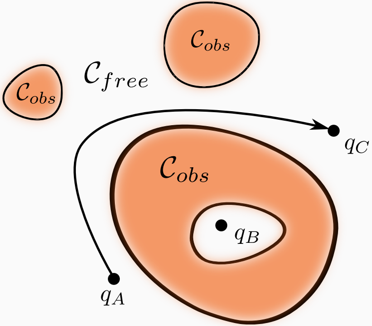
Connectivity in Real Algebraic Sets:
Algorithms & Applications

17$^{\text{th}}$ October 2024 CFHP Seminar



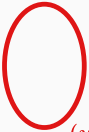
Rémi Prébet
Joint works with M. Safey El Din, É. Schost
Md N. Islam, A. Poteaux
D. Chablat, D. Salunkhe, P. Wenger

Slides:
rprebet.github.io/#talks
Interactive example:
rprebet.github.io/cristal.html

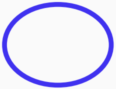
Sem-algebraic sets

Real solutions of **polynomial** systems of **equations** and **inequalities**

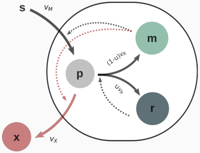
$$\begin{cases} 4y + x^3 - 4x^2 - 2x - 8 = 0 \\ -2 \leq x \leq 0 \end{cases}$$



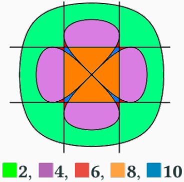
$$\frac{x^2}{4} + y^2 - 1 = 0$$



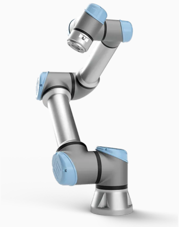
$$(x-1)^2 + \frac{(y-1)^2}{9} - 1 = 0$$



Biology



Physics

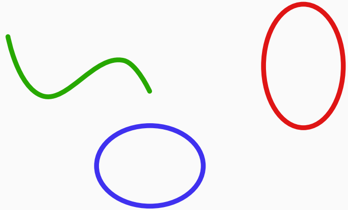


Robotics

Algorithms for polynomial systems with real variables

Sem-algebraic sets

Real solutions of **polynomial** systems of **equations** and **inequalities**



Fundamental algorithmic problems

Project : what is the set of possible values?

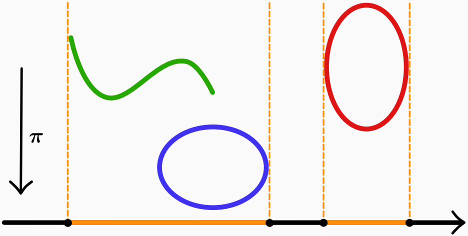
Sample : are there any solutions?

Connect : are two points connected?

Algorithms for polynomial systems with real variables

Sem-algebraic sets

Real solutions of **polynomial** systems of **equations** and **inequalities**

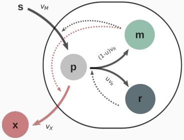


Fundamental algorithmic problems

Project : what is the set of possible values?

Sample : are there any solutions?

Connect : are two points connected?

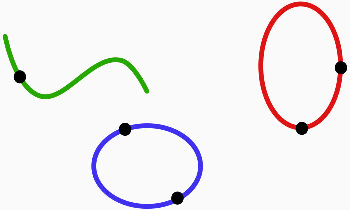


Biology

Algorithms for polynomial systems with real variables

Sem-algebraic sets

Real solutions of **polynomial** systems of **equations** and **inequalities**

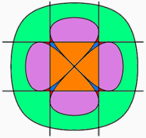


Fundamental algorithmic problems

Project : what is the set of possible values?

Sample : are there any solutions?

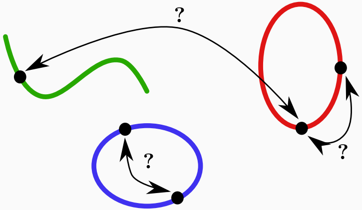
Connect : are two points connected?



Algorithms for polynomial systems with real variables

Sem-algebraic sets

Real solutions of **polynomial** systems of **equations** and **inequalities**



Fundamental algorithmic problems

Project : what is the set of possible values?

Sample : are there any solutions?

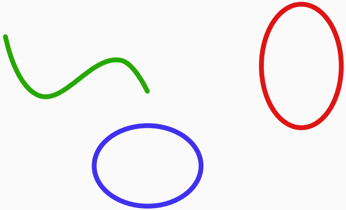
Connect : are two points connected?

Robotics

Algorithms for polynomial systems with real variables

Sem-algebraic sets

Real solutions of **polynomial** systems of **equations** and **inequalities**

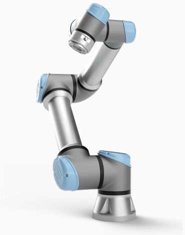


Fundamental algorithmic problems

Project : what is the set of possible values?

Sample : are there any solutions?

Connect : are two points connected?

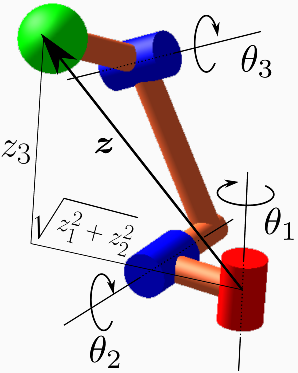


Robotics

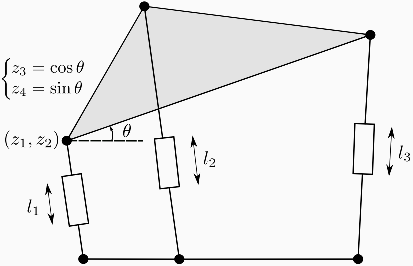
A quick look at robotics

Kinematic map of a robot

$$\mathcal{K}: \begin{array}{ccc} \mathbb{R}^d & \to & \mathbb{R}^d \\ (\boldsymbol{\ell}, \boldsymbol{\theta}) & \mapsto & \boldsymbol{z} = \big(z_1(\boldsymbol{\ell}, \boldsymbol{\theta}), \dots, z_d(\boldsymbol{\ell}, \boldsymbol{\theta})\big) \end{array}$$



An Orthogonal 3R Serial Robot



A 3-RPR Planar Parallel Robot

A quick look at robotics

Kinematic map of a robot

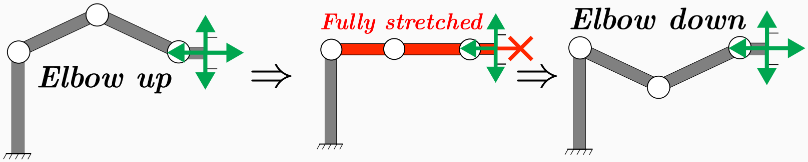
$$\begin{array}{cccc} \mathcal{K}: & \mathbb{R}^d & \to & \mathbb{R}^d \\ & (\boldsymbol{\ell}, \boldsymbol{\theta}) & \mapsto & \boldsymbol{z} = \Big(z_1(\boldsymbol{\ell}, \boldsymbol{\theta}), \ldots, z_d(\boldsymbol{\ell}, \boldsymbol{\theta})\Big) \end{array}$$

Not cuspidal: controlled posture

Kinematic map of a robot

$$\begin{array}{rccc} \mathcal{K}: & \mathbb{R}^d & \to & \mathbb{R}^d \\ & (\boldsymbol{\ell}, \boldsymbol{\theta}) & \mapsto & \boldsymbol{z} = \Big(z_1(\boldsymbol{\ell}, \boldsymbol{\theta}), \ldots, z_d(\boldsymbol{\ell}, \boldsymbol{\theta}) \Big) \end{array}$$

Not cuspidal: controlled posture



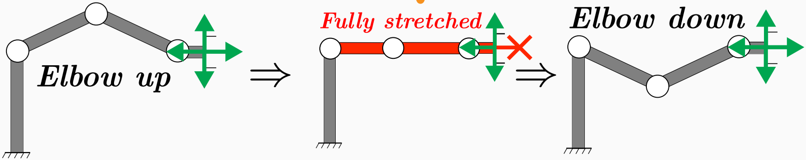
Kinematic map of a robot

$$\mathcal{K}\colon \quad \mathbb{R}^d \to \mathbb{R}^d, \qquad (\boldsymbol{\ell},\boldsymbol{\theta}) \mapsto \boldsymbol{z} = \Big(z_1(\boldsymbol{\ell},\boldsymbol{\theta}),\dots,z_d(\boldsymbol{\ell},\boldsymbol{\theta})\Big)$$

Singular posture

Configurations $(\boldsymbol{\ell},\boldsymbol{\theta})$ s.t. $\mathrm{Jac}_{\boldsymbol{\ell},\boldsymbol{\theta}}(\mathcal{K})$ is rank deficient

Not cuspidal: controlled posture



A quick look at robotics

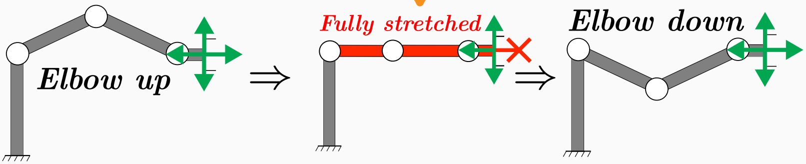
Kinematic map of a robot

$$\begin{array}{rccc} \mathcal{K}\colon & \mathbb{R}^d & \to & \mathbb{R}^d \\ & (\boldsymbol{\ell},\boldsymbol{\theta}) & \mapsto & \boldsymbol{z} = \big(z_1(\boldsymbol{\ell},\boldsymbol{\theta}),\ldots,z_d(\boldsymbol{\ell},\boldsymbol{\theta})\big) \end{array}$$

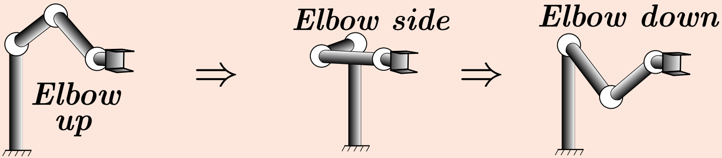
Singular posture

Configurations $(\boldsymbol{\ell},\boldsymbol{\theta})$ s.t. $\mathrm{Jac}_{\boldsymbol{\ell},\boldsymbol{\theta}}(\mathcal{K})$ is rank deficient

Not cuspidal: controlled posture



Cuspidal robot: uncontrolled posture



Cuspidal robot

Theorem [Borrel & Liégeois, 1986]

A robot **cannot** move between two associated postures, without passing by a singular posture

Cuspidal robot

~~Theorem~~ [Borrel & Liégois, 1986]

A robot **cannot** move between two associated postures, without passing by a singular posture

[**Wenger, 1992**] $\rightarrow$ **WRONG !**

~~Theorem~~ [Borrel & Liégois, 1986]

A robot **cannot** move between two associated postures, without passing by a singular posture

[Wenger, 1992] $\rightarrow$ **WRONG !**

Cuspidal robot [Wenger, 1992]

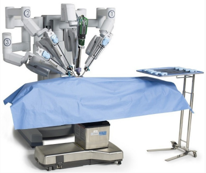
Cuspidal robots **can** move between two associated postures, without passing by a singular posture

Motivation

Cuspidal robots can **induce problem** for task planning

Open problem

Cuspidality decision for a general robot



Cuspidal robot

~~Theorem~~ [Borrel & Liégois, 1986]

A robot **cannot** move between two associated postures, without passing by a singular posture

[Wenger, 1992] $\rightarrow$ **WRONG !**

Cuspidal robot [Wenger, 1992]

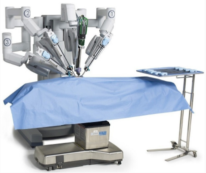
Cuspidal robots **can** move between two associated postures, without passing by a singular posture

Motivation

Cuspidal robots can **induce problem** for task planning

Open problem

Cuspidality decision for a general robot



Contribution NEW!

First general algorithm

Cuspidal robot

~~Theorem~~ [Borrel & Liégois, 1986]

A robot **cannot** move between two associated postures, without passing by a singular posture

[Wenger, 1992] $\rightarrow$ **WRONG !**

Cuspidal robot [Wenger, 1992]

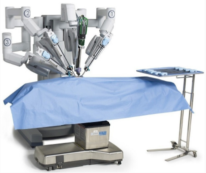
Cuspidal robots **can** move between two associated postures, without passing by a singular posture

Motivation

Cuspidal robots can **induce problem** for task planning

Open problem

Cuspidality decision for a general robot



Contribution NEW!

