The $\mathcal{G}$rid Method in Numerical $\mathbb{R}$eal Algebraic Geometry

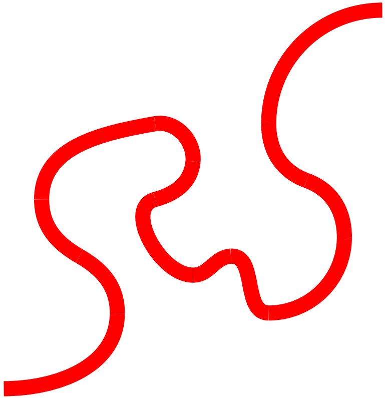
A short presentation at the XXI Santaló School of Mathematics

Josué Tonelli-Cueto
Inria Paris & IMJ-PRG
Website: https://tonellicueto.xyz
Blog: https://illsaidthoughts.xyz/

11/Jul/2022

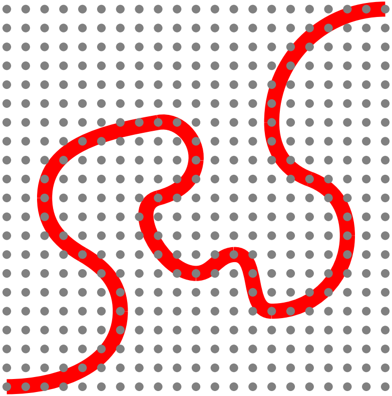
What is the $\mathcal{G}$rid Method?

What is the $\mathcal{G}$rid Method? I



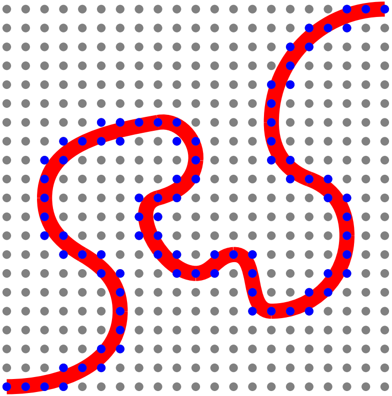
We want to capture a geometric object

What is the $\mathcal{G}$rid Method? I



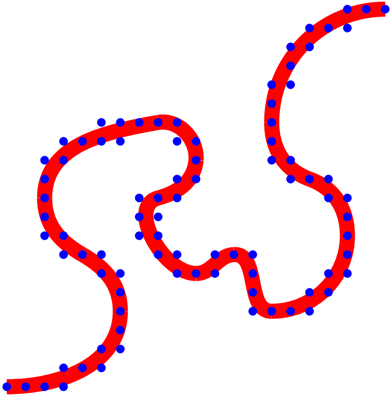
We cover the ambient space by a grid

What is the $\mathcal{G}$rid Method? I



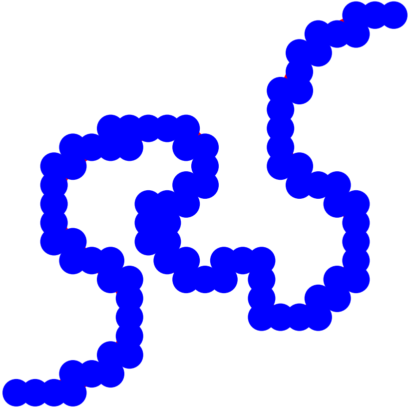
We select the nearby point to approximate the object

What is the $\mathcal{G}$rid Method? I



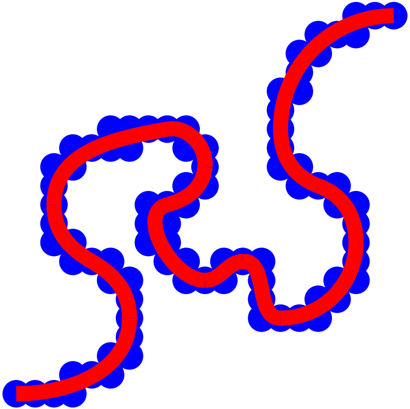
We select the nearby point to approximate the object
...and forget the rest!

What is the $\mathcal{G}$rid Method? I



We postprocess the selection

What is the $\mathcal{G}$rid Method? I



We postprocess the selection
to capture what we want about the geometric object of interes

What is the $\mathcal{G}$rid Method? II

1. Cover with a grid the ambient space where the desired object lies
   - Which properties should the covering grid have?
   - How can we generate such a covering grid efficiently?
2. Select the points that approximate the desired object
   - What means to 'approximate' the desired object?
   - How do we select the points so that we approximate the object?
   - How to refine the grid when selection fails? (Non-Adaptive vs. Adaptive)
3. Postprocess the points to obtain the information about the desired object
   - How can I combine the information coming from the selected points?
   - Can I do so in a fast way?

Numerical $\mathbb{R}$eal Algebraic Geometry (N$\mathbb{R}$AG)

N$\mathbb{R}$AG I: What is N$\mathbb{R}$AG?

What do we deal with?

Problems defined by real polynomials
considering errors in the given input

E.g. feasibility—is there a real zero?—, counting real zeros,
Betti numbers/homology groups of a semialgebraic set...

