

CONDITION NUMBERS
& PROBABILITY
for EXPLAINING ALGORITHMS
in
COMPUTATIONAL
GEOMETRY

Josué TONELLI-CUETO
UT San Antonio

MINDS & CIS
SEMINAR SERIES
6/sep/2022

Fun Fact of the Day (6/SEP/2022)

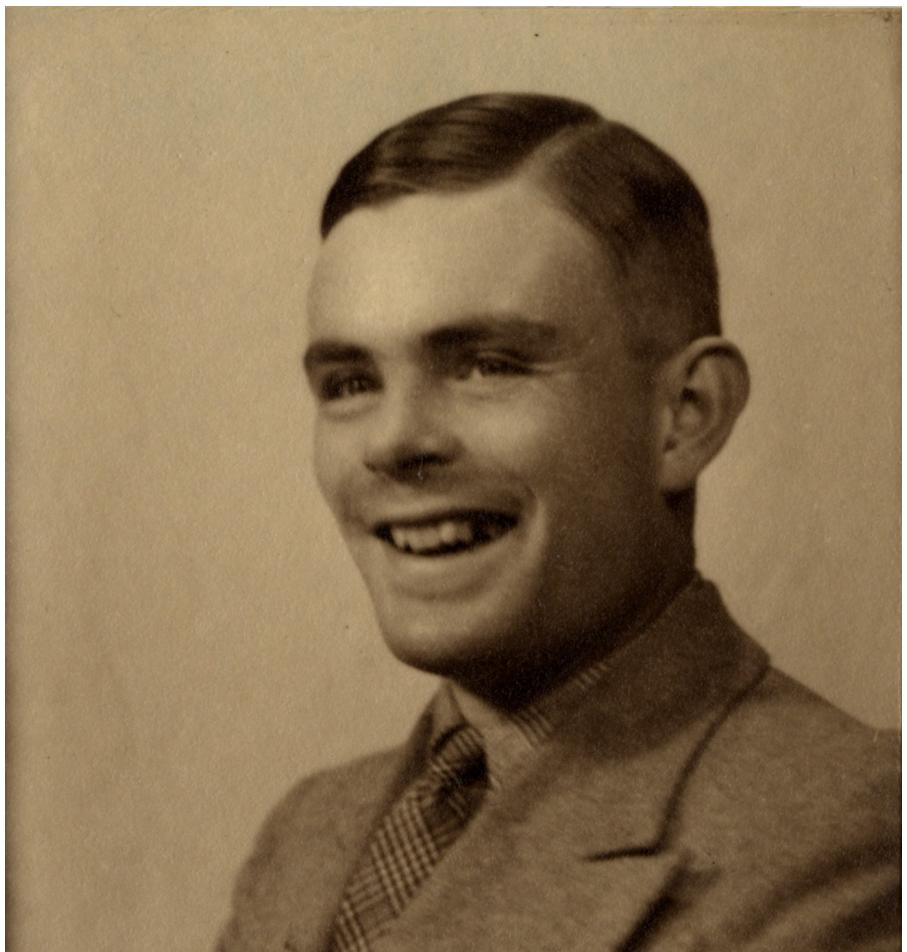


Source: Donna Haraway — Storytelling for Earthly Survival

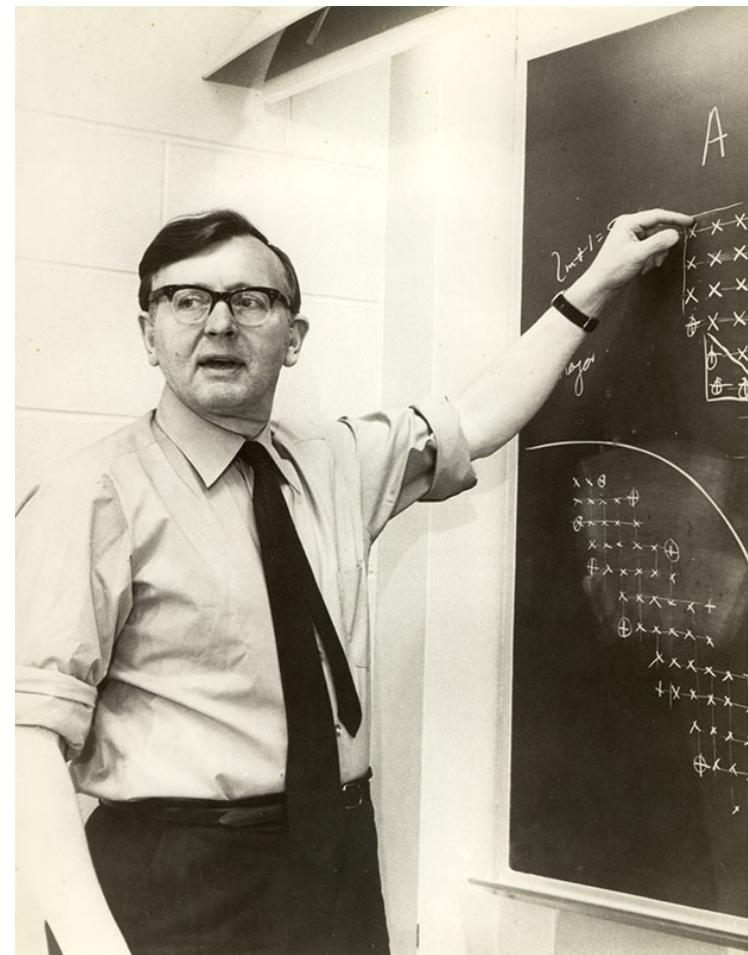
BIRTHDAY OF DONNA HARAWAY (78 years old)
— among other things, known by the Cyborg Manifesto

Why some algorithms
work a lot better
than predicted?

THE FOUNDATIONAL MYTH: Turing vs. Wilkinson



Source: King's College
[ATM/K/3/11]



Source: Higham's web

THE FOUNDATIONAL MYTH: Turing vs. Wilkinson

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. The operation was manned by Fox, Goodwin, Turing, and me, and we decided on Gaussian elimination with complete pivoting. Turing was not particularly enthusiastic, partly because he was not an experienced performer on a desk machine and partly because he was convinced that it would be a failure. History repeated itself remarkably closely. Again the system was mildly ill-conditioned, the last equation had a coefficient of order 10^{-4} (the original coefficients being of order unity) and the residuals were again of order 10^{-10} , that is of the size corresponding to the exact solution rounded to ten decimals. It is interesting that in connection with this example we subsequently performed one or two steps of what would now be called "iterative refinement," and this convinced us that the first solution had had almost six correct figures.

THE FOUNDATIONAL MYTH:

Turing vs. Wilkinson

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. However, I'm sure that this experience made quite an impression on him and set him thinking afresh on the problem of rounding errors in elimination processes. About a year later he produced his famous paper "Rounding-off errors in matrix processes" [1] which together with the paper of J. von Neumann and H. Goldstine [4] did a great deal to dispel the gloom. The second round undoubtedly went to Turing!