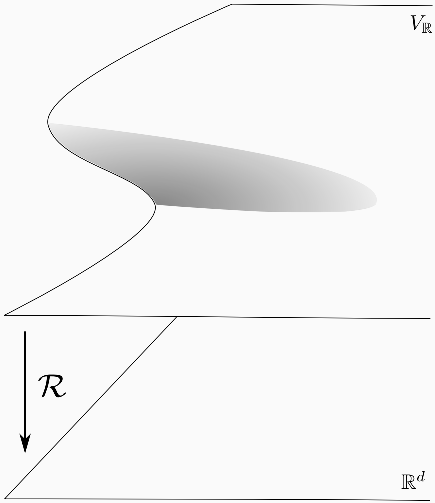
First general algorithm with singly exponential complexity

The algebraic cuspidality problem

Data

Data: $\boldsymbol{f} = (f_1, \ldots, f_s)$ and $\mathcal{R} = (r_1, \ldots, r_d)$ polynomials in $\mathbb{R}[x_1, \ldots, x_n]$

Assumptions: $V = \boldsymbol{V}(\boldsymbol{f})$ is d-equidimensional and $V_{\mathbb{R}} = V \cap \mathbb{R}^n \subsetneq \mathrm{sing}(V)$

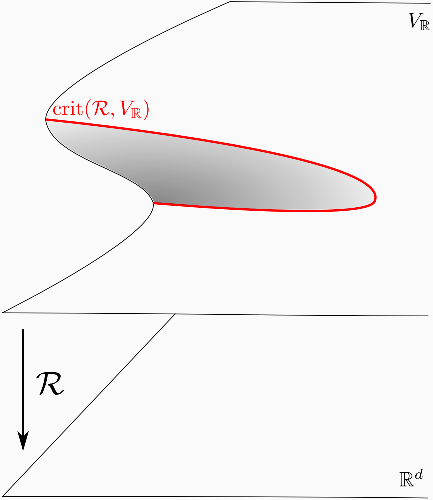


The algebraic cuspidality problem

Data

Data: $\boldsymbol{f} = (f_1, \ldots, f_s)$ and $\mathcal{R} = (r_1, \ldots, r_d)$ polynomials in $\mathbb{R}[x_1, \ldots, x_n]$

Assumptions: $V = \boldsymbol{V}(\boldsymbol{f})$ is d-equidimensional and $V_{\mathbb{R}} = V \cap \mathbb{R}^n \subsetneq \operatorname{sing}(V)$



The algebraic cuspidality problem

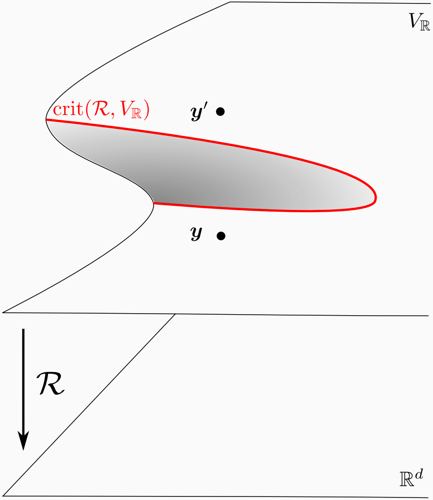
Data

Data: $\boldsymbol{f} = (f_1, \ldots, f_s)$ and $\mathcal{R} = (r_1, \ldots, r_d)$ polynomials in $\mathbb{R}[x_1, \ldots, x_n]$

Assumptions: $V = \boldsymbol{V}(\boldsymbol{f})$ is d-equidimensional and $V_{\mathbb{R}} = V \cap \mathbb{R}^n \subsetneq \operatorname{sing}(V)$

Algebraic cuspidality problem

The restriction of $\mathcal{R}$ to $V_{\mathbb{R}}$ is cuspidal if there is $\boldsymbol{y} \neq \boldsymbol{y}' \in V_{\mathbb{R}}$ such that



Data

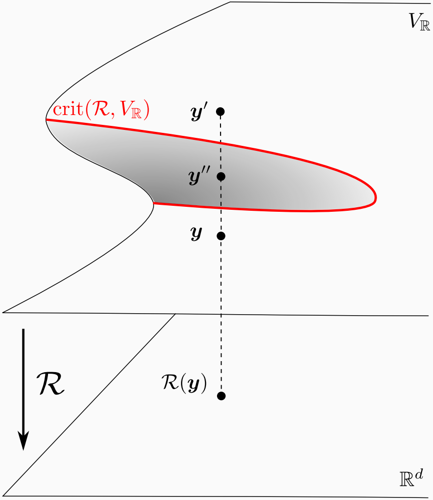
Data: $\boldsymbol{f} = (f_1, \ldots, f_s)$ and $\mathcal{R} = (r_1, \ldots, r_d)$ polynomials in $\mathbb{R}[x_1, \ldots, x_n]$

Assumptions: $V = \boldsymbol{V}(\boldsymbol{f})$ is d-equidimensional and $V_{\mathbb{R}} = V \cap \mathbb{R}^n \subsetneq \mathrm{sing}(V)$

Algebraic cuspidality problem

The restriction of $\mathcal{R}$ to $V_{\mathbb{R}}$ is cuspidal if there is $\boldsymbol{y} \neq \boldsymbol{y}' \in V_{\mathbb{R}}$ such that

1. $\mathcal{R}(\boldsymbol{y}) = \mathcal{R}(\boldsymbol{y}')$



Data

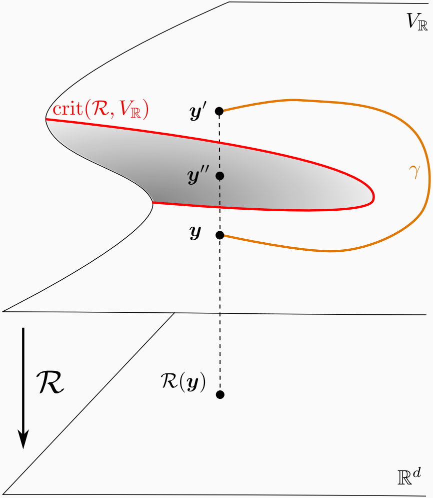
Data: $\boldsymbol{f} = (f_1, \dots, f_s)$ and $\mathcal{R} = (r_1, \dots, r_d)$ polynomials in $\mathbb{R}[x_1, \dots, x_n]$

Assumptions: $V = \boldsymbol{V}(\boldsymbol{f})$ is d-equidimensional and $V_{\mathbb{R}} = V \cap \mathbb{R}^n \subsetneq \mathrm{sing}(V)$

Algebraic cuspidality problem

The restriction of $\mathcal{R}$ to $V_{\mathbb{R}}$ is cuspidal if there is $\boldsymbol{y} \neq \boldsymbol{y}' \in V_{\mathbb{R}}$ such that

1. $\mathcal{R}(\boldsymbol{y}) = \mathcal{R}(\boldsymbol{y}')$
2. they are path-connected in $V_{\mathbb{R}} - \mathrm{crit}(\mathcal{R}, V)$



The algebraic cuspidality problem

Data

Data: $\boldsymbol{f} = (f_1, \ldots, f_s)$ and $\mathcal{R} = (r_1, \ldots, r_d)$ polynomials in $\mathbb{R}[x_1, \ldots, x_n]$

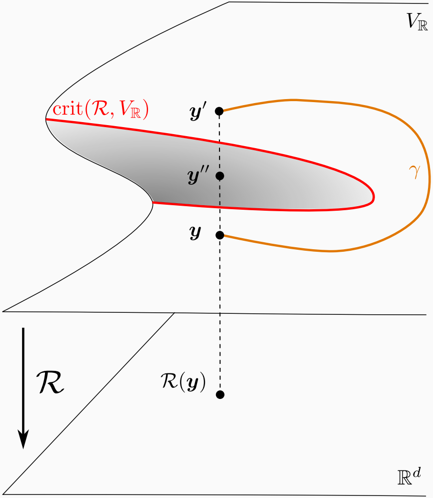
Assumptions: $V = \boldsymbol{V}(\boldsymbol{f})$ is d-equidimensional and $V_{\mathbb{R}} = V \cap \mathbb{R}^n \subsetneq \operatorname{sing}(V)$

Algebraic cuspidality problem

The restriction of $\mathcal{R}$ to $V_{\mathbb{R}}$ is cuspidal if there is $\boldsymbol{y} \neq \boldsymbol{y}' \in V_{\mathbb{R}}$ such that

1. $\mathcal{R}(\boldsymbol{y}) = \mathcal{R}(\boldsymbol{y}')$
2. they are path-connected in $V_{\mathbb{R}} - \operatorname{crit}(\mathcal{R}, V)$

$(\boldsymbol{y}, \boldsymbol{y}')$ is a cuspidal pair



The algebraic cuspidality problem

Data

Data: $\boldsymbol{f} = (f_1, \ldots, f_s)$ and $\mathcal{R} = (r_1, \ldots, r_d)$ polynomials in $\mathbb{R}[x_1, \ldots, x_n]$

Assumptions: $V = \boldsymbol{V}(\boldsymbol{f})$ is d-equidimensional and $V_{\mathbb{R}} = V \cap \mathbb{R}^n \subsetneq \operatorname{sing}(V)$

Algebraic cuspidality problem

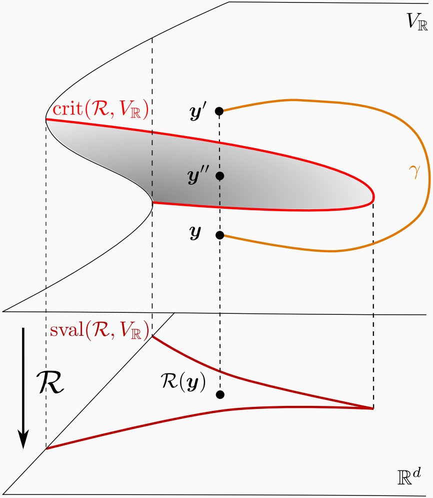
The restriction of $\mathcal{R}$ to $V_{\mathbb{R}}$ is cuspidal if there is $\boldsymbol{y} \neq \boldsymbol{y}' \in V_{\mathbb{R}}$ such that

1. $\mathcal{R}(\boldsymbol{y}) = \mathcal{R}(\boldsymbol{y}')$
2. they are path-connected in $V_{\mathbb{R}} - \operatorname{crit}(\mathcal{R}, V)$

$(\boldsymbol{y}, \boldsymbol{y}')$ is a <u>cuspidal pair</u>

Singular values of $\mathcal{R}$

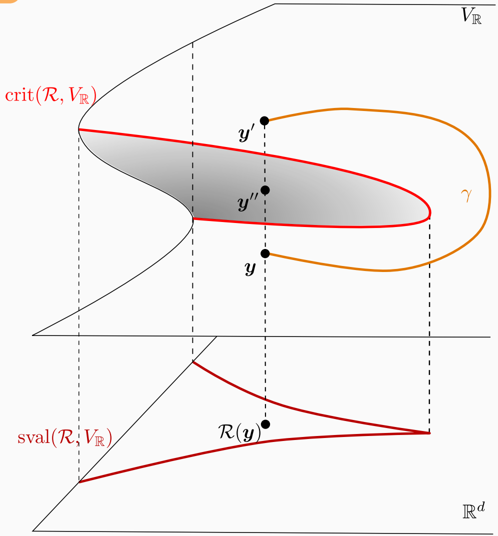
$\operatorname{sval}(\mathcal{R}, V) = \mathcal{R}\big(\operatorname{crit}(\mathcal{R}, V)\big)$



The cuspidality algorithm

Thom's First Isotopy Lemma

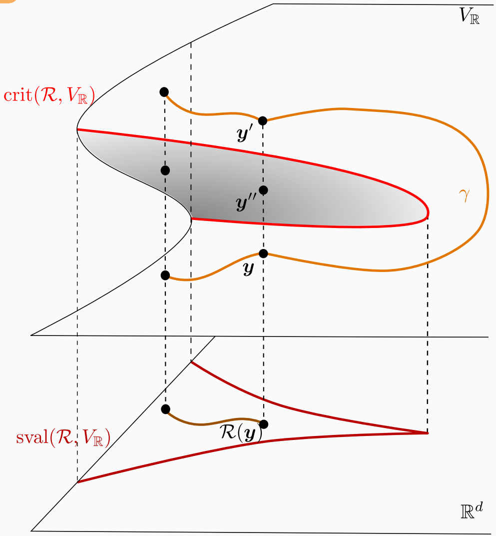
Fibers from the same connected component of $\mathbb{R}^d - \mathrm{sval}(\mathcal{R}, V)$ have the **same type**



The cuspidality algorithm

Thom's First Isotopy Lemma

Fibers from the same connected component of $\mathbb{R}^d - \mathrm{sval}(\mathcal{R}, V)$ have the **same type**



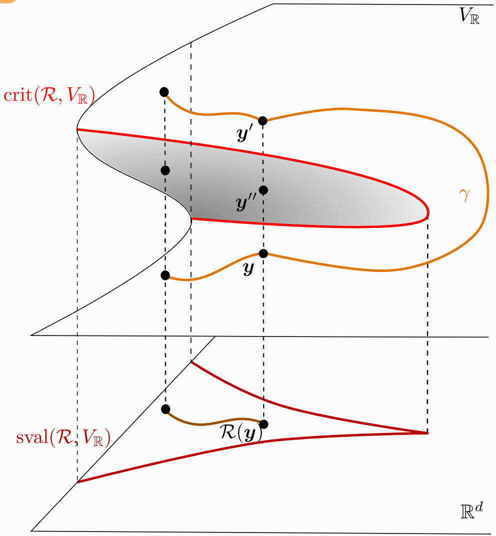
The cuspidality algorithm

Thom's First Isotopy Lemma

Fibers from the same connected component of $\mathbb{R}^d - \mathrm{sval}(\mathcal{R}, V)$ have the **same type**

$\Downarrow$

One fiber from each connected component of $\mathbb{R}^d - \mathrm{sval}(\mathcal{R}, V)$ is enough



The cuspidality algorithm

Thom's First Isotopy Lemma

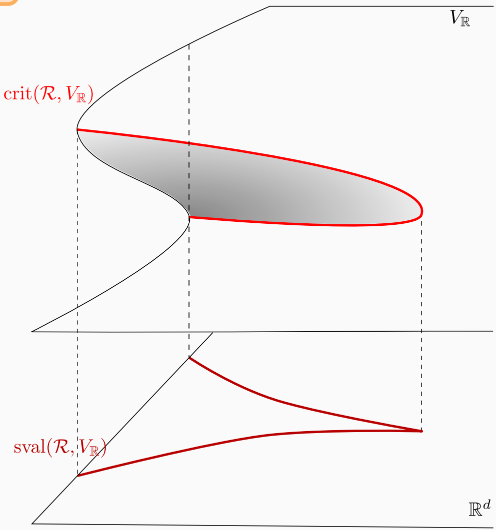
Fibers from the same connected component of $\mathbb{R}^d - \mathrm{sval}(\mathcal{R}, V)$ have the **same type**

$\Downarrow$

One fiber from each connected component of $\mathbb{R}^d - \mathrm{sval}(\mathcal{R}, V)$ is enough

