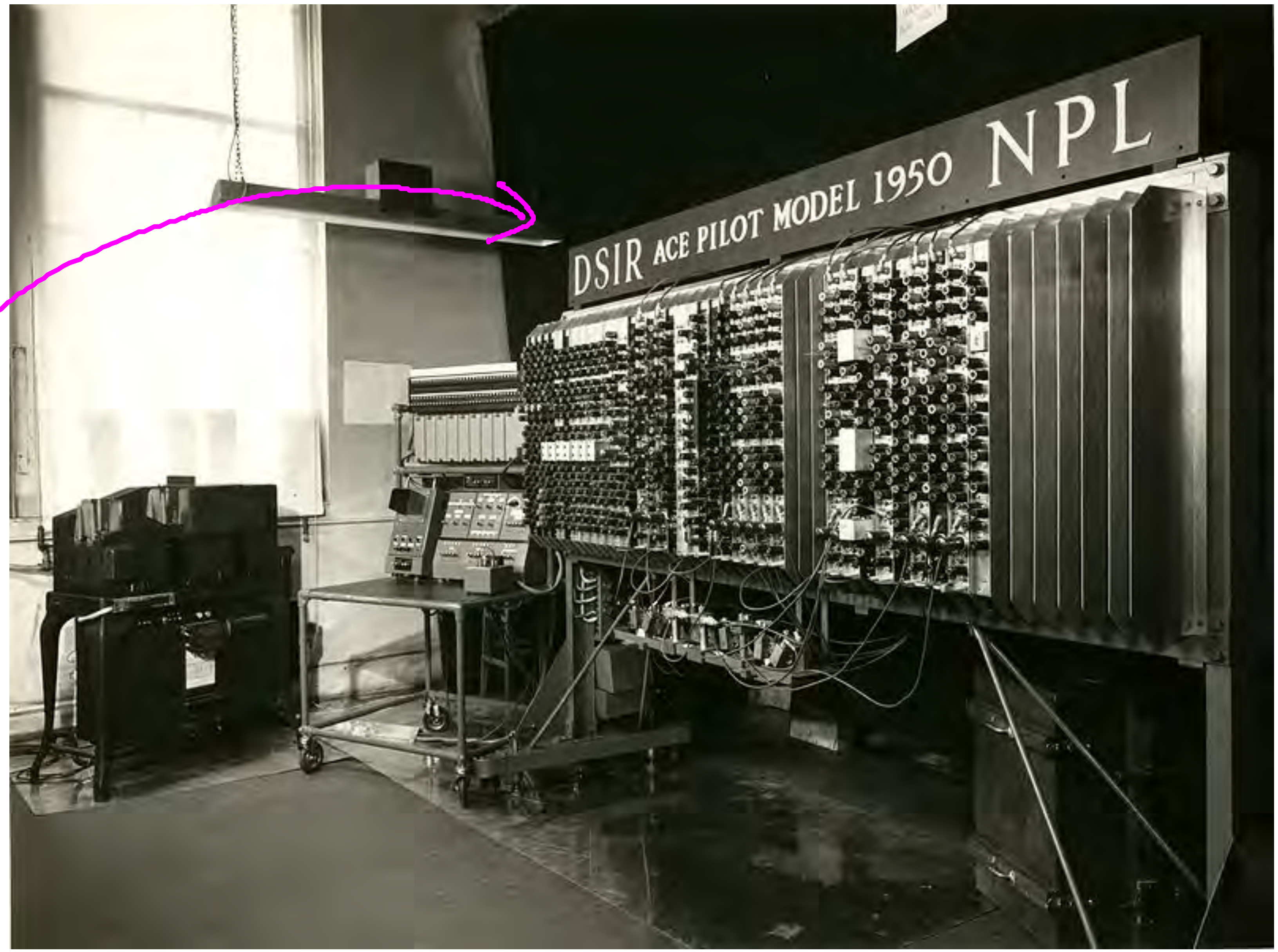


Source: U. of Manchester

Source: 1970 Turing Lecture

We are in 1946...
at the NPL in Manchester

The computer 4 years after!



Source: U. of Manchester

However, it happened that some time after my arrival, a system of 18 equations arrived in Mathematics Division and after talking around it for some time we finally decided to abandon theorizing and to solve it. A system of 18 is surprisingly formidable, even when one has had previous experience with 12, and we accordingly decided on a joint effort.