ROUNDING-OFF ERRORS IN MATRIX PROCESSES

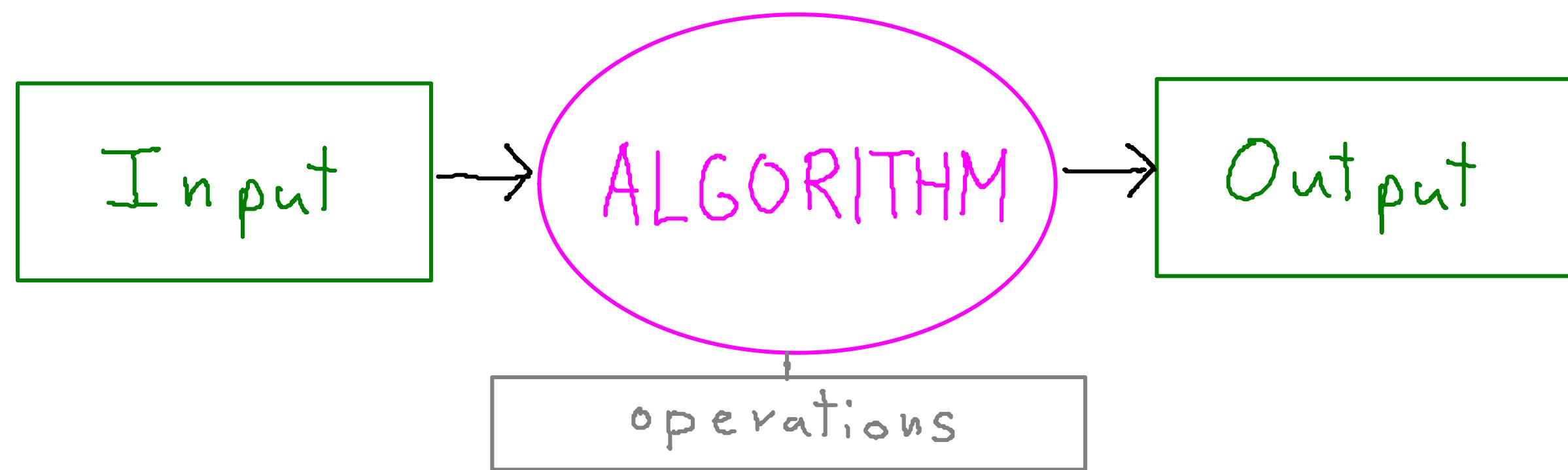
By A. M. TURING

(National Physical Laboratory, Teddington, Middlesex)

[Received 4 November 1947]

Wilkinson, 1970 Turing Lecture

Complexity of (Traditional) Algorithms



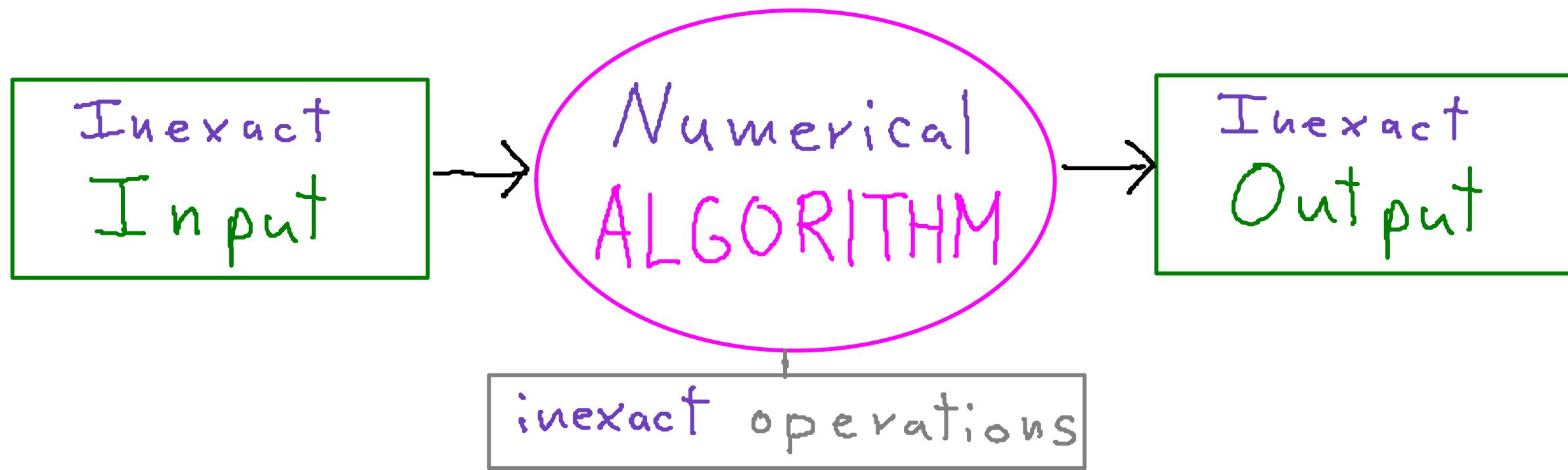
Worst-case form of complexity estimate:

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



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

Complexity of Numerical Algorithms I



⚠️ 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)

$\text{cond}(\text{Input})$: measures numerical sensitivity of Input

cond big \Rightarrow Small variations of Input
 \rightarrow big variations of Output

cond small \Rightarrow 'big' variations of Input
 \rightarrow small variations of Output

⚠ cond is a property of the computational problem,
not of the algorithm!

Complexity of Numerical Algorithms III

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

Condition-based form of complexity estimates

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

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

Complexity of Numerical Algorithms II

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 V

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

Probabilistic form of complexity estimates

$$\Pr_{\text{input}} \left[\text{run-time}(\text{ALGORITHM}, \text{input}) \geq t \right] \leq g(s, t)$$

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

... and if we are lucky

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

Complexity of (Numerical) Algorithms VI

Smoothed complexity] (Spielman, Teng)

Smoothed form of complexity estimates

$$\sup_{\substack{\text{Input} \\ \text{size(Input)}=s}} P_{\text{noise}} \left[\text{runtime}(\text{ALGORITHM}, \text{Input} + \sigma \text{noise}) \geq t \right] \leq g(s, t, \sigma)$$

... and if we are lucky

$$\sup_{\substack{\text{Input} \\ \text{size(Input)}=s}} E_{\text{noise}} \left[\text{runtime}(\text{ALGORITHM}, \text{Input} + \sigma \text{noise}) \right] \leq g(s, \sigma)$$

Complexity of (Numerical) Algorithms VI

Smoothed complexity II (Spielman, Teng)

Worst-case form of complexity estimate

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

$$\uparrow \sigma \rightarrow 0$$

Smoothed form of complexity estimates

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

$$\downarrow \sigma \rightarrow \infty$$

Probabilistic form of complexity estimates

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

Complexity of (Numerical) Algorithms

Success stories

Solving Linear Equations

See any intro to numerical analysis/random matrix theory

Linear Programming

Condition Goffin, Renegar, Cheng, Cucker, Peñar, ...

Prob./Smoothed Bürgisser, Cucker, Lotz,
Dunagan, Spielman, Teng...