Main steps

1. Compute polynomials defining $\mathrm{sval}(\mathcal{R}, V) = \mathcal{R}(\mathrm{crit}(\mathcal{R}, V))$



The cuspidality algorithm

Thom's First Isotopy Lemma

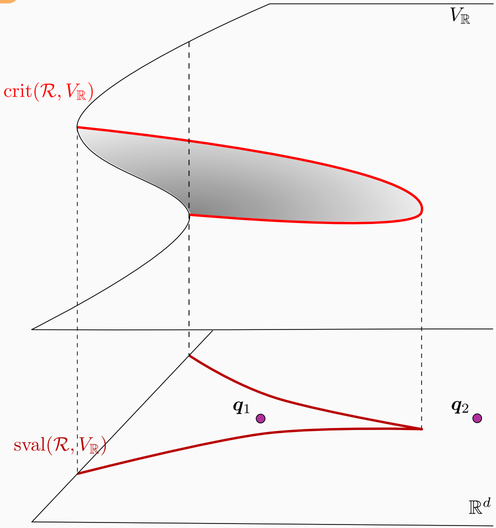
Fibers from the same connected component of $\mathbb{R}^d - \mathrm{sval}(\mathcal{R}, V)$ have the **same type**

$\Downarrow$

One fiber from each connected component of $\mathbb{R}^d - \mathrm{sval}(\mathcal{R}, V)$ is enough

Main steps

1. Compute polynomials defining $\mathrm{sval}(\mathcal{R}, V) = \mathcal{R}(\mathrm{crit}(\mathcal{R}, V))$
2. Compute a set $\mathcal{Q}$ of representatives in each connected component of $\mathbb{R}^d - \mathrm{sval}(\mathcal{R}, V)$



The cuspidality algorithm

Thom's First Isotopy Lemma

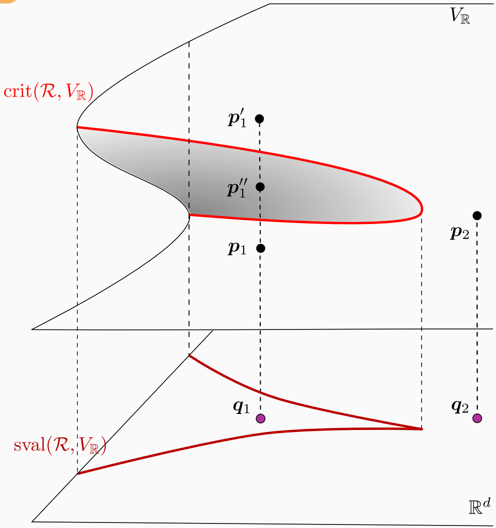
Fibers from the same connected component of $\mathbb{R}^d - \mathrm{sval}(\mathcal{R}, V)$ have the **same type**

$\Downarrow$

One fiber from each connected component of $\mathbb{R}^d - \mathrm{sval}(\mathcal{R}, V)$ is enough

Main steps

1. Compute polynomials defining $\mathrm{sval}(\mathcal{R}, V) = \mathcal{R}(\mathrm{crit}(\mathcal{R}, V))$
2. Compute a set $\mathcal{Q}$ of representatives in each connected component of $\mathbb{R}^d - \mathrm{sval}(\mathcal{R}, V)$
3. Compute their preimages $\mathcal{P} = V \cap \mathcal{R}^{-1}(\mathcal{Q})$



The cuspidality algorithm

Thom's First Isotopy Lemma

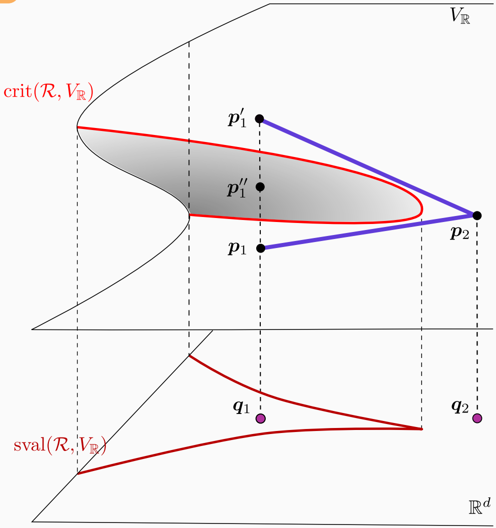
Fibers from the same connected component of $\mathbb{R}^d - \mathrm{sval}(\mathcal{R}, V)$ have the **same type**

$\Downarrow$

One fiber from each connected component of $\mathbb{R}^d - \mathrm{sval}(\mathcal{R}, V)$ is enough

Main steps

1. Compute polynomials defining $\mathrm{sval}(\mathcal{R}, V) = \mathcal{R}(\mathrm{crit}(\mathcal{R}, V))$
2. Compute a set $\mathcal{Q}$ of representatives in each connected component of $\mathbb{R}^d - \mathrm{sval}(\mathcal{R}, V)$
3. Compute their preimages $\mathcal{P} = V \cap \mathcal{R}^{-1}(\mathcal{Q})$
4. Search for cuspidal pairs in $\mathcal{P}$ by connecting points in the same connected component of $V_{\mathbb{R}} - \mathrm{crit}(\mathcal{R}, V)$



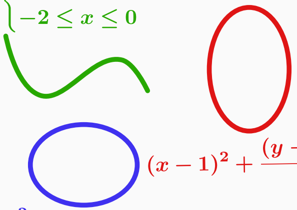
Complexities of semi-algebraic set algorithms

Semi-algebraic set S

Defined by s polynomials (equations+inequalities) with n variables of deg $\leq D$

General algorithm [Collins; '75]

Complexity: $(sD)^{2^{O(n)}}$



Fundamental algorithmic problems

Project S on t coordinates $\rightsquigarrow (sD)^{O(nt)}$ [Renegar, '92] **Optimal !**

Sample points of S $\rightsquigarrow (sD)^{O(n)}$ [Basu-Pollack-Roy, '98] **Optimal !**

Connect two points in S $\rightsquigarrow (sD)^{O(n^2)}$ [Canny, '88] **Not optimal !**

Quantitative bounds on algebraic sets

Real algebraic sets

$$V_{\mathbb{R}} = \{f_1 = \cdots f_p = 0\} \subset \mathbb{R}^n$$
where
$$(f_1, \ldots, f_p) \subset \mathbb{R}[x_1, \ldots, x_n]$$

$\iff$

Real trace of algebraic sets

$$V_{\mathbb{R}} = V \cap \mathbb{R}^n$$
where
$$V = \{f_1 = \cdots f_p = 0\} \subset \mathbb{C}^n$$

Quantitative bounds on algebraic sets

Real algebraic sets

$$V_{\mathbb{R}} = \{f_1 = \cdots f_p = 0\} \subset \mathbb{R}^n$$
where
$$(f_1, \ldots, f_p) \subset \mathbb{R}[x_1, \ldots, x_n]$$

$\Longleftrightarrow$

Real trace of algebraic sets

$$V_{\mathbb{R}} = V \cap \mathbb{R}^n$$
where
$$V = \{f_1 = \cdots f_p = 0\} \subset \mathbb{C}^n$$

$\mathcal{H}_1, \ldots, \mathcal{H}_n$ generic hyperplanes

Dimension

$\dim V$ = smallest d s.t. $\mathrm{card}\left(V \cap \mathcal{H}_1 \cap \ldots \cap \mathcal{H}_d\right) < +\infty$

Degree

$$\deg V = \mathrm{card}\left(V \cap \mathcal{H}_1 \cap \ldots \cap \mathcal{H}_d\right)$$

Quantitative bounds on algebraic sets

Real algebraic sets

$$V_{\mathbb{R}} = \{f_1 = \cdots f_p = 0\} \subset \mathbb{R}^n$$
where
$$(f_1, \dots, f_p) \subset \mathbb{R}[x_1, \dots, x_n]$$

$\Longleftrightarrow$

Real trace of algebraic sets

$$V_{\mathbb{R}} = V \cap \mathbb{R}^n$$
where
$$V = \{f_1 = \cdots f_p = 0\} \subset \mathbb{C}^n$$

$\mathcal{H}_1, \dots, \mathcal{H}_n$ generic hyperplanes

Dimension

$$\dim V = \text{smallest } d \quad \text{s.t.}$$
$$\text{card}\left(V \cap \mathcal{H}_1 \cap \ldots \cap \mathcal{H}_d\right) < +\infty$$

Degree

$$\deg V = \text{card}\left(V \cap \mathcal{H}_1 \cap \ldots \cap \mathcal{H}_d\right)$$

Quantitative bounds on algebraic sets

Real algebraic sets

$$V_{\mathbb{R}} = \{f_1 = \cdots f_p = 0\} \subset \mathbb{R}^n$$
where
$$(f_1, \ldots, f_p) \subset \mathbb{R}[x_1, \ldots, x_n]$$

$\iff$

Real trace of algebraic sets

$$V_{\mathbb{R}} = V \cap \mathbb{R}^n$$
where
$$V = \{f_1 = \cdots f_p = 0\} \subset \mathbb{C}^n$$

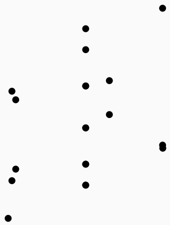
$\mathcal{H}_1, \ldots, \mathcal{H}_n$ generic hyperplanes

Dimension

$\dim V =$ smallest d s.t.
$$\operatorname{card}\left(V \cap \mathcal{H}_1 \cap \ldots \cap \mathcal{H}_d\right) < +\infty$$

Degree

$$\deg V = \operatorname{card}\left(V \cap \mathcal{H}_1 \cap \ldots \cap \mathcal{H}_d\right)$$



$$V = \{p_1, \ldots, p_{15}\}$$
$$\deg V = 15$$

Quantitative bounds on algebraic sets

Real algebraic sets

$$V_{\mathbb{R}} = \{f_1 = \cdots f_p = 0\} \subset \mathbb{R}^n$$
where
$$(f_1, \ldots, f_p) \subset \mathbb{R}[x_1, \ldots, x_n]$$

$\iff$

Real trace of algebraic sets

$$V_{\mathbb{R}} = V \cap \mathbb{R}^n$$
where
$$V = \{f_1 = \cdots f_p = 0\} \subset \mathbb{C}^n$$

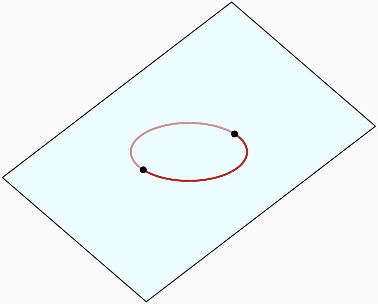
$\mathcal{H}_1, \ldots, \mathcal{H}_n$ generic hyperplanes

Dimension

$\dim V$ = smallest d s.t. $\operatorname{card}\left(V \cap \mathcal{H}_1 \cap \ldots \cap \mathcal{H}_d\right) < +\infty$

Degree

$$\deg V = \operatorname{card}\left(V \cap \mathcal{H}_1 \cap \ldots \cap \mathcal{H}_d\right)$$



$$\boldsymbol{V}\left(x^2 + y^2 - 1,\ z\right)$$
$$\Rightarrow \deg V = 2$$

Quantitative bounds on algebraic sets

Real algebraic sets

$$V_{\mathbb{R}} = \{f_1 = \cdots f_p = 0\} \subset \mathbb{R}^n$$
where
$$(f_1, \ldots, f_p) \subset \mathbb{R}[x_1, \ldots, x_n]$$

$\Longleftrightarrow$

Real trace of algebraic sets

$$V_{\mathbb{R}} = V \cap \mathbb{R}^n$$
where
$$V = \{f_1 = \cdots f_p = 0\} \subset \mathbb{C}^n$$

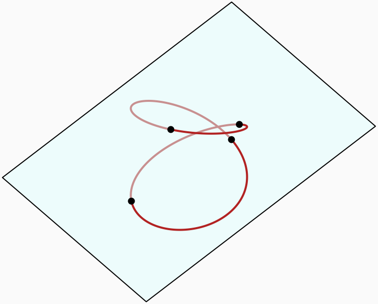
$\mathcal{H}_1, \ldots, \mathcal{H}_n$ generic hyperplanes

Dimension

$\dim V$ = smallest d s.t.
$$\text{card}\left(V \cap \mathcal{H}_1 \cap \ldots \cap \mathcal{H}_d\right) < +\infty$$

Degree

$$\deg V = \text{card}\left(V \cap \mathcal{H}_1 \cap \ldots \cap \mathcal{H}_d\right)$$



Bézout Bound

$$\deg V \leq \prod_{j=1}^{p} \deg f_j$$

$$\boldsymbol{V}\left(x^2 + y^2 - 1,\ 2z^2 - x - 1\right)$$
$$\Rightarrow \deg V = 4$$

Quantitative bounds on algebraic sets

Real algebraic sets

$$V_{\mathbb{R}} = \{f_1 = \cdots f_p = 0\} \subset \mathbb{R}^n$$
where
$$(f_1, \ldots, f_p) \subset \mathbb{R}[x_1, \ldots, x_n]$$

$\Longleftrightarrow$

Real trace of algebraic sets

$$V_{\mathbb{R}} = V \cap \mathbb{R}^n$$
where
$$V = \{f_1 = \cdots f_p = 0\} \subset \mathbb{C}^n$$

$\mathcal{H}_1, \ldots, \mathcal{H}_n$ generic hyperplanes

Dimension

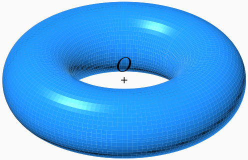
$$\dim V = \text{smallest } d \quad \text{s.t.}$$
$$\operatorname{card}\left(V \cap \mathcal{H}_1 \cap \ldots \cap \mathcal{H}_{\boldsymbol{d}}\right) < +\infty$$

Degree

$$\deg V = \operatorname{card}\left(V \cap \mathcal{H}_1 \cap \ldots \cap \mathcal{H}_{\boldsymbol{d}}\right)$$

Bézout Bound

$$\deg V \leq \prod_{j=1}^{p} \deg f_j$$



$$\boldsymbol{V}\left((x^2+y^2+z^2+\alpha)^2 - \beta(x^2+y^2)\right)$$

Quantitative bounds on algebraic sets

Real algebraic sets

$$V_{\mathbb{R}} = \{f_1 = \cdots f_p = 0\} \subset \mathbb{R}^n$$
where
$$(f_1, \ldots, f_p) \subset \mathbb{R}[x_1, \ldots, x_n]$$

$\Longleftrightarrow$

Real trace of algebraic sets

$$V_{\mathbb{R}} = V \cap \mathbb{R}^n$$
where
$$V = \{f_1 = \cdots f_p = 0\} \subset \mathbb{C}^n$$