Wilkinson, 1970 Turing Lecture

Gaussian Elimination will not work!

Source: Beryl Turing & King's College

It will work! Let's do it with complete pivoting.

Source: U. of Manchester

And it succeeded!

I suppose this must be regarded as a defeat for Turing since he, at that time, was a keener adherent than any of the rest of us to the pessimistic school.

ROUNDING-OFF ERRORS IN MATRIX PROCESSES

By A. M. TURING

(*National Physical Laboratory, Teddington, Middlesex*)

[Received 4 November 1947]

The second round undoubtedly went to Turing!

Wilkinson, 1970 Turing Lecture

Why

do some algorithms
perform better than predicted?

Not an isolated phenomenon: the Simplex Method

Linear Programming

$$\max \; c^T x$$
$$\text{s.t. } Ax \le b$$

Problem

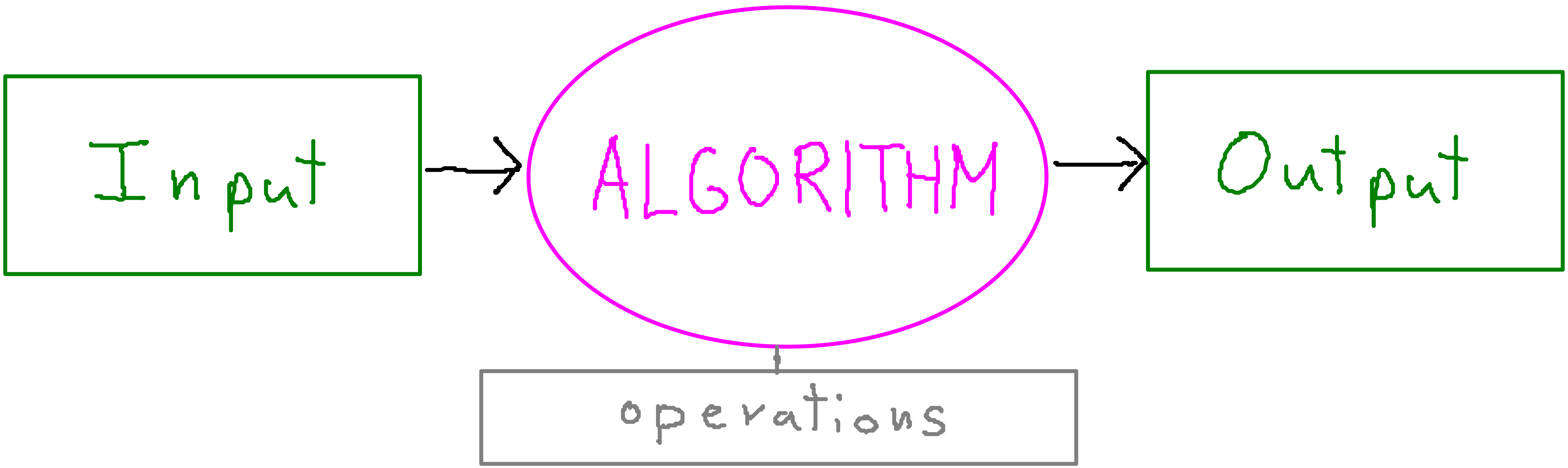
Danzig (1947)
Simplex Method

Very efficient in practice,
but... why?