Solving Polynomial Systems

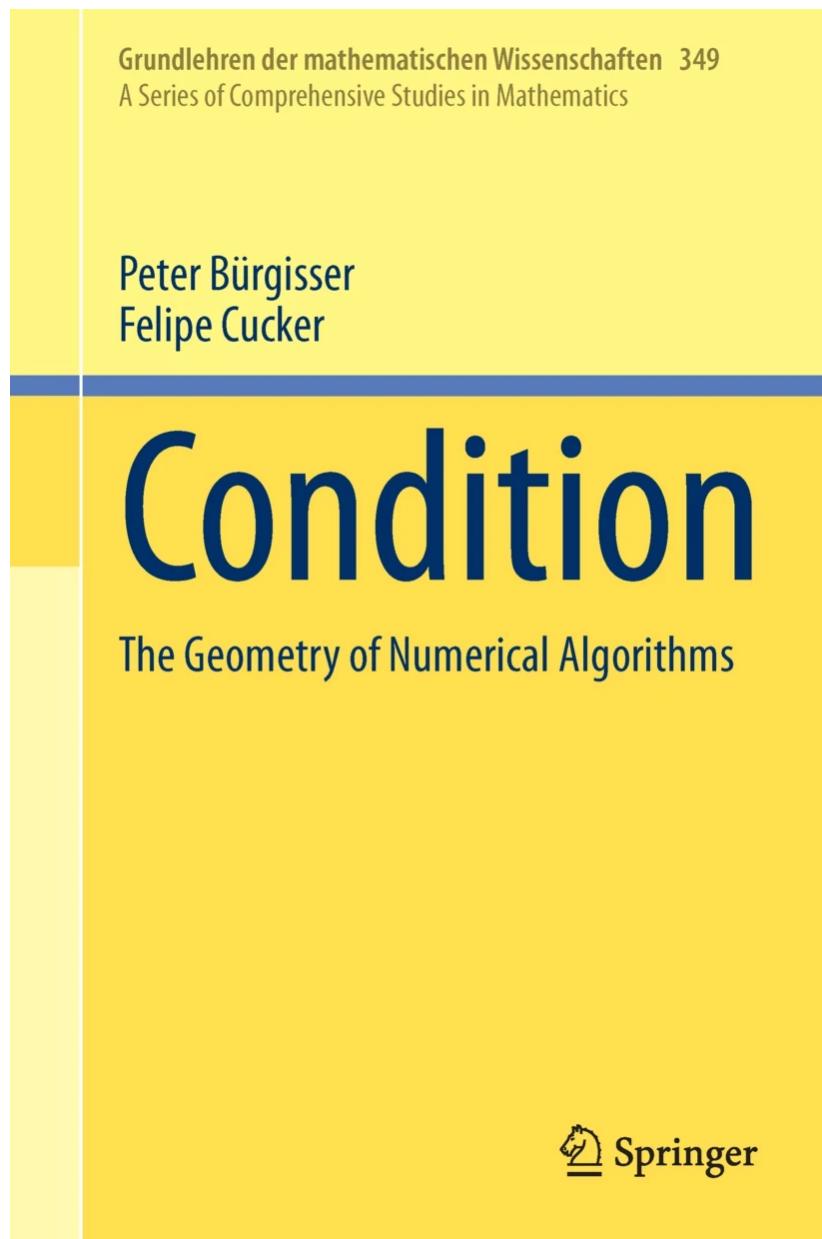
Smale's 17th Problem

Beltráñ, Pardo, Bürgisser, Cucker, Lairez...

...

Complexity of (Numerical) Algorithms IX

A good introduction...



THE FRAMEWORK IN ACTION I

the
PLANTINGA-VEGTER
algorithm

Joint work with F. Cucker & A.A. Ergür
Plus extra work with E. Tsigaridas



Photo while working on another project

An algorithm for Visualizing Implicit Curves & Surfaces

Eurographics Symposium on Geometry Processing (2004)
R. Scopigno, D. Zorin, (Editors)

Isotopic Approximation of Implicit Curves and Surfaces

Simon Plantinga and Gert Vegter

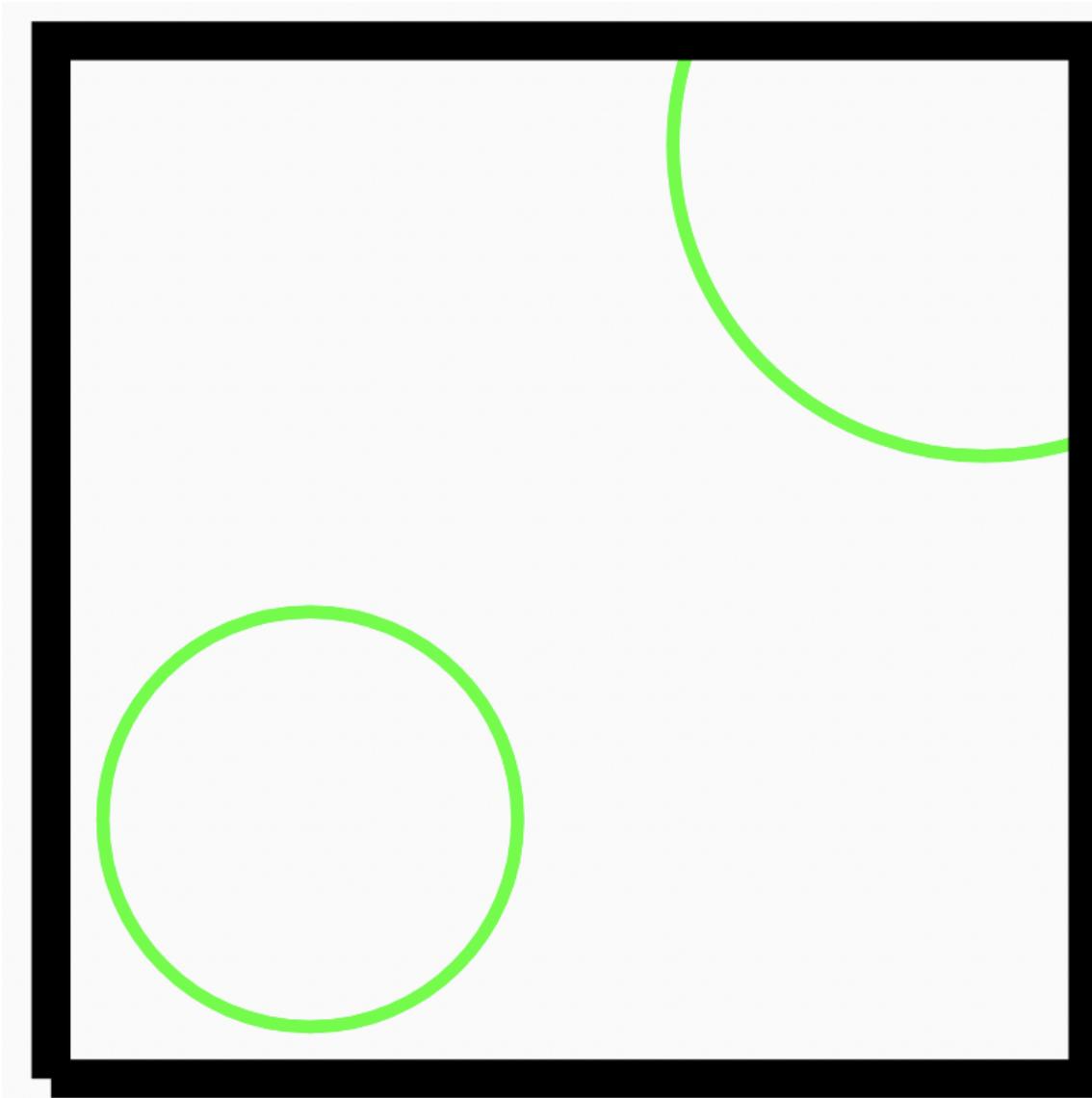
Institute for Mathematics and Computing Science
University of Groningen
simon@cs.rug.nl gert@cs.rug.nl



Extended to hypersurfaces by Galehouse

PV Algorithm in Action I

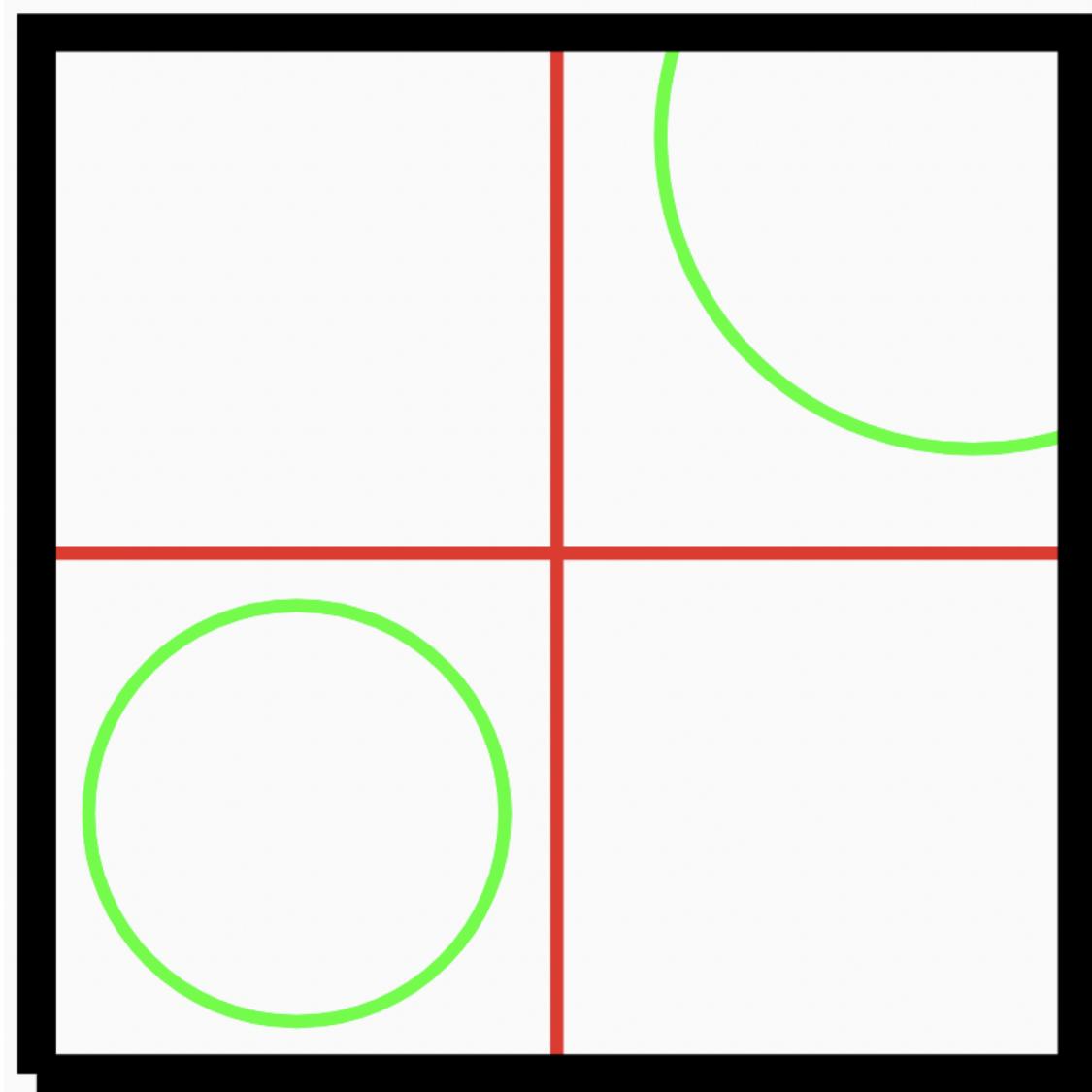
$$g = (x^2 + y^2)^2 - 6(x^3 + x^2y + xy^2 + y^3) \\ - 34(x^2 + y^2) - 320xy + 376(x+y) + 3128$$