$\mathcal{H}_1, \ldots, \mathcal{H}_n$ generic hyperplanes

Dimension

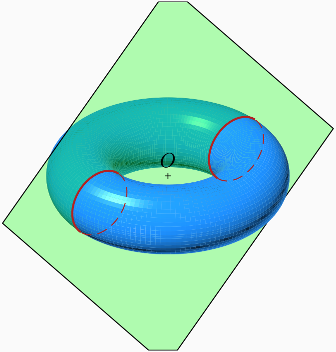
$\dim V$ = smallest d s.t.
$$\operatorname{card}\left(V \cap \mathcal{H}_1 \cap \ldots \cap \mathcal{H}_{d}\right) < +\infty$$

Degree

$$\deg V = \operatorname{card}\left(V \cap \mathcal{H}_1 \cap \ldots \cap \mathcal{H}_{d}\right)$$

Bézout Bound

$$\deg V \leq \prod_{j=1}^{p} \deg f_j$$



$$\boldsymbol{V}\left((x^2+y^2+z^2+\alpha)^2 - \beta(x^2+y^2)\right)$$

Quantitative bounds on algebraic sets

Real algebraic sets

$$V_{\mathbb{R}} = \{f_1 = \cdots f_p = 0\} \subset \mathbb{R}^n$$
where
$$(f_1, \ldots, f_p) \subset \mathbb{R}[x_1, \ldots, x_n]$$

$\iff$

Real trace of algebraic sets

$$V_{\mathbb{R}} = V \cap \mathbb{R}^n$$
where
$$V = \{f_1 = \cdots f_p = 0\} \subset \mathbb{C}^n$$

$\mathcal{H}_1, \ldots, \mathcal{H}_n$ generic hyperplanes

Dimension

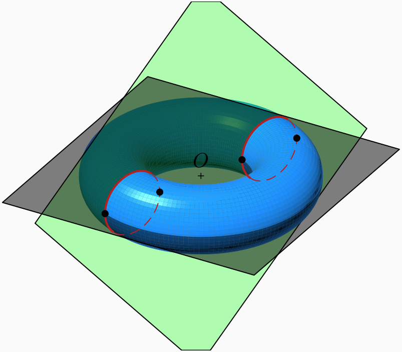
$\dim V$ = smallest d s.t.
$$\text{card}\left(V \cap \mathcal{H}_1 \cap \ldots \cap \mathcal{H}_d\right) < +\infty$$

Degree

$$\deg V = \text{card}\left(V \cap \mathcal{H}_1 \cap \ldots \cap \mathcal{H}_d\right)$$

Bézout Bound

$$\deg V \leq \prod_{j=1}^{p} \deg f_j$$



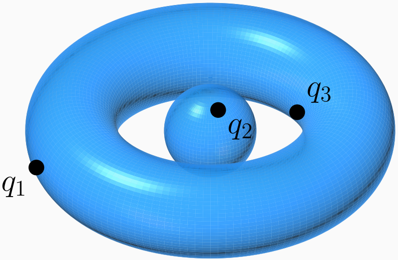
$$\boldsymbol{V}\left((x^2 + y^2 + z^2 + \alpha)^2 - \beta(x^2 + y^2)\right)$$
$$\Rightarrow \deg V = 4$$

Computing connectivity properties: Roadmaps

[Canny, 1988] Compute $\mathscr{R} \subset S$ one-dimensional, sharing its connectivity

Roadmap of $(S, \mathcal{P})$

A semi-algebraic curve $\mathscr{R} \subset S$, containing query points $(q_1, \ldots, q_N)$ s.t.
for all connected components C of S: $C \cap \mathscr{R}$ is non-empty and connected

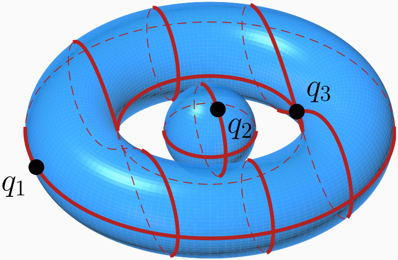


Computing connectivity properties: Roadmaps

[Canny, 1988] Compute $\mathscr{R} \subset S$ one-dimensional, sharing its connectivity

Roadmap of $(S, \mathcal{P})$

A semi-algebraic curve $\mathscr{R} \subset S$, containing query points $(q_1, \ldots, q_N)$ s.t.
for all connected components C of S: $C \cap \mathscr{R}$ is non-empty and connected



Computing connectivity properties: Roadmaps

💡[Canny, 1988] Compute $\mathscr{R} \subset S$ one-dimensional, sharing its connectivity

Roadmap of $(S, \mathcal{P})$

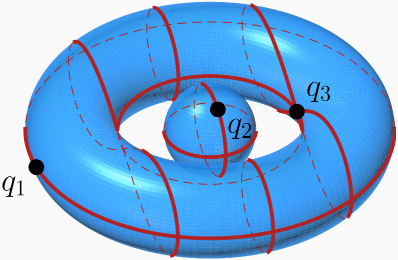
A semi-algebraic curve $\mathscr{R} \subset S$, containing query points $(q_1, \ldots, q_N)$ s.t. for all connected components C of S: $C \cap \mathscr{R}$ is non-empty and connected

Proposition

q_i and q_j are path-connected in $S \iff$ they are in $\mathscr{R}$

Problem reduction

Arbitrary dimension



Computing connectivity properties: Roadmaps

[Canny, 1988] Compute $\mathscr{R} \subset S$ one-dimensional, sharing its connectivity

Roadmap of $(S, \mathcal{P})$

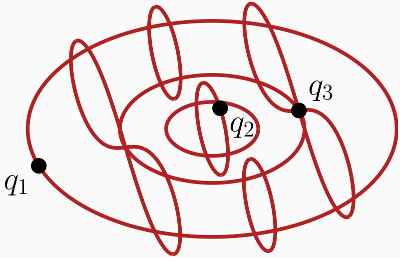
A semi-algebraic curve $\mathscr{R} \subset S$, containing query points $(q_1, \ldots, q_N)$ s.t. for all connected components C of S: $C \cap \mathscr{R}$ is non-empty and connected

Proposition

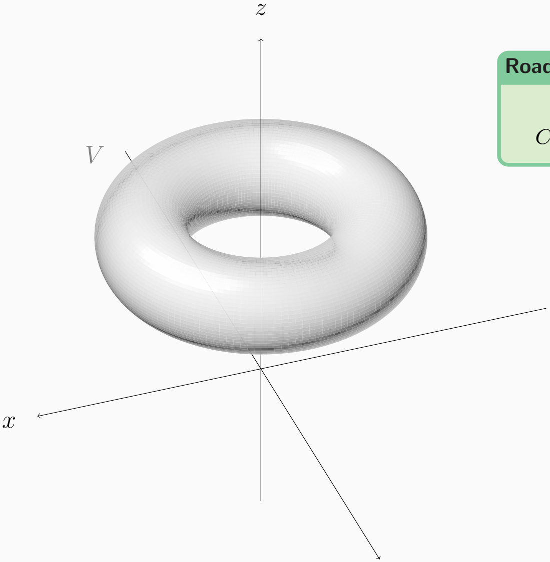
q_i and q_j are path-connected in $S \iff$ they are in $\mathscr{R}$

Problem reduction

Arbitrary dimension $\underset{\text{ROADMAP}}{\Longrightarrow}$ Dimension 1



Canny's strategy

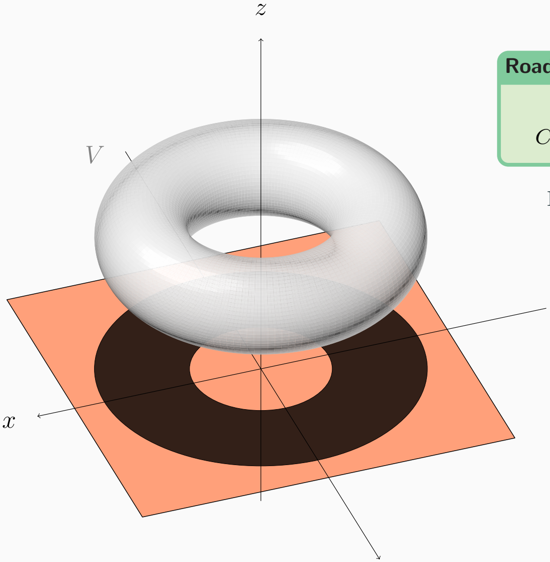


Roadmap property

$\forall\, C$ connected component,

$C \cap \mathscr{R}$ is non-empty and connected

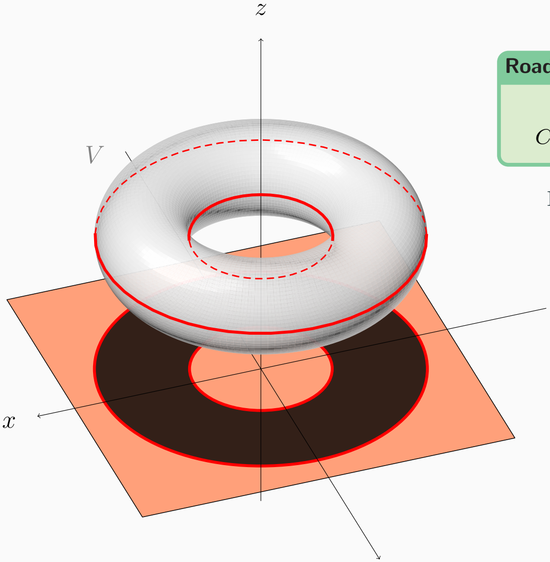
Canny's strategy



Roadmap property

$\forall\, C$ connected component,
$C \cap \mathscr{R}$ is non-empty and connected

Projection through:
$\pi_2 \colon (x_1, \ldots, x_n) \mapsto (x_1, x_2)$



Roadmap property

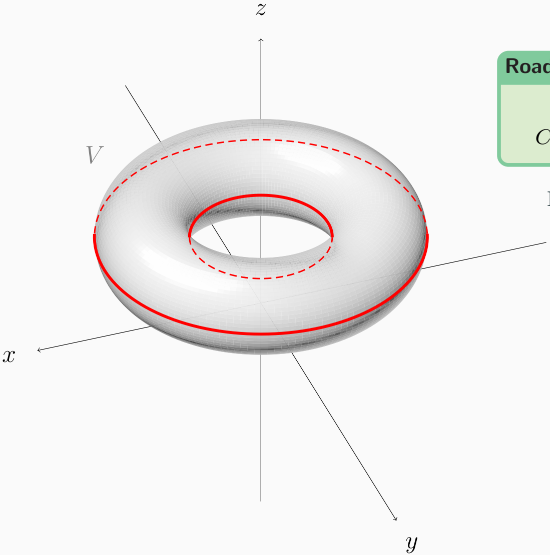
$\forall\, C$ connected component,
$C \cap \mathscr{R}$ is non-empty and connected

Projection through:
$\pi_2 : (x_1, \ldots, x_n) \mapsto (x_1, x_2)$

$W(\pi_2, V)$ critical locus of π_2.

Intersects all the
connected components of V

Canny's strategy



Roadmap property

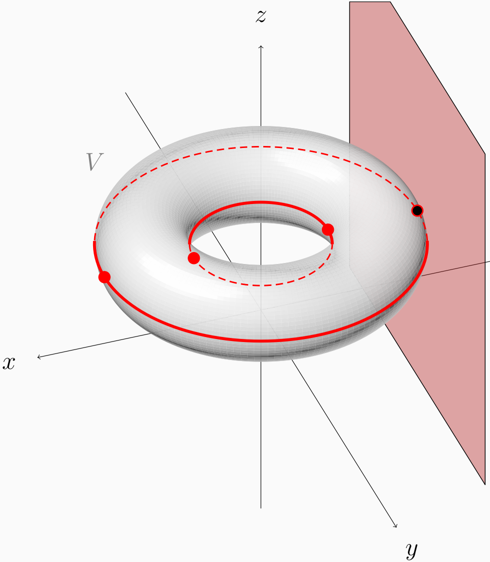
$\forall\, C$ connected component,
$C \cap \mathscr{R}$ is non-empty and connected

Projection through:
$\pi_2 \colon (x_1, \ldots, x_n) \mapsto (x_1, x_2)$

$W(\pi_2, V)$ critical locus of π_2.