N$\mathbb{R}$AG II: The basics of condition numbers

Condition number:

- Measure of the sensitivity of the output to errors of the input for a specific problem
- Complexity of numerical algorithms depends on input size and condition number

  —same input-size doesn't mean similar run-times!

  —inputs with infinite condition number (*ill-posed*) cannot be handled numerically*
- Probabilistic analysis of the condition number for a random input

  gives probabilistic complexity

*If input is assumed to be real. If input is assume to be integer, then. . .

N$\mathbb{R}$AG III: Condition-based and prob. complexity

Worst-case complexity:

$$\max\{\mathrm{run\text{-}time}(\textsc{Algorithm}, i) \mid \mathrm{size}(i) \le s\}$$

Condition-based complexity:

$$\max\{\mathrm{run\text{-}time}(\textsc{Algorithm}, i) \mid \mathrm{size}(i) \le s,\ \mathrm{cond}(i) \le c\}$$

—the condition number allows to explain better the behaviour of $\textsc{Algorithm}$ at input i

Probabilistic complexity:

$$\mathbb{E}\{\mathrm{run\text{-}time}(\textsc{Algorithm}, \mathfrak{i}) \mid \mathrm{size}(\mathfrak{i}) \le s\}$$

...or randomly perturbed arbitrary input (smoothed paradigm)

$$\max\left\{\mathbb{E}\left(\mathrm{run\text{-}time}(\textsc{Algorithm}, \tilde{\mathfrak{i}}) \mid \mathrm{dist}(\tilde{\mathfrak{i}}, i) \le \sigma\right) \mid \mathrm{size}(i) \le s\right\}$$

N$\mathbb{R}$AG IV: Local condition number

Let $f \in \mathcal{H}_{n,\boldsymbol{d}}[q] := \prod_{i=1}^{q} \mathbb{R}[X_0, \ldots, X_n]_{d_i}$ and $x \in \mathbb{S}^n$, the *local condition number* of f at x is

$$\kappa_W(f, x) := \frac{\|f\|_W}{\sqrt{\|f(x)\|_2^2 + \sigma_q(\Delta^{-1/2}\mathrm{D}_x f)^2}}$$

where

- $\| \ \|_W$ is the Weyl norm—the Weyl norm is not the only choice!—,
- σ_q the qth singular value,
- Δ the diagonal matrix with $d_1, \ldots, d_q$ in the diagonal,and
- $\mathrm{D}_x f$ the tangent map $\mathrm{T}_x\mathbb{S}^n \to \mathbb{R}^q$ of f at x.

Main observation: $\kappa_W(f, x) = \infty$ iff x is a singular zero of f

N$\mathbb{R}$AG V: Global condition number

Let $f \in \mathcal{H}_{n,\boldsymbol{d}}[q] := \prod_{i=1}^{q} \mathbb{R}[X_0, \ldots, X_n]_{d_i}$,

(G) The *global condition number* of f is

$$\kappa_W(f) := \max_{x \in \mathbb{S}^n} \kappa_W(f, x).$$

- Controls complexity of non-adaptive grid methods
- **Probabilistic properties**: Small with high probability, but infinite expectation

(A) The *global-average condition number* is

$$\kappa_W^{\mathrm{av}}(f) := \sqrt[n]{\mathbb{E}_{x \in \mathbb{S}^n} \kappa_W(f, x)^n}.$$

- Controls complexity of adaptive grid methods
- **Probabilistic properties**: Finite moments up to order strictly less than $n + 1$

A Brief History of the $\mathcal{G}$rid Method in N$\mathbb{R}$AG

A Brief History of the $\mathcal{G}$rid Method in N$\mathbb{R}$AG

- (Cucker & Smale, 1999) Feasibility of semialgebraic sets
  —no probabilistic analysis
- (Cucker, Krick, Malajovich & Wschebor; 2008, 2009 & 2011) Counting real zeros
  —with high probability under Gaussian assumptions, but no finite expectation
  —under robust assumptions by (Ergür, Paouris & Rojas; 2019 & 2021)
- (Cucker, Krick & Shub, 2018) Homology of zero sets
  —with high probability under under Gaussian assumptions, but no finite expectation
  —also under robust assumptions (T-C; unpublished)
- (Bürgisser, Cucker & Lairez; 2018) Homology of basic semialgebraic sets
  —with high probability, but no finite expectation
- (Bürgisser, Cucker & T-C; 2020 & 2022) Homology of general semialgebraic sets
  —with high probability, but no finite expectation
- (T-C; MEGA 2021 & to appear in arXiv in 2022) Counting & computing real zeros
  —adaptively with finite expectation under robust assumptions!!!

Challenges and Future of the $\mathcal{G}$rid Method in N$\mathbb{R}$AG

Challenges and Future of the $\mathcal{G}$rid Method in N$\mathbb{R}$AG

- Construction of nice grids efficiently
  —we want to construct fast grids with structure we can exploit to compute faster
  (this is why I am here!)
- Postprocessing the subselection of the grid
  —this is one of the bottlenecks for using adaptive grids for homology computation
- Can we make our grids depend on the input?
  —the grids we construct are not input-sensitive
- Exploit the polynomial nature of the input
  —until now we only use that polynomials are C^2-functions
  —recent progress on this! (keep attention to arXiv this year)

Eskerrik asko por su atención!