[-10, 10]

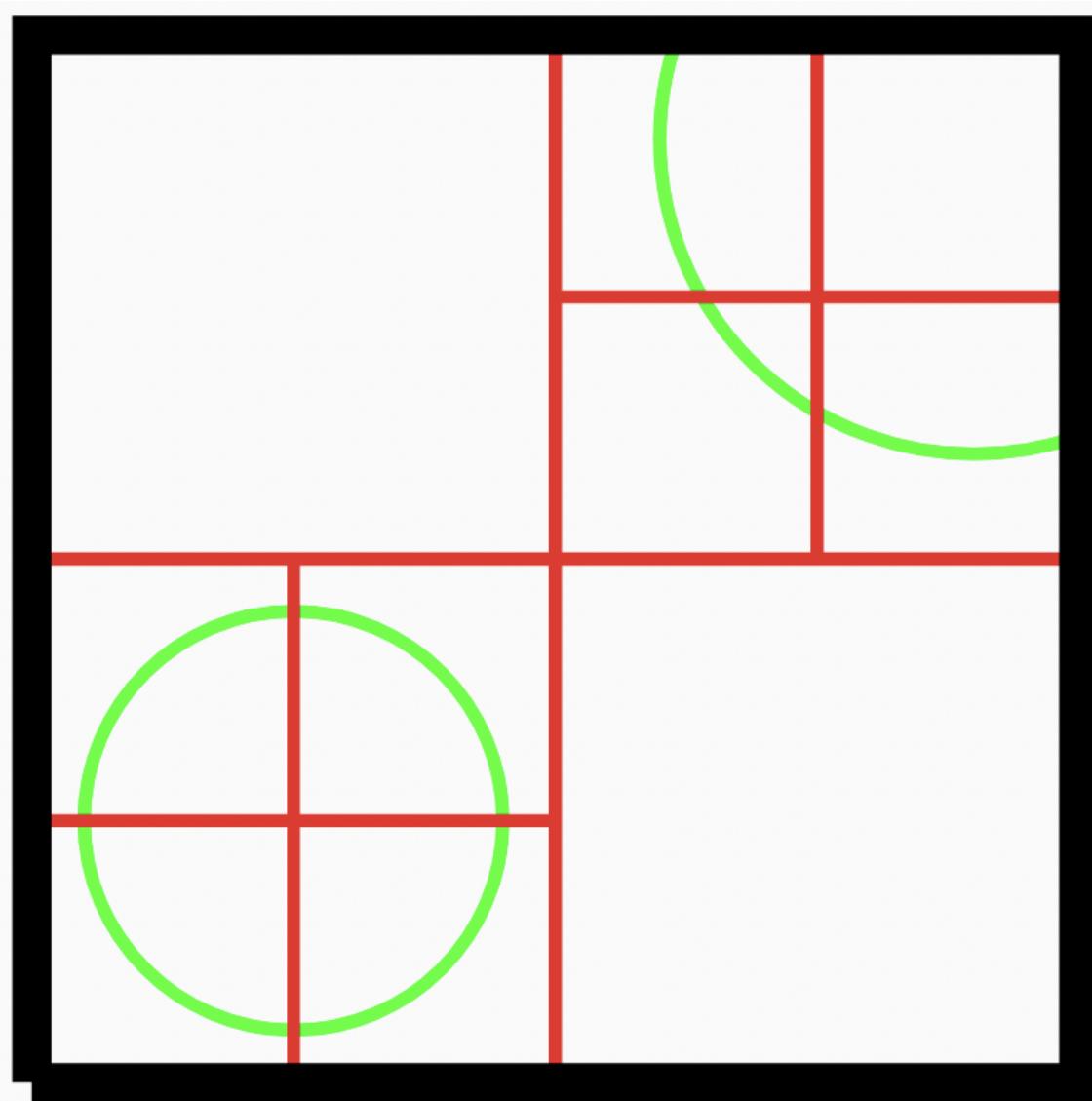
PV Algorithm in Action I

Subdivision STEP I



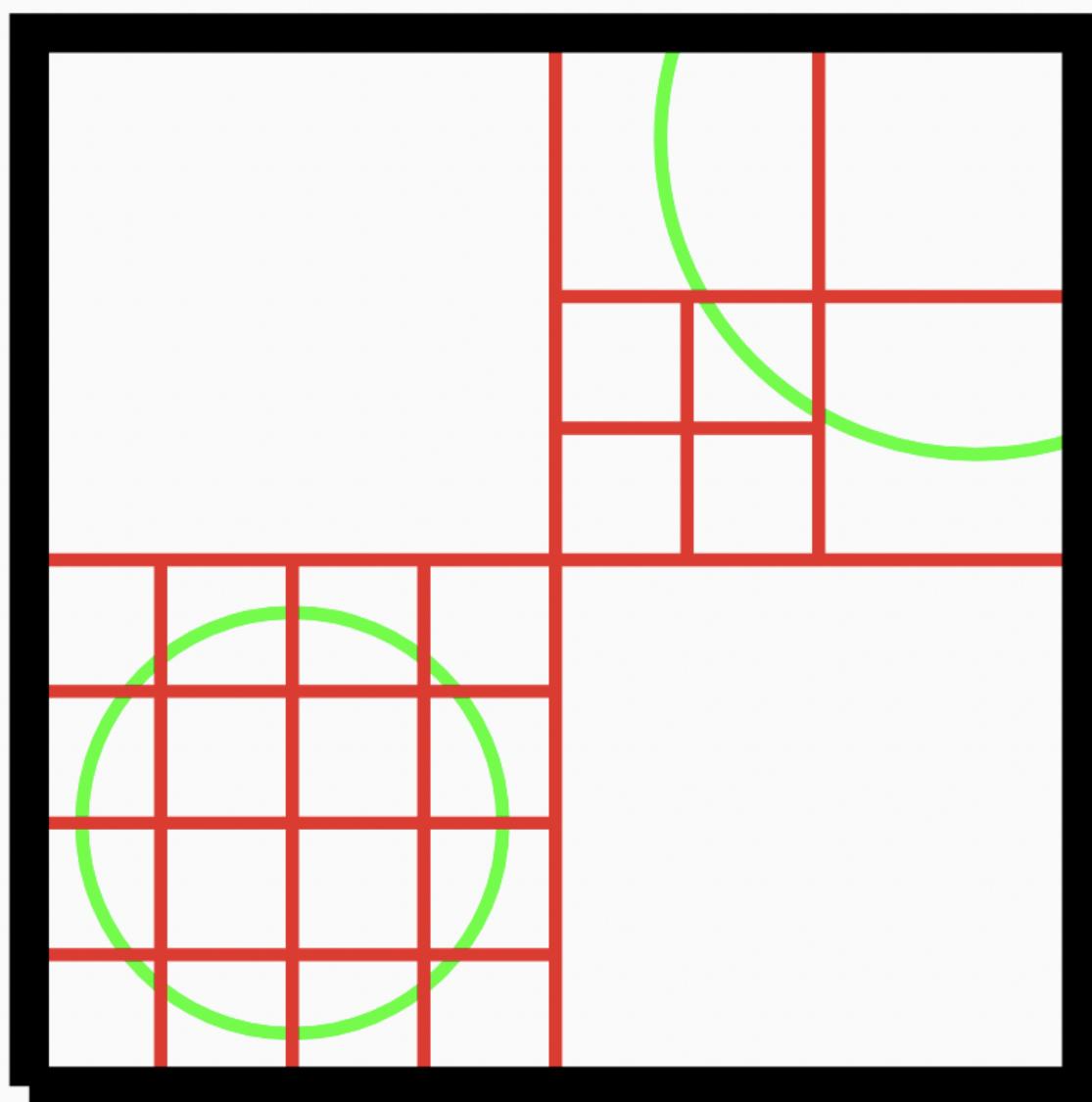
PV Algorithm in Action I

Subdivision STEP II



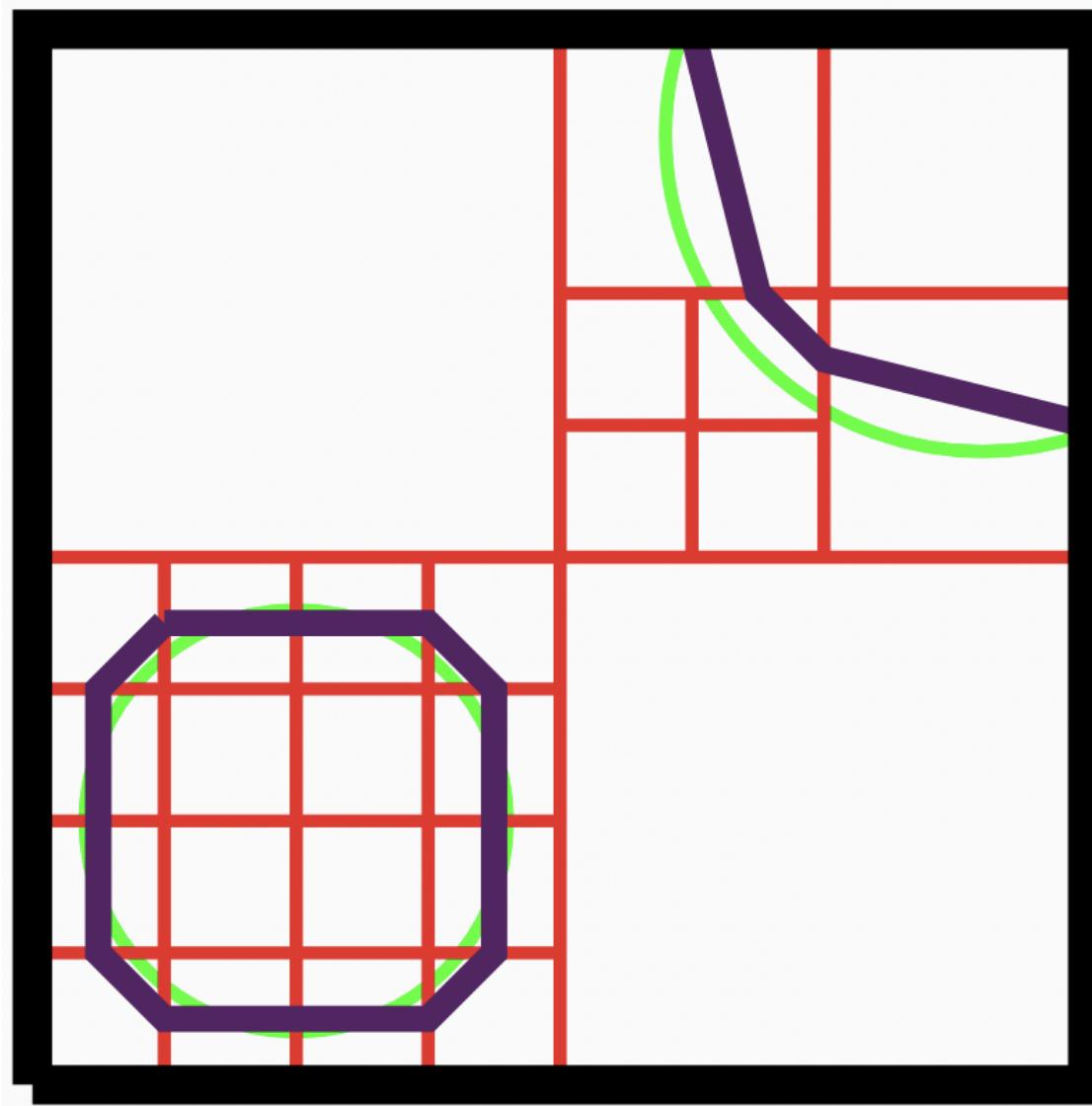
PV Algorithm in Action I

Subdivision STEP III

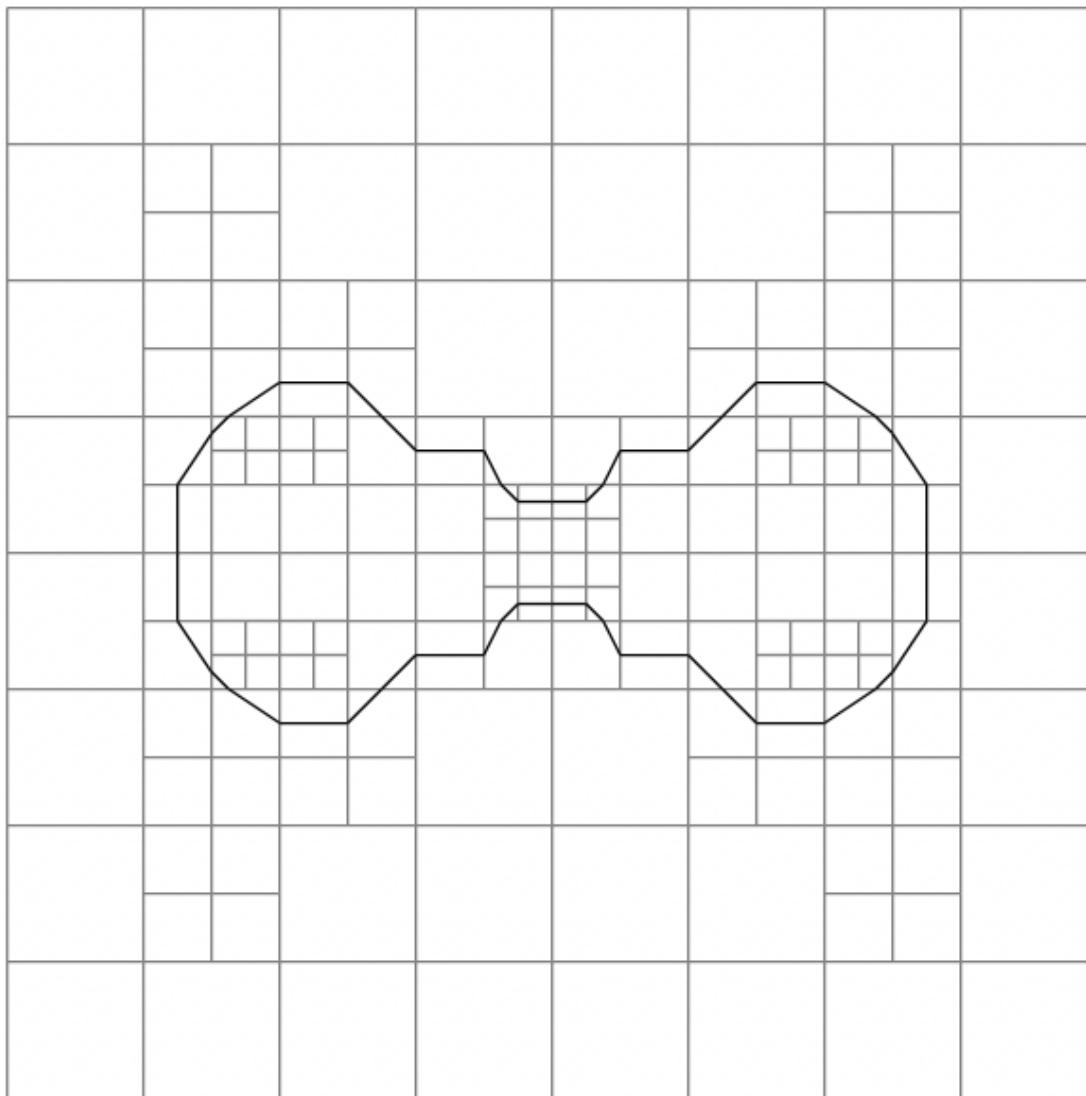


PV Algorithm in Action I