Intersects all the
connected components of V

Canny's strategy



Roadmap property

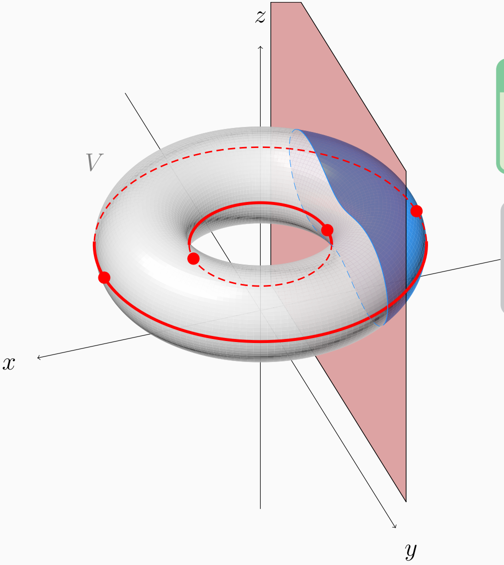
$\forall\, C$ connected component,
$C \cap \mathscr{R}$ is non-empty and connected

Morse theory

"Scan" $\boldsymbol{W(\pi_2, V)}$ at the critical values of π_1

- We repair the connectivity failures with critical fibers
- We repeat the process at every critical value

Canny's strategy



Roadmap property

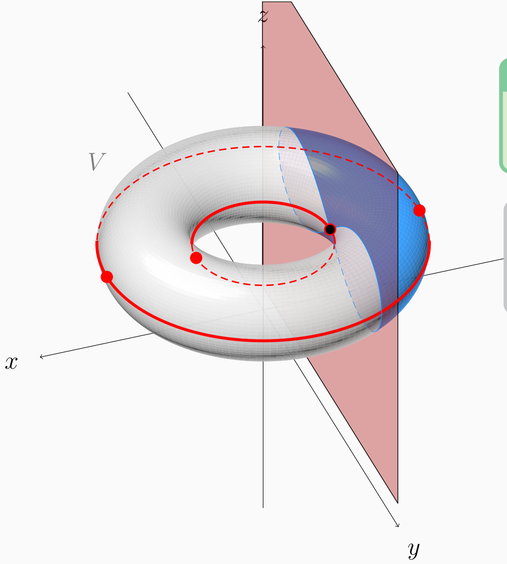
$\forall\, C$ connected component,
$C \cap \mathscr{R}$ is non-empty and connected

Morse theory

"Scan" $\boldsymbol{W(\pi_2, V)}$ at the critical values of π_1

- We repair the connectivity failures with critical fibers
- We repeat the process at every critical value

Canny's strategy



Roadmap property

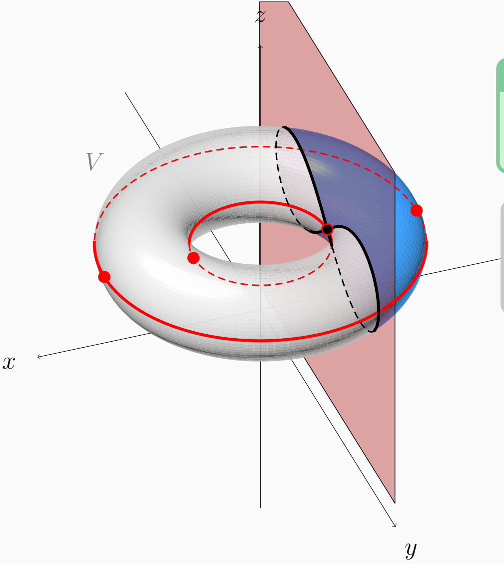
$\forall\, C$ connected component,
$C \cap \mathscr{R}$ is non-empty and connected

Morse theory

"Scan" $\boldsymbol{W(\pi_2, V)}$ at the critical values of π_1

- We repair the connectivity failures with critical fibers
- We repeat the process at every critical value

Canny's strategy



Roadmap property

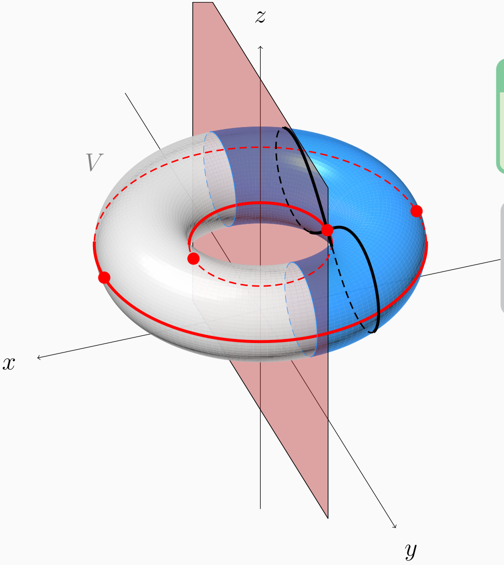
$\forall\, C$ connected component,
$C \cap \mathscr{R}$ is non-empty and connected

Morse theory

"Scan" $\boldsymbol{W(\pi_2, V)}$ at the critical values of π_1

- We repair the connectivity failures with critical fibers
- We repeat the process at every critical value

Canny's strategy



Roadmap property

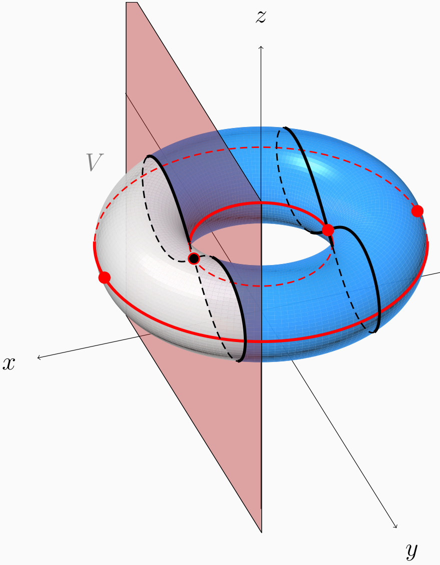
$\forall\, C$ connected component,
$C \cap \mathscr{R}$ is non-empty and connected

Morse theory

"Scan" $\boldsymbol{W(\pi_2, V)}$ at the critical values of π_1

- We repair the connectivity failures with critical fibers
- We repeat the process at every critical value

Canny's strategy



Roadmap property

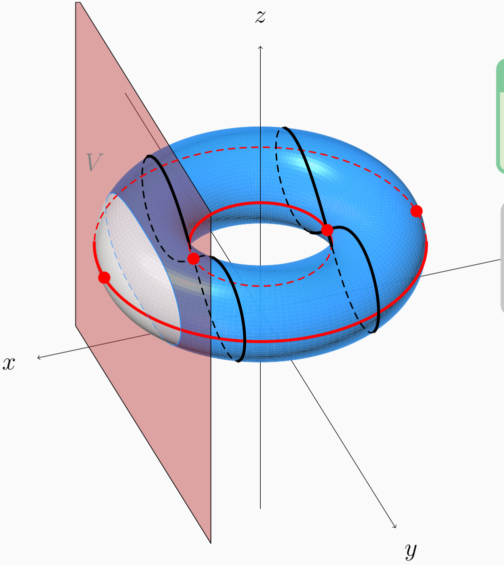
$\forall\, C$ connected component,
$C \cap \mathscr{R}$ is non-empty and connected

Morse theory

"Scan" $\boldsymbol{W(\pi_2, V)}$ at the critical values of π_1

- We repair the connectivity failures with critical fibers
- We repeat the process at every critical value

Canny's strategy



Roadmap property

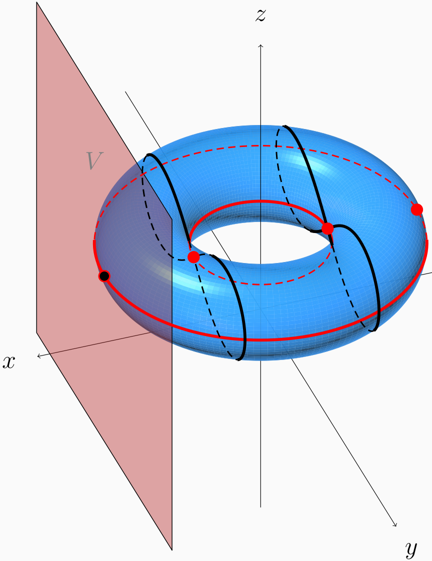
$\forall\, C$ connected component,
$C \cap \mathscr{R}$ is non-empty and connected

Morse theory

"Scan" $\boldsymbol{W(\pi_2, V)}$ at the critical values of π_1

- We repair the connectivity failures with critical fibers
- We repeat the process at every critical value

Canny's strategy



Roadmap property

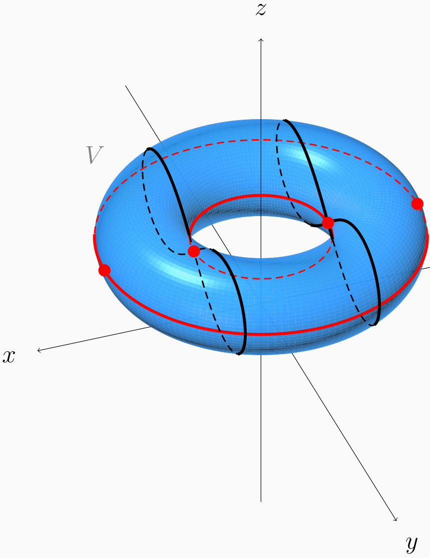
$\forall C$ connected component,
$C \cap \mathscr{R}$ is non-empty and connected

Morse theory

"Scan" $\boldsymbol{W(\pi_2, V)}$ at the critical values of π_1

- We repair the connectivity failures with critical fibers
- We repeat the process at every critical value

Canny's strategy



Roadmap property

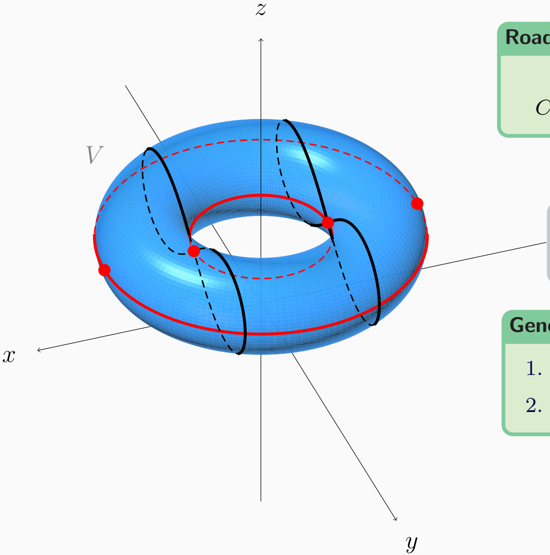
$\forall\, C$ connected component,

$C \cap \mathscr{R}$ is non-empty and connected

$\boldsymbol{W(\pi_2, V)}$ **polar variety**

$\boldsymbol{F}$ **critical fibers**

Canny's strategy



Roadmap property

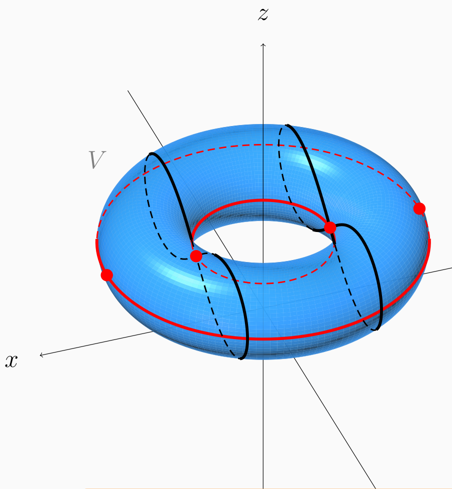
$\forall\, C$ connected component,
$C \cap \mathscr{R}$ is non-empty and connected

$\boldsymbol{W(\pi_2, V)}$ **polar variety**
$\boldsymbol{F}$ **critical fibers**

Genericity assumptions

1. $\boldsymbol{W(\pi_2, V)}$ has dimension 1
2. $\boldsymbol{F}$ has dimension $\dim(V) - 1$

Canny's strategy



Roadmap property

$\forall\, C$ connected component,
$C \cap \mathscr{R}$ is non-empty and connected

$\boldsymbol{W(\pi_2, V)}$ **polar variety**
$\boldsymbol{F}$ **critical fibers**

Genericity assumptions

1. $\boldsymbol{W(\pi_2, V)}$ has dimension 1
2. $\boldsymbol{F}$ has dimension $\dim(V) - 1$

Theorem [Canny, 1988]

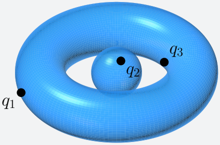
If V is bounded, $\boldsymbol{W(\pi_2, V)} \bigcup \boldsymbol{F}$ has dimension $\dim(V) - 1$
and satisfies the Roadmap property

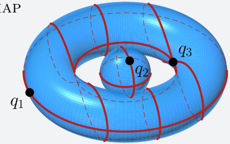
Connectivity queries, roadmaps

[Canny, 1988] **Roadmap**: curve capturing the connectivity of the set

Reduction chain

Arbitrary dimension $\overset{\Longrightarrow}{\text{ROADMAP}}$ Dimension 1



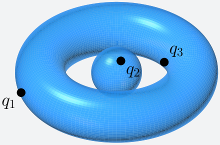


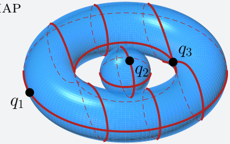
Connectivity queries, roadmaps

[Canny, 1988] **Roadmap**: curve capturing the connectivity of the set

Reduction chain

Arbitrary dimension $\underset{\text{ROADMAP}}{\Longrightarrow}$ Dimension 1





Towards optimal

1982 $D^{2^{O(n)}}$

$\downarrow$

1988 $(nD)^{O(n^2)}$

$\downarrow$

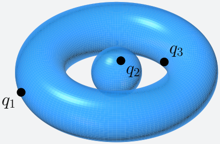
$(nD)^{O(n)}$

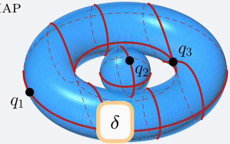
Connectivity queries, roadmaps

[Canny, 1988] **Roadmap**: curve capturing the connectivity of the set

Reduction chain

Arbitrary dimension $\xrightarrow[\text{ROADMAP}]{}$ Dimension 1





Towards optimal

1982 $D^{2^{O(n)}}$

↓

1988 $(nD)^{O(n^2)}$

↓

$(nD)^{O(n)}$

Complexity result (P., Safey El Din, Schost; 2024) (smooth & **unbounded** case)

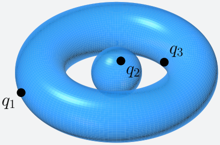
We can compute a roadmap of degree $\delta = (n^2 D)^{\mathbf{2n \log_2 n} + O(n)}$ in $O(\delta^3)$ arithmetic ops

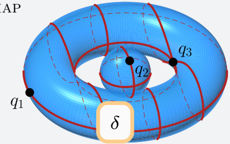
Connectivity queries, roadmaps

[Canny, 1988] **Roadmap**: curve capturing the connectivity of the set

Reduction chain

Arbitrary dimension $\xrightarrow{}$ ROADMAP Dimension 1





Towards optimal

1982 $D^{2^{O(n)}}$

↓

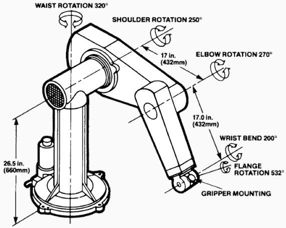
1988 $(nD)^{O(n^2)}$

↓

$(nD)^{O(n)}$

Complexity result (P., Safey El Din, Schost; 2024) (smooth & **unbounded** case)

We can compute a roadmap of degree $\delta = (n^2D)^{\mathbf{2n\log_2 n}+O(n)}$ in $O(\delta^3)$ arithmetic ops



PUMA 560 [Unimation, 1984]

Computation of a roadmap NEW!