Spielman & Teng (2001)
Justify Simplex Method using smoothed analysis

Complexity of Algorithms

Complexity of (Traditional) Algorithms

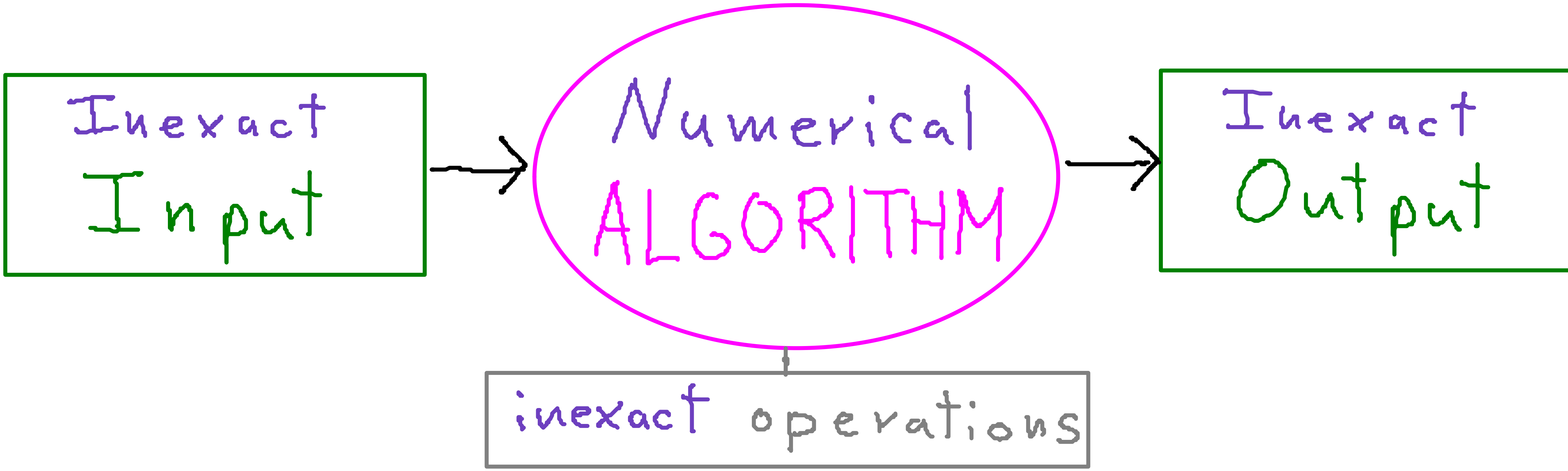


Worst-case form of complexity estimate:

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

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

Complexity of Numerical Algorithms



⚠ usual form of complexity fails!

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

Condition Numbers

(Turing) (Goldstine, von Neumann)

cond(Input):

measure of 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!

Turing Condition Number

$$A \in \mathbb{C}^{n \times n}$$

$$\mathrm{cond}(A) := \|A\| \, \|A^{-1}\|$$

Linear System: $Ax = b$

$$\text{rel-error}(x) \lesssim \mathrm{cond}(A) \max\{\text{rel-error}(A), \text{rel-error}(b)\}$$

Condition-based Complexity

(Turing) (Goldstine, von Neumann)

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

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

Randomize your Input

(Goldstine & von Neumann) (Smale) (Demmel)

Random Input → Probabilistic Complexity

How do we randomize the Input?

Choice depends on the context!

Probabilistic Complexity

(Goldstine & von Neumann) (Smale) (Demmel)

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

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

...and if we are lucky

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

Smoothed Complexity

(Spielman & Teng)

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

... and if we are lucky

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

Why Smoothed is better?