POSTPROCESSING STEP

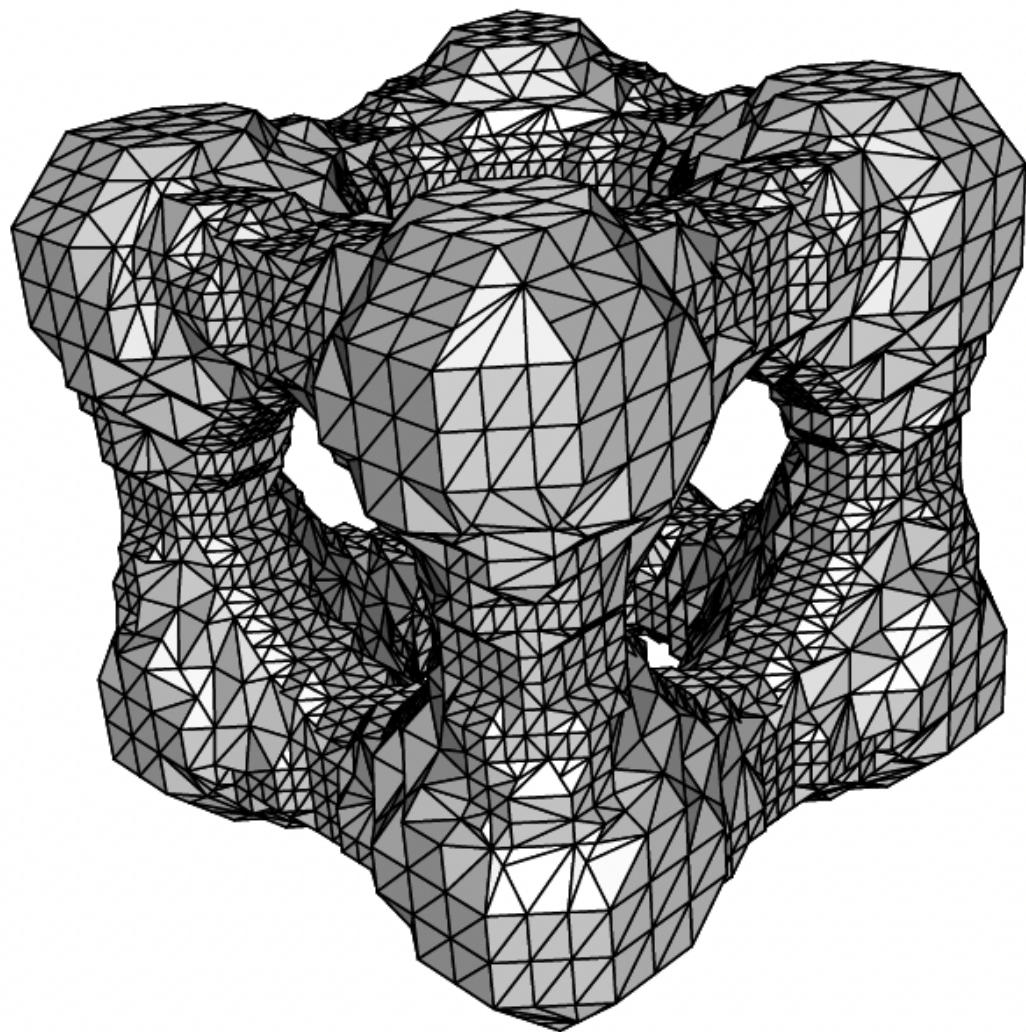


PV Algorithm in Action III



$$g = x^2(1-x)(1+x) - y^2 + 0.01$$

PV Algorithm in Action III



$$g = X^4 - 5X^2 + Y^4 - 5Y^2 + Z^4 - 5Z^2 + 10$$

PV Algorithm works in practice,

but worst-case bounds* were

too pessimistic!

* by Burr, Gao, Tsigaridas

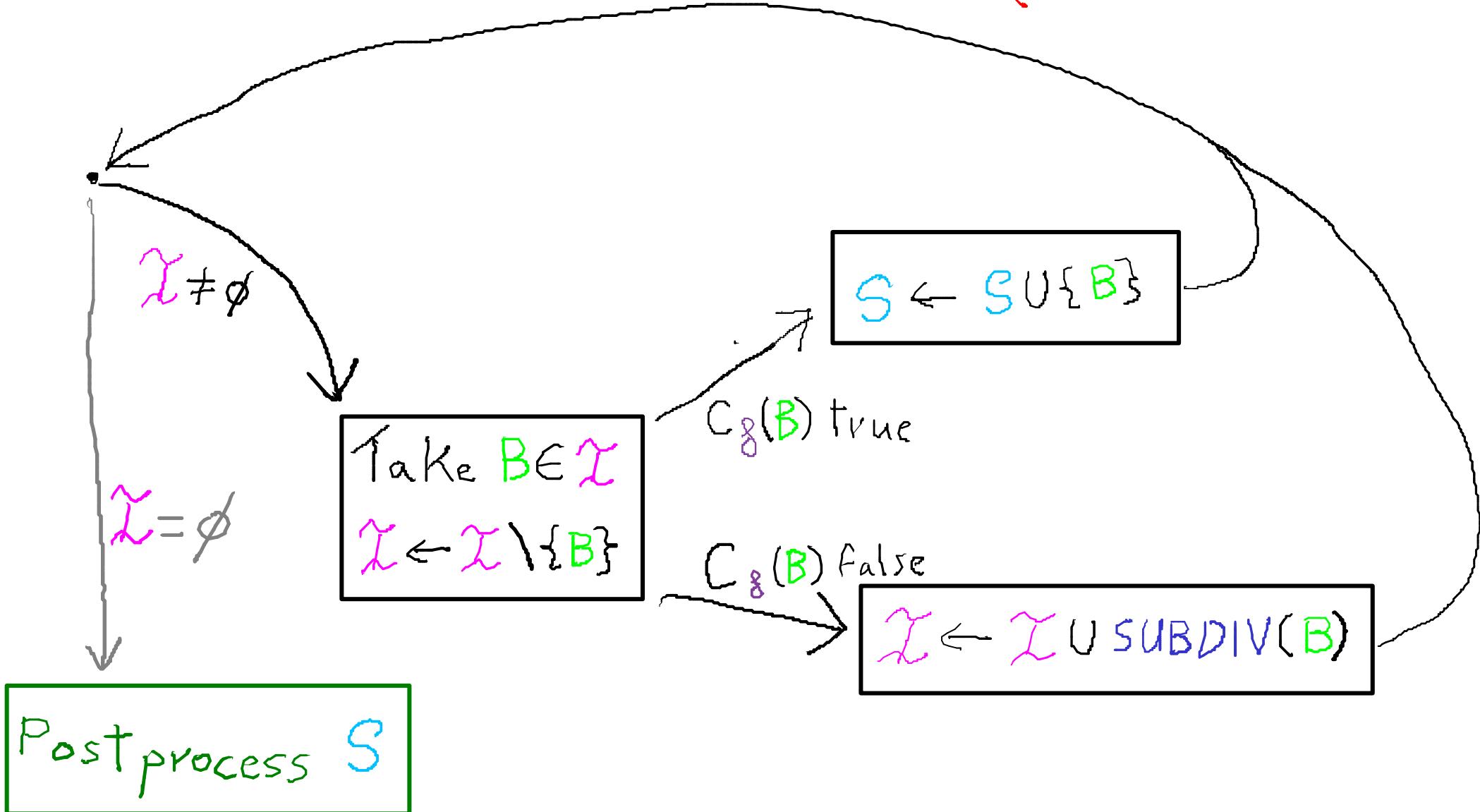
PLANTINGA-VEGTER CRITERION I

$$C_g(B) : \left\{ \begin{array}{l} 0 \notin g(B) \\ \text{OR} \\ \forall x, y \in B, \langle \nabla_x g, \nabla_y g \rangle \neq 0 \end{array} \right.$$

THM. If S is a subdivision of $[-a, a]^h$ s.t. for all $B \in S$, $C_g(B)$ holds. Then we can produce a PL approx of $\mathcal{Z}(g) \cap [-a, a]$ that is isotopically equiv.