<u>Output</u>: polynomials of max deg **200**
1 polynomial ≈ 20 000 coeffs
1 coeff ≈ 3000 digits

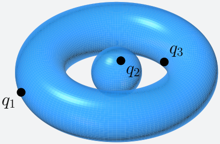
<u>Time</u>: **4h10**

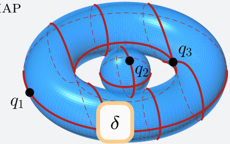
Connectivity queries, roadmaps

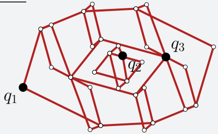
[Canny, 1988] **Roadmap**: curve capturing the connectivity of the set

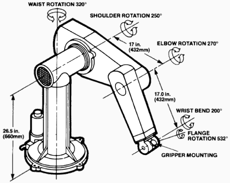
Reduction chain

Arbitrary dimension $\xrightarrow{\text{Roadmap}}$ Dimension 1 $\xrightarrow{\text{Analysis}}$ Graph









Puma 560 [Unimation, 1984]

Computation of a roadmap

Output: polynomials of max deg **200**
1 polynomial ≈ 20 000 coeffs
1 coeff ≈ 3000 digits

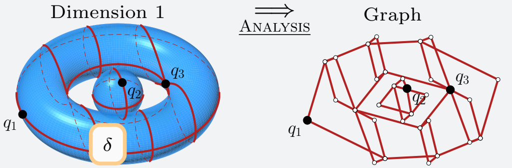
Time: **4h10**

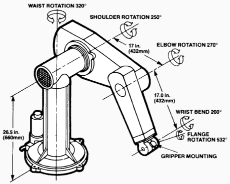
Connectivity queries, roadmaps

[Canny, 1988] **Roadmap**: curve capturing the connectivity of the set

State-of-the-art : geometry

Space	Complexity
$\mathbb{R}^2$	$O\tilde{}(\delta^6)$
$\mathbb{R}^3$	$O\tilde{}(\delta^{18})$
$\mathbb{R}^n$	$\delta^{O(1)}$





Puma 560 [Unimation, 1984]

Computation of a roadmap

Output: polynomials of max deg **200**
1 polynomial ≈ 20 000 coeffs
1 coeff ≈ 3000 digits

Time: **4h10**

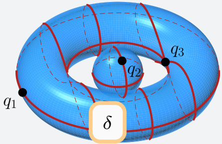
Connectivity queries, roadmaps

[Canny, 1988] **Roadmap**: curve capturing the connectivity of the set

State-of-the-art : geometry

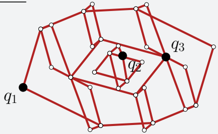
Space	Complexity
$\mathbb{R}^2$	$\tilde{O}(\delta^6)$
$\mathbb{R}^3$	$\tilde{O}(\delta^{18})$
$\mathbb{R}^n$	$\delta^{O(1)}$

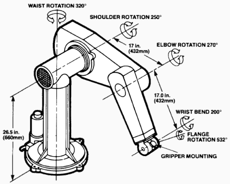
Dimension 1



$\xrightarrow{\text{ANALYSIS}}$

Graph





PUMA 560 [Unimation, 1984]

Computation of a roadmap

Output: polynomials of max deg **200**
1 polynomial $\approx$ 20 000 coeffs
1 coeff $\approx$ 3000 digits

Time: **4h10**

Avoid computing the complete geometry (Islam, Poteaux, P.; 2023)

We can compute a connectivity graph for a curve of degree δ in $\tilde{O}(\delta^6)$ bit operations

Data representation and quantitative estimate

Theorem

In a *generic* system of coordinates,
V is *birational* to a hypersurface of $\mathbb{C}^{d+1}$ through:

$$\pi_{d+1} \colon (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \mapsto (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{d+1})$$

$\longleftarrow$ V equidimensional of dimension d

Data representation and quantitative estimate

Theorem

In a *generic* system of coordinates,
V is *birational* to a hypersurface of $\mathbb{C}^{d+1}$ through:

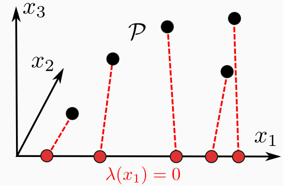
$$\pi_{d+1} \colon (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \mapsto (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{d+1})$$

$\longleftarrow$ V equidimensional of dimension d

Zero-dimensional parametrization of $\mathcal{P} \subset \mathbb{C}^n$ finite

$(\lambda, \vartheta_2, \ldots, \vartheta_n) \subset \mathbb{Z}[x_1]$ s.t.

$$\mathcal{P} = \left\{ \left(\boldsymbol{x}_1, \frac{\vartheta_2(\boldsymbol{x}_1)}{\lambda'(\boldsymbol{x}_1)}, \ldots, \frac{\vartheta_n(\boldsymbol{x}_1)}{\lambda'(\boldsymbol{x}_1)} \right) \text{ s.t. } \lambda(\boldsymbol{x}_1) = 0 \right\}$$



Data representation and quantitative estimate

Theorem

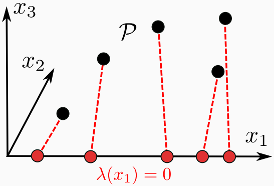
In a *generic* system of coordinates,
V is *birational* to a hypersurface of $\mathbb{C}^{d+1}$ through:
$\pi_{d+1} \colon (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \mapsto (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{d+1})$

$\longleftarrow$ V equidimensional of dimension d

Zero-dimensional parametrization of $\mathcal{P} \subset \mathbb{C}^n$ finite

$(\lambda, \vartheta_2, \ldots, \vartheta_n) \subset \mathbb{Z}[x_1]$ s.t.

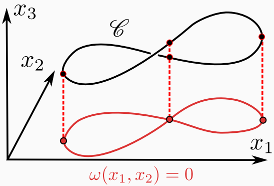
$$\mathcal{P} = \left\{ \left(\boldsymbol{x}_1, \frac{\vartheta_2(\boldsymbol{x}_1)}{\lambda'(\boldsymbol{x}_1)}, \ldots, \frac{\vartheta_n(\boldsymbol{x}_1)}{\lambda'(\boldsymbol{x}_1)} \right) \text{ s.t. } \lambda(\boldsymbol{x}_1) = 0 \right\}$$



One-dimensional parametrization of $\mathscr{C} \subset \mathbb{C}^n$ algebraic curve

$(\omega, \rho_3, \ldots, \rho_n) \subset \mathbb{Z}[x_1, x_2]$ s.t.

$$\mathscr{C} = \overline{\left\{ \begin{array}{c} \left(\boldsymbol{x}_1, \boldsymbol{x}_2, \frac{\rho_3(\boldsymbol{x}_1, \boldsymbol{x}_2)}{\partial_{x_2}\omega(\boldsymbol{x}_1, \boldsymbol{x}_2)}, \ldots, \frac{\rho_n(\boldsymbol{x}_1, \boldsymbol{x}_2)}{\partial_{x_2}\omega(\boldsymbol{x}_1, \boldsymbol{x}_2)} \right) \\ \text{s.t. } \omega(\boldsymbol{x}_1, \boldsymbol{x}_2) = 0 \quad \text{and} \quad \partial_{x_2}\omega(\boldsymbol{x}_1, \boldsymbol{x}_2) \neq 0 \end{array} \right\}}^{Z}$$



Data representation and quantitative estimate

Theorem

In a *generic* system of coordinates,
V is *birational* to a hypersurface of $\mathbb{C}^{d+1}$ through:

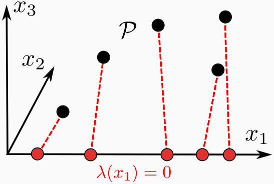
$$\pi_{d+1} \colon (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \mapsto (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{d+1})$$

$\longleftarrow$ V equidimensional of dimension d

Zero-dimensional parametrization of $\mathcal{P} \subset \mathbb{C}^n$ finite

$(\lambda, \vartheta_2, \ldots, \vartheta_n) \subset \mathbb{Z}[x_1]$ s.t.

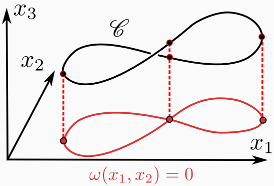
$$\mathcal{P} = \left\{ \left(\boldsymbol{x}_1, \frac{\vartheta_2(\boldsymbol{x}_1)}{\lambda'(\boldsymbol{x}_1)}, \ldots, \frac{\vartheta_n(\boldsymbol{x}_1)}{\lambda'(\boldsymbol{x}_1)} \right) \text{ s.t. } \lambda(\boldsymbol{x}_1) = 0 \right\}$$



One-dimensional parametrization of $\mathscr{C} \subset \mathbb{C}^n$ algebraic curve

$(\omega, \rho_3, \ldots, \rho_n) \subset \mathbb{Z}[x_1, x_2]$ s.t.

$$\mathscr{C} = \overline{\left\{ \begin{array}{c} \left(\boldsymbol{x}_1, \boldsymbol{x}_2, \frac{\rho_3(\boldsymbol{x}_1, \boldsymbol{x}_2)}{\partial_{x_2}\omega(\boldsymbol{x}_1, \boldsymbol{x}_2)}, \ldots, \frac{\rho_n(\boldsymbol{x}_1, \boldsymbol{x}_2)}{\partial_{x_2}\omega(\boldsymbol{x}_1, \boldsymbol{x}_2)} \right) \\ \text{s.t. } \omega(\boldsymbol{x}_1, \boldsymbol{x}_2) = 0 \quad \text{and} \quad \partial_{x_2}\omega(\boldsymbol{x}_1, \boldsymbol{x}_2) \neq 0 \end{array} \right\}}^{Z}$$



Magnitude of a polynomial

$\boldsymbol{f} \in \mathbb{Z}[x_1, \ldots, x_n]$ has *magnitude* (δ, τ) if

$\deg(\boldsymbol{f}) \leq \delta$ and $|\mathrm{coeffs}(\boldsymbol{f})| \leq 2^{\tau}$

Soft-O notation

$$\tilde{O}(N) = O(N \log(N)^a)$$

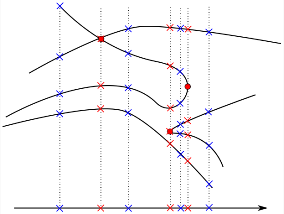
Results

Data

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^n$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;

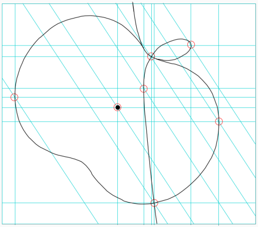
Computing topology

Ambient dimension	Bit complexity	Reference
$n = 2$	$\tilde{O}(\delta^5(\delta + \tau))$	[Kobel, Sagraloff; '15] [D.Diatta, S.Diatta, Rouiller, Roy, Sagraloff; '22]



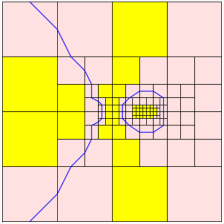
Cylindrical Algebraic Decomposition

[Collins, '75] [Kerber, Sagraloff; '12]



Multiple projections

[Seidel, Wolpert; '05]



Subdivision

[Burr, Choi, Galehouse, Yap; '05]

Results

Data

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^{\boldsymbol{n}}$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;

Computing topology

Ambient dimension	Bit complexity	Reference
$n = 2$	$\tilde{O}(\delta^5(\delta + \tau))$	[Kobel, Sagraloff; '15] [D.Diatta, S.Diatta, Rouiller, Roy, Sagraloff; '22]
$n = 3$	$\tilde{O}(\delta^{17}(\delta + \tau))$	[Cheng, Jin, Pouget, Wen, Zhang; '21]

Results

Data

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^{\boldsymbol{n}}$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;

Computing topology

Ambient dimension	Bit complexity	Reference
$n = 2$	$\tilde{O}(\delta^5(\delta + \tau))$	[Kobel, Sagraloff; '15] [D.Diatta, S.Diatta, Rouiller, Roy, Sagraloff; '22]
$n = 3$	$\tilde{O}(\delta^{17}(\delta + \tau))$	[Cheng, Jin, Pouget, Wen, Zhang; '21]
$n > 3$	$\tilde{O}(\delta^{O(1)}(\delta + \tau))$	[Safey El Din, Schost; '11]

Results

Data

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^{\boldsymbol{n}}$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;

Computing topology

Ambient dimension	Bit complexity	Reference
$n = 2$	$\tilde{O}(\delta^5(\delta + \tau))$	[Kobel, Sagraloff; '15] [D.Diatta, S.Diatta, Rouiller, Roy, Sagraloff; '22]
$n = 3$	$\tilde{O}(\delta^{17}(\delta + \tau))$	[Cheng, Jin, Pouget, Wen, Zhang; '21]
$n > 3$	$\tilde{O}(\delta^{O(1)}(\delta + \tau))$	[Safey El Din, Schost; '11]

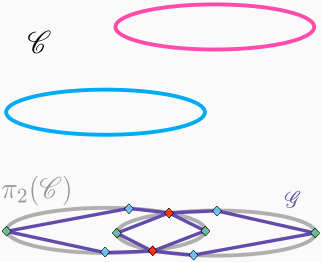
Results

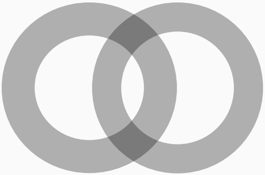
Data

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^n$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;

Computing topology

Ambient dimension	Bit complexity	Reference
$n = 2$	$\tilde{O}(\delta^5(\delta+\tau))$	[Kobel, Sagraloff; '15] [D.Diatta, S.Diatta, Rouiller, Roy, Sagraloff; '22]
$n = 3$	$\tilde{O}(\delta^{17}(\delta+\tau))$	[Cheng, Jin, Pouget, Wen, Zhang; '21]
$n > 3$	$\tilde{O}(\delta^{O(1)}(\delta+\tau))$	[Safey El Din, Schost; '11]