Worst-case form of complexity estimate

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

$$\Big\uparrow \ \sigma \to 0$$

Smoothed form of complexity estimates

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

$$\Big\downarrow \ \sigma \to \infty$$

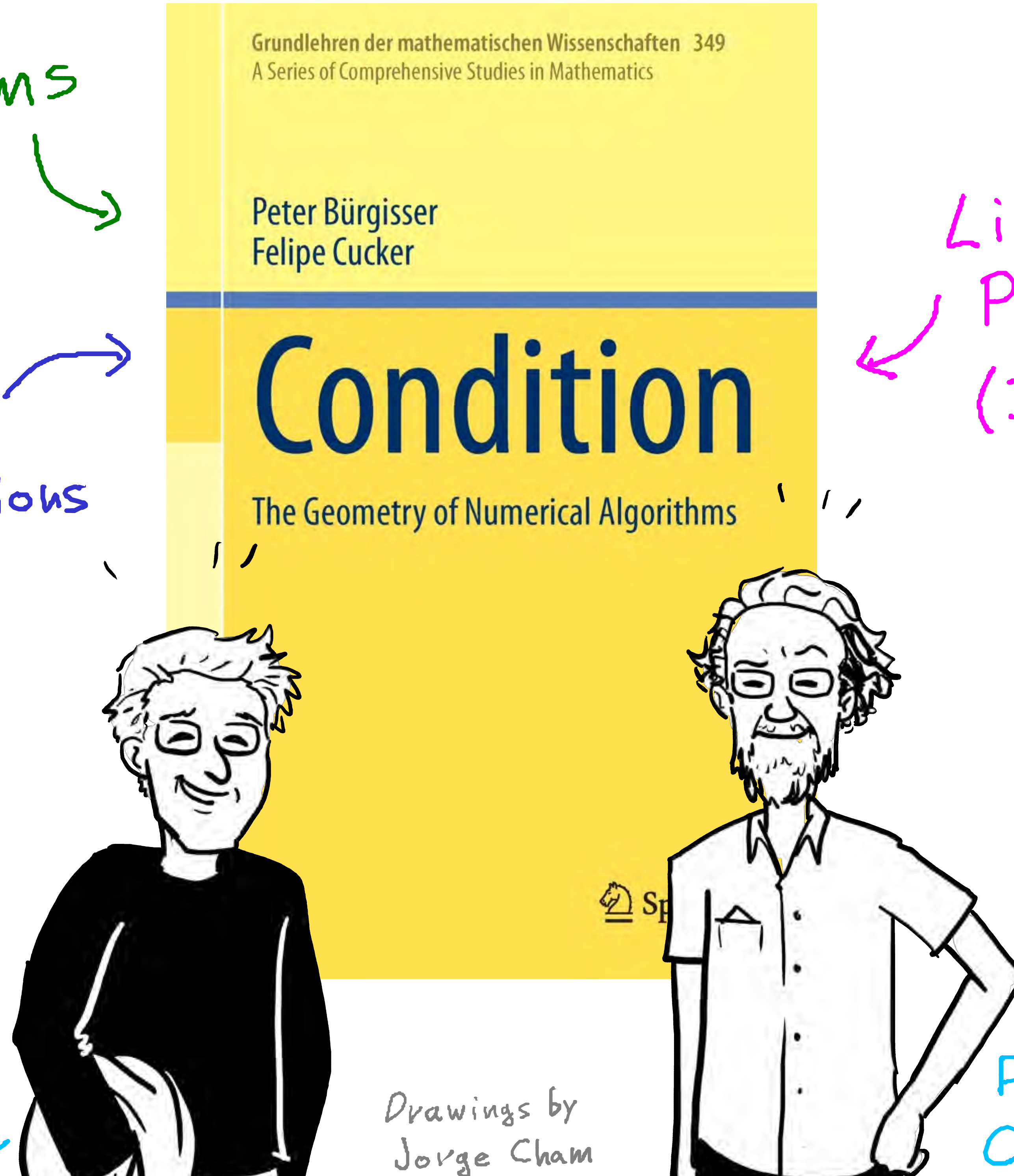
Probabilistic form of complexity estimates

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

Where to find all the details?

Linear Systems

Systems of Polynomial Equations



Linear Programming (Interior Point Method)

Peter Bürgisser

Drawings by Jorge Cham

Felipe Cucker

A Case Study

of the Framework in Action:

the DESCARTES Solver

for finding real roots

of real univariate polynomials

Joint work of

Elias Tsigaridas

Josué Tonelli-Cueto

Alperen A. Ergür

Photo while working on this project

Real Root Isolation I: The Problem

INPUT:

$g \in \mathbb{Z}[X]$

OUTPUT:

Intervals $J_1, \dots, J_k$ s.t.

0) $J_i = (a_i, b_i)$ with $a_i, b_i \in \mathbb{Q}$

1) $Z(g) \cap \mathbb{R} \subseteq \bigcup_{i=1}^{k} J_i$

2) $\forall i, \# Z(g) \cap J_i = 1$

INPUT SIZE PARAMETERS:

d : degree of g

τ : bit-size of coefficients of g

MEASURE OF RUN-TIME

Bit complexity

We can also handle continuous inputs!