PLANTINGA-VEGTER ALGORITHM

(Abstract level)



Q: How do we check $C_g(B)$?

INTERVAL APPROXIMATIONS I

DEF. An interval approximation of

$$F: \mathbb{R}^n \rightarrow \mathbb{R}^m$$

is a map

$$\square F: \square \mathbb{R}^n \rightarrow \square \mathbb{R}^m$$

where $\square \mathbb{R}^k := \{ B \subseteq \mathbb{R}^k \mid B \text{ is a box} \}$

such that For all $B \in \square \mathbb{R}^n$,

$$F(B) \subseteq \square F(B)$$

COMPLEXITY CONTEXT for PV

INPUT

$$g \in \mathbb{R}[x_1, \dots, x_n]$$

OUTPUT

PL-approximation of $\tilde{\chi}(g) \cap [-1, 1]$

COMPLEXITY PARAMETERS

d : degree of g

n : number variables

M : number of monomials in g

MEASURE OF COMPLEXITY

Size of subdivision

INTERVAL APPROXIMATIONS II

$$\|\underline{g}\|_1 := \sum_{\alpha} |\underline{g}_{\alpha}| \quad (\text{1-norm of } \underline{g})$$

$$\square \underline{g}(B) := g(m(B)) + d \|\underline{g}\|_1 \left[-\frac{w(B)}{2}, \frac{w(B)}{2} \right]$$

$$\square \|\nabla \underline{g}\|_1(B) := \nabla_{m(B)} \underline{g} + \sqrt{2^h d^2} \|\underline{g}\|_1 \left[-\frac{w(B)}{2}, \frac{w(B)}{2} \right]$$

THM. $\square \underline{g}$ is an interval approx of g

$\square \|\nabla \underline{g}\|_1$ is an interval approx of $\|\nabla g\|_1$

PLANTINGA-VEGTER CRITERION II

$$C_g^\square(B) : \left\{ \begin{array}{l} o \notin \square g(B) \\ \text{OR} \\ o \notin \square ||\nabla g||_1(B) \end{array} \right.$$

THM.

$$C_g^\square(B) \Rightarrow C_g(B)$$

CONDITION-BASED ESTIMATE I

Condition number

$$C(\varphi, x) := \frac{\|\varphi\|_1}{\max\{|\varphi(x)|, \|\nabla_x \varphi\|_1\}} \in [1, \infty]$$

Prop. $C(\varphi, x) = \infty \Leftrightarrow \varphi$ singular zero at x

Prop. If $x \in B$ and $C(\varphi, x) d \sqrt{2n} w(B) < 1$,

then $C_\varphi^B(B)$ holds

CONDITION-BASED ESTIMATE II

THM. The PV algorithm produces a subdivision with at most

$$2^{\frac{5}{2}n} n^{n/2} d^n \mathbb{E}_{x \in I^n} C(\delta, x)^n$$

boxes on δ

Proof relies on continuous amortization by Burr, Krahmer & Yap

PROBABILISTIC ESTIMATE I

UNIFORM CASE

$$f := \sum_{\alpha \in A} f_\alpha X^\alpha \quad \#A =: M$$

where f_α independent uniformly dist. on intervals of length 2

THM. On F , the PV algorithm produces a subdivision with

$$2^n 32^{n+1} d^{2n} M^{n+2}$$

boxes on average.

PROBABILISTIC ESTIMATE II

GAUSSIAN CASE

$$f := \sum_{\alpha \in A} f_\alpha X^\alpha \quad \#A =: M$$

where f_α independent Gaussians of variance 1

THM. On F , the PV algorithm produces a subdivision with

$$2^{n \frac{3}{2}} (10(n+1))^{n+1} d^{2n} M^{n+2}$$

boxes on average.

PROBABILISTIC ESTIMATE III

GENERAL CASE: ZINTZO POLYNOMIALS

$$f := \sum_{\alpha \in A} f_\alpha X^\alpha \quad \text{where } f_\alpha \text{ independent s.t.}$$

(Anti-concentration)

$$\forall t \in \mathbb{R}, \forall \varepsilon > 0, P(|f_\alpha - t| < \varepsilon) < p_\alpha \varepsilon$$

(Tail bounds)

$$\forall t \in \mathbb{R}_+, P(|f_\alpha| \geq t) \leq 2 \exp \left(- \left(\frac{t}{L_\alpha} \right)^p \right)$$

→ Allow us to use Geometric Functional Analysis

THE MODEL IS ROBUST

So

PV algorithm

is also efficient in theory!

THE FRAMEWORK IN ACTION II

the DESCARTES solver

Joint work with A.A. Ergür & E.Tsigaridas



Photo while working on this project

Real Root Isolation I: The Problem

INPUT:

$$f \in \mathbb{Z}[x]$$



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

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

We can also handle continuous inputs!

INPUT SIZE PARAMETERS:

d : degree of f

r : bit-size of coefficients of f

MEASURE OF RUN-TIME

Bit complexity

Real Root Isolation II:

The State of the Art

STURM SOLVER

$$\tilde{\mathcal{O}}_B(d^4 \gamma^2)$$

DESCARTES SOLVER

$$\tilde{\mathcal{O}}_B(d^4 \gamma^2)$$

ANewDsc

$$\tilde{\mathcal{O}}_B(d^3 + d^2 \gamma)$$

(Sagraloff & Mehlhorn; 2016)

PAN'S ALGORITHM

$$\tilde{\mathcal{O}}_B(d^2 \gamma)$$

(Pan; 2002)

Q: Can we beat the champion?

Real Root Isolation III:

What do we wish?

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

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

DESCARTES SOLVER I: Rule of Signs

$V(\gamma) := \# \text{ sign variations of } \gamma_0, \gamma_1, \dots$

THM (Descartes' rule of signs)

$$\#\mathcal{Z}(\gamma, \mathbb{R}) \leq V(\gamma)$$

Moreover,

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

COR

$$\#\mathcal{Z}(\gamma, (a, b)) \leq V(\gamma, (a, b)) := V\left((x+1)^d \cdot \gamma\left(\frac{bx+a}{x+1}\right)\right)$$

\uparrow
 $(0, \infty) \xrightarrow{\text{bijection}} (a, b)$



Portrait by Frans Hals
Source: Wikimedia Commons

DESCARTES SOLVER II:

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:

$$\# Z(g, D(m(J)), c_w(J)) \leq K \Rightarrow V(g, J) \leq K$$

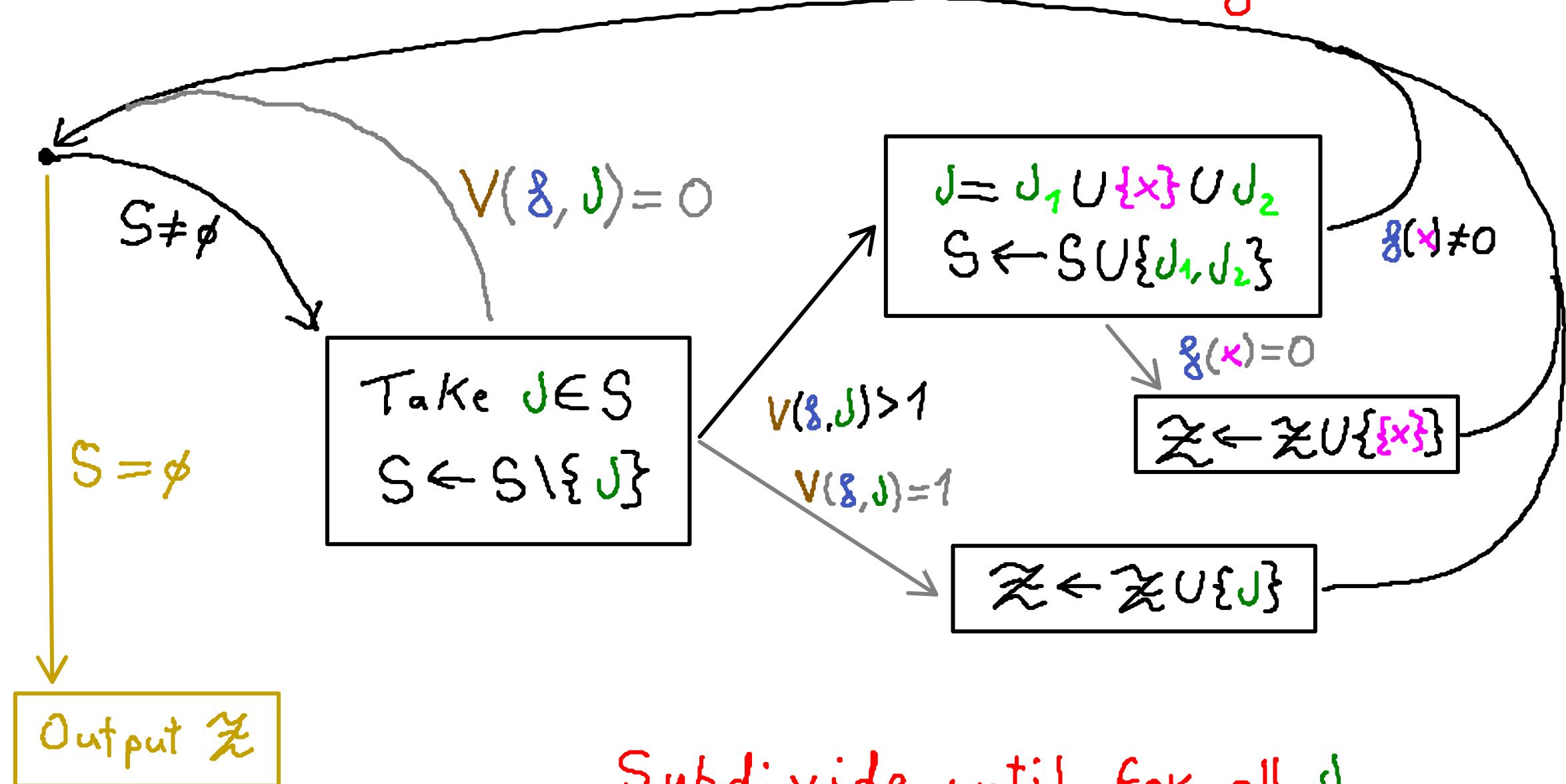
Obreshkoff's Thm: DESCARTES sees the complex roots around!

- 4) Subadditivity:

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

DESCARTES SOLVER III:

The Algorithm

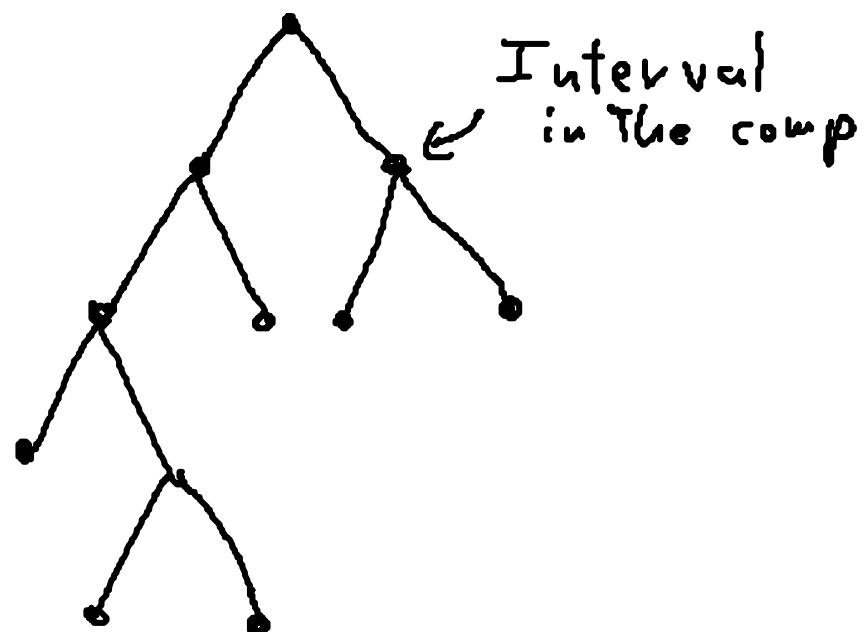


Subdivide until for all J ,
 $V(g, J) \leq 1$!

DESCARTES SOLVER N:

Descartes' tree

$$\gamma(g, I)$$



$$\text{size of } \gamma(g, I)$$



$$\text{run-time of } \text{DESCARTES}(g, I)$$

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

Real Root Isolation IV:

Are we being pessimistic?

DESCARTES SOLVER

seems to behave faster in practice!

↓ Can we explain this?

Real Root Isolation V:

Beyond pessimism

Yes, bounding

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

What's a 'good' random model for 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. $f_k \sim \mathcal{U}([-2^\gamma, 2^\gamma] \cap \mathbb{Z})$ independent

SIMPLE MAIN THM

$$\mathbb{E} \text{cost}(\text{DESCARTES}, f) = \tilde{\mathcal{O}}_B(d^2 + d\gamma)$$

On average, DESCARTES is almost optimal!

Beyond pessimism II:

Random Bit Polynomials

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

bit-size of F :

$$\gamma(F) := \min\{\gamma \mid \forall k, P(|F_k| \leq 2^\gamma) = 1\}$$

weight of F :

$$w(F) := \max \left\{ P(F_k = c) \mid c \in \mathbb{R}, k \in \{0, 1, \downarrow d-1, d\} \right\}$$

uniformity of F :

$$u(F) := \ln(w(F)(1 + 2^{\gamma(F)+1}))$$

Beyond pessimism III:

MAIN THEOREM

MAIN THM

$$\mathbb{E} \text{cost}(\text{DESCARTES}, F) = \tilde{O}_B(d^2 + d\gamma)(1 + u(F))^4$$

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

Claim: For many cases, $u(F) = O(1)$

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

On average, DESCARTES is almost optimal!

Beyond pessimism IV:

Examples of Random Bit Polynomials I

- Support control $\{0, 1, d-1, d\} \subseteq A$

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

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

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

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

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

Beyond pessimism V:

Examples of Random Bit Polynomials II

- Exact bitsize

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

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

+ their combinations

Our random model is flexible!

Beyond pessimism VI:

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
 $\sigma \in \mathbb{Z} \setminus \{0\}$ of entries of size γ

Then:

$$F_\sigma = g + \sigma f$$
 random bit polynomial
 $\& u(F_\sigma) \leq 1 + u(F) + \max\{\gamma - \gamma(F), \gamma(\sigma)\}$

The Ingredients of the Analysis I:

Condition Numbers

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

$C(g) = \infty \iff g$ 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

Upper bounds for

complex roots of f around $[-1, 1]$

→ Upper bounds for width of DESCARTES' tree

We only care
about nearby roots!
↙

Complex analysis!

Titchmarsh thm

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. $y_i \sim \mathcal{U}(-\frac{1}{2}, \frac{1}{2})$ i.i.d.

Then: $x+y$ continuous random var.

We can use our old cont. toolbox!

⚠ I am omitting a lot of technical details.

SUMMING UP:

DESCARTES

is almost optimal on average!

Eskerrik Asko

ZURE ARRETAGATIK!

I.e. Thank You For your attention!