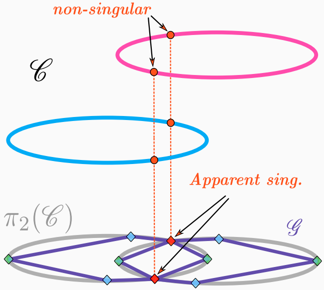
Results

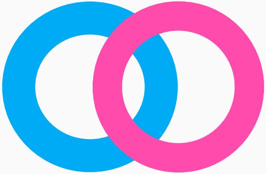
Data

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^{\boldsymbol{n}}$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;

Computing topology

Ambient dimension	Bit complexity	Reference
$n = 2$	$\tilde{O}(\delta^5(\delta + \tau))$	[Kobel, Sagraloff; '15] [D.Diatta, S.Diatta, Rouiller, Roy, Sagraloff; '22]
$n = 3$	$\tilde{O}(\delta^{17}(\delta + \tau))$	[Cheng, Jin, Pouget, Wen, Zhang; '21]
$n > 3$	$\tilde{O}(\delta^{O(1)}(\delta + \tau))$	[Safey El Din, Schost; '11]





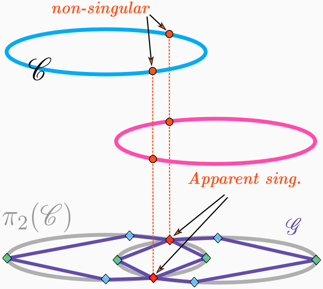
Results

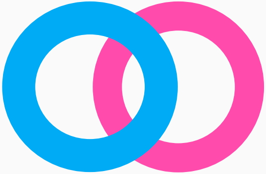
Data

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^n$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;

Computing topology

Ambient dimension	Bit complexity	Reference
$n = 2$	$\tilde{O}(\delta^5(\delta + \tau))$	[Kobel, Sagraloff; '15] [D.Diatta, S.Diatta, Rouiller, Roy, Sagraloff; '22]
$n = 3$	$\tilde{O}(\delta^{17}(\delta + \tau))$	[Cheng, Jin, Pouget, Wen, Zhang; '21]
$n > 3$	$\tilde{O}(\delta^{O(1)}(\delta + \tau))$	[Safey El Din, Schost; '11]





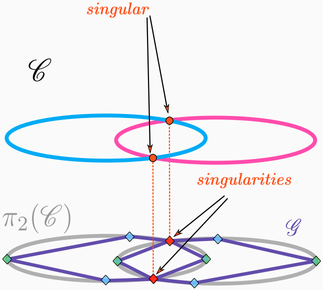
Results

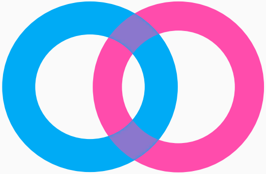
Data

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^{\boldsymbol{n}}$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;

Computing topology

Ambient dimension	Bit complexity	Reference
$n = 2$	$\tilde{O}(\delta^5(\delta + \tau))$	[Kobel, Sagraloff; '15] [D.Diatta, S.Diatta, Rouiller, Roy, Sagraloff; '22]
$n = 3$	$\tilde{O}(\delta^{17}(\delta + \tau))$	[Cheng, Jin, Pouget, Wen, Zhang; '21]
$n > 3$	$\tilde{O}(\delta^{O(1)}(\delta + \tau))$	[Safey El Din, Schost; '11]





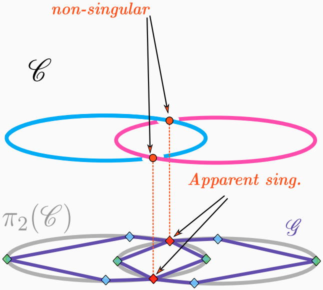
Results

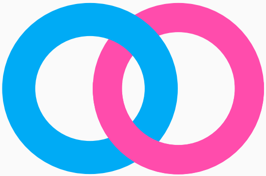
Data

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^{\boldsymbol{n}}$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;

Computing topology

Ambient dimension	Bit complexity	Reference
$n = 2$	$\tilde{O}(\delta^5(\delta + \tau))$	[Kobel, Sagraloff; '15] [D.Diatta, S.Diatta, Rouiller, Roy, Sagraloff; '22]
$n = 3$	$\tilde{O}(\delta^{17}(\delta + \tau))$	[Cheng, Jin, Pouget, Wen, Zhang; '21]
$n > 3$	$\tilde{O}(\delta^{O(1)}(\delta + \tau))$	[Safey El Din, Schost; '11]





Results

Data

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^n$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;

Computing topology

Ambient dimension	Bit complexity	Reference
$n = 2$	$\tilde{O}(\delta^5(\delta + \tau))$	[Kobel, Sagraloff; '15] [D.Diatta, S.Diatta, Rouiller, Roy, Sagraloff; '22]
$n = 3$	$\tilde{O}(\delta^{17}(\delta + \tau))$	[Cheng, Jin, Pouget, Wen, Zhang; '21]
$n > 3$	$\tilde{O}(\delta^{O(1)}(\delta + \tau))$	[Safey El Din, Schost; '11]

Computing connectivity - Main Result NEW!

Ambient dimension	Bit complexity	Reference
$n \geq 2$	$\tilde{O}(\delta^5(\delta + \tau))$	[Islam, Poteaux, P.; 2023]

Avoid computation of the complete topology!

Algorithm

Input

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^n$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;
- `[some assumptions]`

Output

A connectivity graph of $\mathscr{C} \cap \mathbb{R}^n$ capturing the points in $\mathcal{P} \cap \mathbb{R}^n$.

1. $\mathscr{D}, \mathscr{Q} \leftarrow \mathsf{Proj2D}(\mathscr{R}), \mathsf{Proj2D}(\mathscr{P})$;
2. $\mathscr{G} \leftarrow \mathsf{Topo2D}(\mathscr{D}, \mathscr{Q})$;
3. $\mathscr{Q}_{\mathrm{app}} \leftarrow \mathsf{ApparentSingularities}(\mathscr{R})$;
4. $\mathscr{G}' \leftarrow \mathsf{NodeResolution}(\mathscr{G}, \mathscr{Q}_{\mathrm{app}})$;
5. `return` $\mathsf{ConnectGraph}(\mathscr{Q}, \mathscr{G}')$;

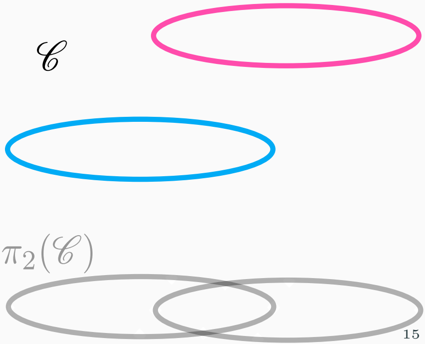
Input

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^n$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;
- `[some assumptions]`

Output

A connectivity graph of $\mathscr{C} \cap \mathbb{R}^n$ capturing the points in $\mathcal{P} \cap \mathbb{R}^n$.

1. $\mathscr{D}, \mathscr{Q} \leftarrow \mathsf{Proj2D}(\mathscr{R}), \mathsf{Proj2D}(\mathscr{P})$;
2. $\mathscr{G} \leftarrow \mathsf{Topo2D}(\mathscr{D}, \mathscr{Q})$;
3. $\mathscr{Q}_{\mathrm{app}} \leftarrow \mathsf{ApparentSingularities}(\mathscr{R})$;
4. $\mathscr{G}' \leftarrow \mathsf{NodeResolution}(\mathscr{G}, \mathscr{Q}_{\mathrm{app}})$;
5. `return` $\mathsf{ConnectGraph}(\mathscr{Q}, \mathscr{G}')$;



Algorithm

Input

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^n$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;
- `[some assumptions]`

Output

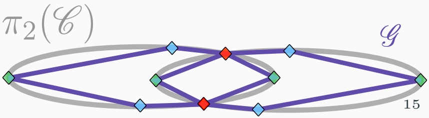
A connectivity graph of $\mathscr{C} \cap \mathbb{R}^n$ capturing the points in $\mathcal{P} \cap \mathbb{R}^n$.

1. $\mathscr{D}, \mathscr{Q} \leftarrow \mathsf{Proj2D}(\mathscr{R}), \mathsf{Proj2D}(\mathscr{P})$;
2. $\mathscr{G} \leftarrow \mathsf{Topo2D}(\mathscr{D}, \mathscr{Q})$;
3. $\mathscr{Q}_{\mathrm{app}} \leftarrow \mathsf{ApparentSingularities}(\mathscr{R})$;
4. $\mathscr{G}' \leftarrow \mathsf{NodeResolution}(\mathscr{G}, \mathscr{Q}_{\mathrm{app}})$;
5. `return` $\mathsf{ConnectGraph}(\mathscr{Q}, \mathscr{G}')$;

$\rightsquigarrow$ resultants
$\rightsquigarrow$ $\mathbb{R}$-root isolation
$\looparrowright$ univariate
$\looparrowright$ bivariate triangular

Planar topology

$\tilde{O}(\delta^5(\delta + \tau))$



Algorithm

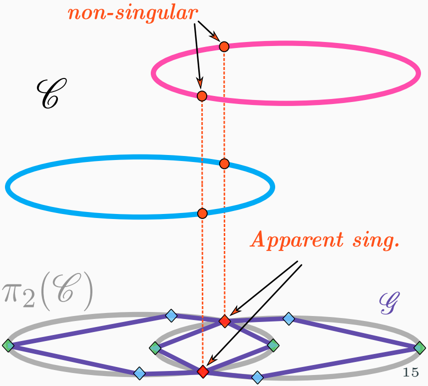
Input

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^n$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;
- `[some assumptions]`

Output

A connectivity graph of $\mathscr{C} \cap \mathbb{R}^n$ capturing the points in $\mathcal{P} \cap \mathbb{R}^n$.

1. $\mathscr{D}, \mathscr{Q} \leftarrow \mathsf{Proj2D}(\mathscr{R}), \mathsf{Proj2D}(\mathscr{P})$;
2. $\mathscr{G} \leftarrow \mathsf{Topo2D}(\mathscr{D}, \mathscr{Q})$;
3. $\mathscr{Q}_{\mathrm{app}} \leftarrow \mathsf{ApparentSingularities}(\mathscr{R})$;
4. $\mathscr{G}' \leftarrow \mathsf{NodeResolution}(\mathscr{G}, \mathscr{Q}_{\mathrm{app}})$;
5. `return` $\mathsf{ConnectGraph}(\mathscr{Q}, \mathscr{G}')$;



Input

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^n$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;
- `[some assumptions]`

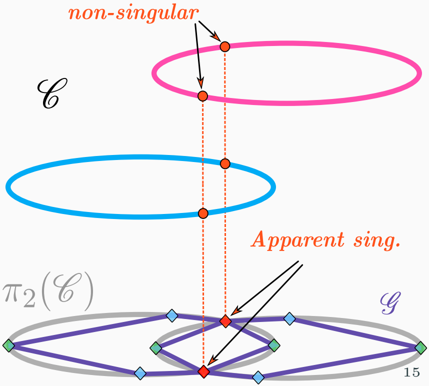
Output

A connectivity graph of $\mathscr{C} \cap \mathbb{R}^n$ capturing the points in $\mathcal{P} \cap \mathbb{R}^n$.

1. $\mathscr{D}, \mathscr{Q} \leftarrow$ Proj2D($\mathscr{R}$), Proj2D($\mathscr{P}$);
2. $\mathscr{G} \leftarrow$ Topo2D($\mathscr{D}, \mathscr{Q}$);
3. $\mathscr{Q}_{\text{app}} \leftarrow$ ApparentSingularities($\mathscr{R}$);
4. $\mathscr{G}' \leftarrow$ NodeResolution($\mathscr{G}$, $\mathscr{Q}_{\text{app}}$);
5. `return` ConnectGraph($\mathscr{Q}$, $\mathscr{G}'$);

Apparent sing. identification

$\tilde{O}(\delta^5(\delta + \tau))$



Algorithm

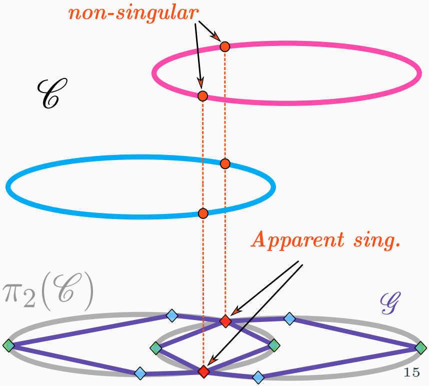
Input

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^n$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;
- `[some assumptions]`

Output

A connectivity graph of $\mathscr{C} \cap \mathbb{R}^n$ capturing the points in $\mathcal{P} \cap \mathbb{R}^n$.

1. $\mathscr{D}, \mathscr{Q} \leftarrow \mathsf{Proj2D}(\mathscr{R}), \mathsf{Proj2D}(\mathscr{P})$;
2. $\mathscr{G} \leftarrow \mathsf{Topo2D}(\mathscr{D}, \mathscr{Q})$;
3. $\mathscr{Q}_{\mathrm{app}} \leftarrow \mathsf{ApparentSingularities}(\mathscr{R})$;
4. $\mathscr{G}' \leftarrow \mathsf{NodeResolution}(\mathscr{G}, \mathscr{Q}_{\mathrm{app}})$;
5. `return` $\mathsf{ConnectGraph}(\mathscr{Q}, \mathscr{G}')$;



Algorithm

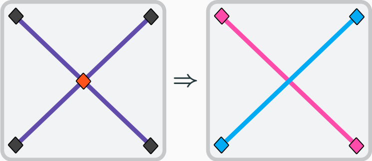
Input

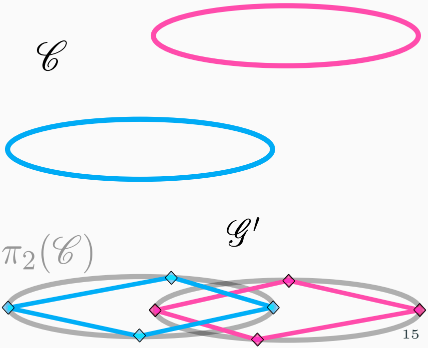
- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^n$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;
- `[some assumptions]`

Output

A connectivity graph of $\mathscr{C} \cap \mathbb{R}^n$ capturing the points in $\mathcal{P} \cap \mathbb{R}^n$.