Descartes Solver I: Rule of Signs

$V(g) :=$ # sign variations of $g_0, g_1, \ldots$

THM (Descartes' rule of signs)

$$\#Z(g, \mathbb{R}_>) \leq V(g)$$

Moreover,

$$V(g) \leq 1 \Rightarrow \text{Equality}$$

Portrait by Frans Hal
Source: WikiMedia Commons

COR

$$\#Z(g,(a,b)) \leq V(g,(a,b)) := V\left((X+1)^d \, g\left(\frac{bX+a}{X+1}\right)\right)$$

$(0,\infty) \to (a,b)$
bijection

Descartes Solver II: Rule of Signs in Action

$(0,\infty)$

$g = 2X^3 - 9X^2 + 12X - 6$

$(x+1)^3 g\left(\frac{x}{x+1}\right)$

$g(x+1)$

$(0,1)$

$-6 - 6X - 3X^2 - X^3$

$(1,\infty)$

$-1 - 3X^2 + 2X^3$

$(1,2)$

$-1 - 3X - 6X^2 - 2X^3$

$(2,\infty)$

$-2 + 3X^2 + 2X^3$

$(2,3)$

$-2 - 6X - 3X^2 + 3X^2$

$(3,\infty)$

$3 + 12X + 9X^2 + 2X^3$

Real Roots of g:

2.677650698804...

Descartes Solver III:
The Descartes' Oracle

1) Overcounting: $\#Z(g,J) \leq V(g,J)$

2) Exactness I: $V(g,J) \leq 1 \Rightarrow$ Equality

3) Exactness II:

$$\#Z(g, \mathbb{D}(m(J), cw(J))) \leq K \Rightarrow V(g,J) \leq K$$

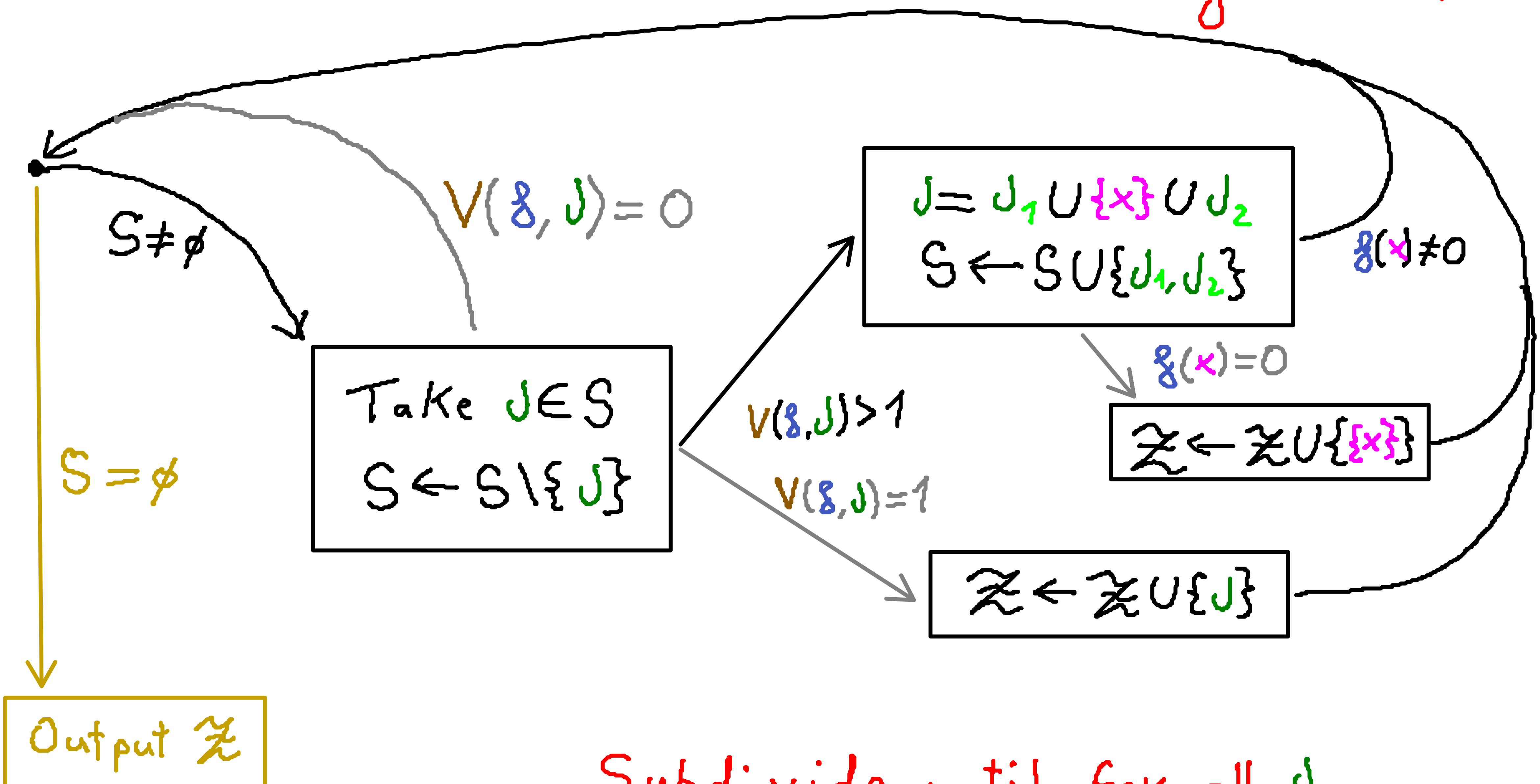
Obreshkoff's Thm: Descartes sees the complex roots around!

4) Subadditivity:

$$\dot{\bigcup} J_i \subseteq J \Rightarrow \sum V(g,J_i) \leq V(g,J)$$

Descartes Solver IV

The Algorithm



Subdivide until for all J,

$V(f, J) \leq 1$!

Real Root Isolation II: The State of the Art

Sturm Solver	$\tilde{O}_B(d^4\tau^2)$
Descartes Solver	$\tilde{O}_B(d^4\tau^2)$
ANewDsc (Sagraloff & Mehlhorn; 2016)	$\tilde{O}_B(d^3+d^2\tau)$
Pan's Algorithm (Pan; 2002)	$\tilde{O}_B(d^2\tau)$

Q: Can we beat the champion?

Real Root Isolation III: What do we wish?

$$\tilde{O}_B(d\tau)$$

We wish to find real roots almost as fast as we read the polynomial!

Real Root Isolation IV:

Are we being pessimistic?

DESCARTES SOLVER

seems to behave faster in practice!

Why?

SPOILER:

DESCARTES is almost-optimal on average!

What do we mean?

Real Root Isolation V:
Beyond pessimism

$$\mathbb{E}_{\mathfrak{g}}\left\{\mathrm{cost}(\mathrm{SOLVER}, \mathfrak{g})^{\ell} \mid \text{bit-size}(\mathfrak{g}) \leq \tau, \deg(\mathfrak{g}) \leq d\right\}$$

What's a 'good' random model for $\mathfrak{g}$?

↑ Many choices of randomness 😱

Beyond pessimism I: Uniform Random Bit Polynomials & a Simple Main Theorem

$$F = \sum_{k=0}^{d} f_k X^k$$

s.t. $f_k \sim \mathcal{U}([-2^\tau, 2^\tau] \cap \mathbb{Z})$ independent

SIMPLE MAIN THM

$$\mathbb{E}\,\mathrm{cost}(\text{DESCARTES}, F) = \widetilde{\mathcal{O}}_B(d^2 + d\tau)$$

On average, DESCARTES is almost optimal!

Beyond pessimism II: Random Bit Polynomials

$$f = \sum_{k=0}^{d} f_k X^k \in \mathbb{Z}[X]$$

s.t. f_k independent

bit-size of f:

$$\tau(f) := \min\{\tau \mid \forall k, \mathbb{P}(|f_k| \leq 2^{\tau}) = 1\}$$

weight of f:

No middle indexes!

$$w(f) := \max\{\mathbb{P}(f_k = c) \mid c \in \mathbb{R}, k \in \{0, d\}\}$$

uniformity of f: $u(f) := \ln\left(w(f)\left(1 + 2^{\tau(f)+1}\right)\right)$

Beyond pessimism III: MAIN THEOREM

MAIN THM

$$\mathbb{E}\,\mathrm{cost}(\text{DESCARTES}, f) = \tilde{\mathcal{O}}_B\left(d^2 + d\tau\right)\left(1 + u(f)\right)^4$$

Note: f uniform $\Rightarrow u(f) = 0$

Claim: For many cases, $u(f) = \mathcal{O}(1)$

IF $\tau = \Omega(d)$, almost like reading!

On average, DESCARTES is almost optimal!

Beyond pessimism IV: Examples of Random Bit Polynomials I

- Support control $\{0,d\} \subseteq A$

  $$f = \sum_{k \in A} f_k X^k \quad \text{with } f_k \sim \mathcal{U}([-2^\tau, 2^\tau] \cap \mathbb{Z})$$

  ... then $u(f) = 0$

- Sign control $\sigma \in \{-1,+1\}^{\{0,\ldots,d\}}$

  $$f = \sum_{k=1}^{d} f_k X^k \quad \text{with } f_k \sim \mathcal{U}(\sigma_k([1,2^\tau] \cap \mathbb{N}))$$

  ... then $u(f) \leq \ln 3$

Beyond pessimism V:
Examples of Random Bit Polynomials II

- Exact bitsize

$$F = \sum_{k=1}^{d} f_k X^k \quad \text{with} \quad f_k \sim \mathcal{U}(\{n \in \mathbb{Z} \mid \lfloor \log n \rfloor = \tau\})$$

$$\ldots \text{ then } u(F) \leq \ln 3$$

+ their combinations

Our random model is flexible!

Beyond pessimism $\boxed{V}$: Smoothed case included!

$f = \sum_{k=1}^{d} f_k X^k$ random bit polynomial

$g = \sum_{k=1}^{d} g_k X^k$ fix polynomial of entries of size τ

$\sigma \in \mathbb{Z} \setminus \{0\}$

Then:

$f_\sigma = g + \sigma f$ random bit polynomial

& $u(f_\sigma) \leq 1 + u(f) + \max\{\tau - \tau(f), \tau(\sigma)\}$

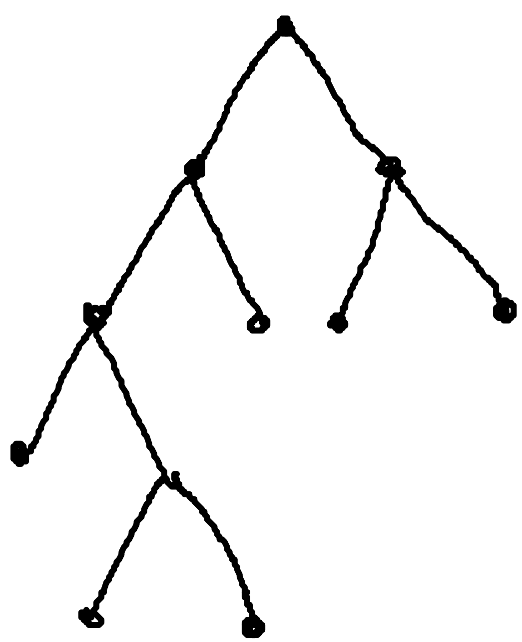
Summing Up:

Descartes

is almost-optimal on average!

The Ingredients of the Analysis 0: Descartes' tree

$\mathcal{T}(g, I)$



run-time of DESCARTES(g, I)

$\updownarrow$

size of $\mathcal{T}(g, I)$

$\lessapprox$

depth$(\mathcal{T}(g, I))$ width$(\mathcal{T}(g, I))$

We only need to control the size of subdiv. tree!

The Ingredients of the Analysis I: Condition Numbers

$$C(g) := \max_{x \in [-1,1]} \frac{\sum_{k=0}^{d} |g_k|}{\max\{|g(x)|, |g'(x)|/d\}}$$

$C(g) = \infty \Leftrightarrow g$ has a singular root in $[-1,1]$

Upper bounds on $C(g)$

→ Lower bounds for root separation of g

→ Upper bounds for depth of Descartes' tree

The Ingredients of the Analysis II: Bounds for Number of Complex Roots

We only care about nearby roots!

Upper bounds for

\# complex roots of g around $[-1,1]$

↓

Upper bounds

for width of Descartes' Tree

The Ingredients of the Analysis III: Probabilistic Toolbox

Ball's smoothing:

$x \in \mathbb{Z}^N$ discrete random variable

$y \in \mathbb{R}^N$ s.t. $y_i \sim \mathcal{U}((-1/2, 1/2))$ i.i.d.

Then: $x + y$ continuous random var.

We can use our old cont. toolbox!

⚠ I am omitting a lot of technical details.

TAKE HOME MESSAGE:

to EXPLAIN
the SUCCESS of some ALGORITHMS,
we need
CONDITION NUMBERS & PROBABILITY
to avoid PESSIMISTIC ESTIMATES

Eskerrik Asko

zure arretagatik!

Transl.: Thank you for your attention!