1. $\mathscr{D}, \mathscr{Q} \leftarrow \mathsf{Proj2D}(\mathscr{R}), \mathsf{Proj2D}(\mathscr{P})$;
2. $\mathscr{G} \leftarrow \mathsf{Topo2D}(\mathscr{D}, \mathscr{Q})$;
3. $\mathscr{Q}_{\mathrm{app}} \leftarrow \mathsf{ApparentSingularities}(\mathscr{R})$;
4. $\mathscr{G}' \leftarrow \mathsf{NodeResolution}(\mathscr{G}, \mathscr{Q}_{\mathrm{app}})$;
5. `return` $\mathsf{ConnectGraph}(\mathscr{Q}, \mathscr{G}')$;





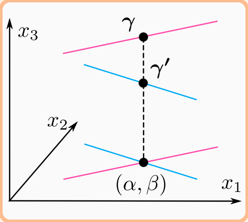
Apparent singularities: key idea

- $\mathscr{R} = (\boldsymbol{\omega}, \boldsymbol{\rho_3}, \dots, \rho_n) \subset \mathbb{Z}[x_1, x_2]$ encoding $\mathscr{C} \subset \mathbb{C}^n$ in generic coordinates;
- $\mathcal{A}(x_1, x_2) = \partial^2_{x_2}\boldsymbol{\omega} \cdot \partial_{x_1}\boldsymbol{\rho_3} - \partial^2_{x_1 x_2}\boldsymbol{\omega} \cdot \partial_{x_2}\boldsymbol{\rho_3} \in \mathbb{Z}[x_1, x_2]$

Generic apparent singularities

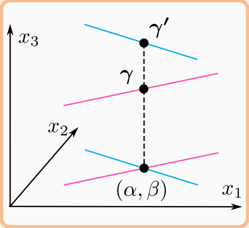
NEW!

Generic projection only introduce finitely many nodes:



Below

$\neq$

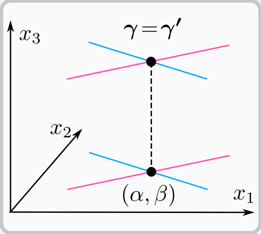


Above

$\neq$

Space singularities

Spatial nodes project as:



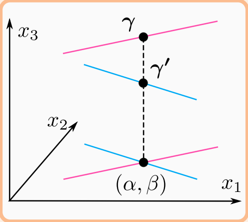
Same

Apparent singularities: key idea

- $\mathscr{R} = (\omega, \rho_3, \dots, \rho_n) \subset \mathbb{Z}[x_1, x_2]$ encoding $\mathscr{C} \subset \mathbb{C}^n$ in generic coordinates;
- $\mathcal{A}(x_1, x_2) = \partial^2_{x_2}\omega \cdot \partial_{x_1}\rho_3 - \partial^2_{x_1 x_2}\omega \cdot \partial_{x_2}\rho_3 \in \mathbb{Z}[x_1, x_2]$

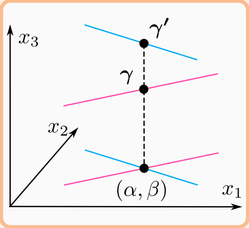
Generic apparent singularities NEW!

Generic projection only introduce finitely many nodes:



Below

$\neq$

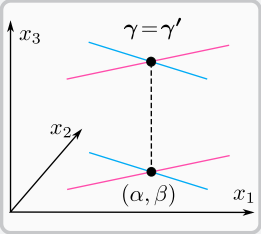


Above

$\neq$

Space singularities

Spatial nodes project as:



Same

$\boxed{\mathcal{A}(\alpha, \beta) > 0}$ $\boxed{\mathcal{A}(\alpha, \beta) < 0}$ $\boxed{\mathcal{A}(\alpha, \beta) = 0}$

Proposition - Generalization of [El Kahoui; '08]

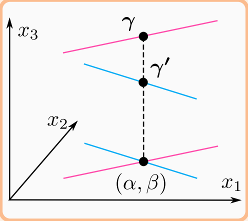
If (α, β) is a node then $\mathcal{A}(\alpha, \beta) = \gamma - \gamma'$

Apparent singularities: key idea

- $\mathscr{R} = (\boldsymbol{\omega}, \boldsymbol{\rho_3}, \ldots, \rho_n) \subset \mathbb{Z}[x_1, x_2]$ encoding $\mathscr{C} \subset \mathbb{C}^n$ in generic coordinates;
- $\mathcal{A}(x_1, x_2) = \partial^2_{x_2}\boldsymbol{\omega} \cdot \partial_{x_1}\boldsymbol{\rho_3} - \partial^2_{x_1 x_2}\boldsymbol{\omega} \cdot \partial_{x_2}\boldsymbol{\rho_3} \in \mathbb{Z}[x_1, x_2]$

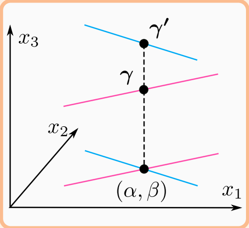
Generic apparent singularities NEW!

Generic projection only introduce finitely many nodes:



Apparent singularity

$\approx$



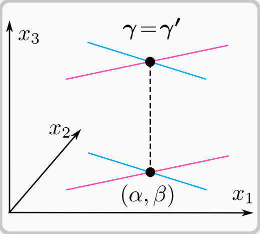
Apparent singularity

$$\boldsymbol{\mathcal{A}(\alpha, \beta) \neq 0}$$

$\not\approx$

Space singularities

Spatial nodes project as:



Actual singularity

$$\boldsymbol{\mathcal{A}(\alpha, \beta) = 0}$$

Proposition - Generalization of [El Kahoui; '08]

If (α, β) is a node then $\mathcal{A}(\alpha, \beta) = \gamma - \gamma'$

Algorithm

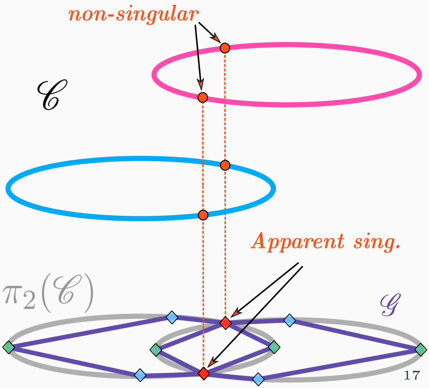
Input

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^n$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;
- $\mathscr{C}$ satisfies genericity assumptions w.r.t. $\mathcal{P}$

Output

A connectivity graph of $\mathscr{C} \cap \mathbb{R}^n$ capturing the points in $\mathcal{P} \cap \mathbb{R}^n$.

1. $\mathscr{D}, \mathscr{Q} \leftarrow \mathsf{Proj2D}(\mathscr{R}), \mathsf{Proj2D}(\mathscr{P})$;
2. $\mathscr{G} \leftarrow \mathsf{Topo2D}(\mathscr{D}, \mathscr{Q})$;
3. $\mathscr{Q}_{\mathrm{app}} \leftarrow \mathsf{ApparentSingularities}(\mathscr{R})$;
4. $\mathscr{G}' \leftarrow \mathsf{NodeResolution}(\mathscr{G}, \mathscr{Q}_{\mathrm{app}})$;
5. `return` $\mathsf{ConnectGraph}(\mathscr{Q}, \mathscr{G}')$;



Algorithm

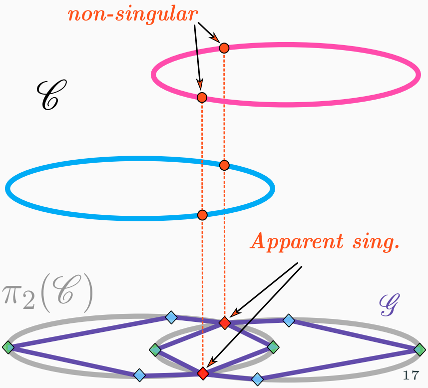
Input

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^n$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;
- $\mathscr{C}$ satisfies genericity assumptions w.r.t. $\mathcal{P}$

Output

A connectivity graph of $\mathscr{C} \cap \mathbb{R}^n$ capturing the points in $\mathcal{P} \cap \mathbb{R}^n$.

1. $\mathscr{D}, \mathscr{Q} \leftarrow \mathsf{Proj2D}(\mathscr{R}), \mathsf{Proj2D}(\mathscr{P})$;
2. $\mathscr{G} \leftarrow \mathsf{Topo2D}(\mathscr{D}, \mathscr{Q})$;
3. $\mathscr{Q}_{\mathrm{app}} \leftarrow \mathsf{ApparentSingularities}(\mathscr{R})$;
4. $\mathscr{G}' \leftarrow \mathsf{NodeResolution}(\mathscr{G}, \mathscr{Q}_{\mathrm{app}})$;
5. `return` $\mathsf{ConnectGraph}(\mathscr{Q}, \mathscr{G}')$;



⇝ subresultant seqs
⇝ GCD computations
⇝ multi-modularization

Apparent sing. identification

$\tilde{O}(\delta^5(\delta + \tau))$

Algorithm

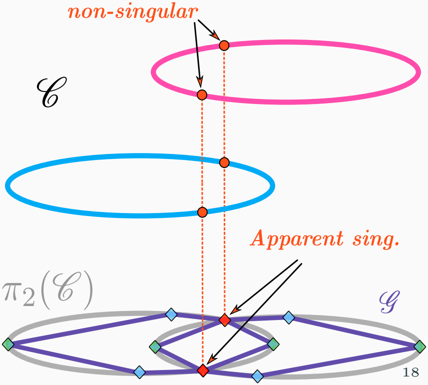
Input

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^n$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;
- $\mathscr{C}$ satisfies genericity assumptions w.r.t. $\mathcal{P}$

Output

A connectivity graph of $\mathscr{C} \cap \mathbb{R}^n$ capturing the points in $\mathcal{P} \cap \mathbb{R}^n$.

1. $\mathscr{D}, \mathscr{Q} \leftarrow \mathsf{Proj2D}(\mathscr{R}), \mathsf{Proj2D}(\mathscr{P})$;
2. $\mathscr{G} \leftarrow \mathsf{Topo2D}(\mathscr{D}, \mathscr{Q})$;
3. $\mathscr{Q}_{\text{app}} \leftarrow \mathsf{ApparentSingularities}(\mathscr{R})$;
4. $\mathscr{G}' \leftarrow \mathsf{NodeResolution}(\mathscr{G}, \mathscr{Q}_{\text{app}})$;
5. `return` $\mathsf{ConnectGraph}(\mathscr{Q}, \mathscr{G}')$;



Algorithm

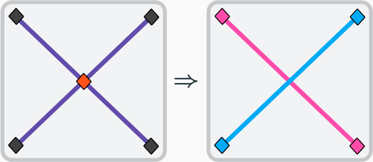
Input

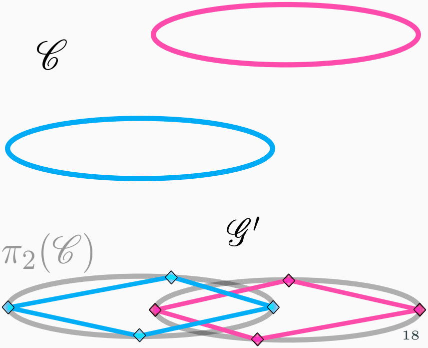
- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^n$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;
- $\mathscr{C}$ satisfies genericity assumptions w.r.t. $\mathcal{P}$

Output

A connectivity graph of $\mathscr{C} \cap \mathbb{R}^n$ capturing the points in $\mathcal{P} \cap \mathbb{R}^n$.

1. $\mathscr{D}, \mathscr{Q} \leftarrow \mathsf{Proj2D}(\mathscr{R}), \mathsf{Proj2D}(\mathscr{P})$;
2. $\mathscr{G} \leftarrow \mathsf{Topo2D}(\mathscr{D}, \mathscr{Q})$;
3. $\mathscr{Q}_{\mathrm{app}} \leftarrow \mathsf{ApparentSingularities}(\mathscr{R})$;
4. $\mathscr{G}' \leftarrow \mathsf{NodeResolution}(\mathscr{G}, \mathscr{Q}_{\mathrm{app}})$;
5. `return` $\mathsf{ConnectGraph}(\mathscr{Q}, \mathscr{G}')$;





Algorithm

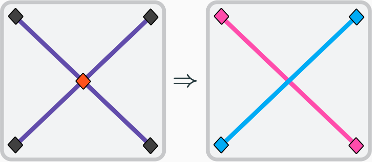
Input

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^n$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;
- $\mathscr{C}$ satisfies genericity assumptions w.r.t. $\mathcal{P}$

Output

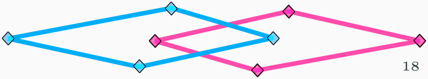
A connectivity graph of $\mathscr{C} \cap \mathbb{R}^n$ capturing the points in $\mathcal{P} \cap \mathbb{R}^n$.

1. $\mathscr{D}, \mathscr{Q} \leftarrow \mathsf{Proj2D}(\mathscr{R}), \mathsf{Proj2D}(\mathscr{P})$;
2. $\mathscr{G} \leftarrow \mathsf{Topo2D}(\mathscr{D}, \mathscr{Q})$;
3. $\mathscr{Q}_{\text{app}} \leftarrow \mathsf{ApparentSingularities}(\mathscr{R})$;
4. $\mathscr{G}' \leftarrow \mathsf{NodeResolution}(\mathscr{G}, \mathscr{Q}_{\text{app}})$;
5. `return` $\mathsf{ConnectGraph}(\mathscr{Q}, \mathscr{G}')$;



$\mathscr{C}$

$\mathscr{G}'$



Algorithm

Input

- $\mathscr{R} \subset \mathbb{Z}[x_1, x_2]$ of magnitude (δ, τ), encoding an algebraic curve $\mathscr{C} \subset \mathbb{C}^n$;
- $\mathscr{P} \subset \mathbb{Z}[x_1]$ of magnitude (δ, τ), encoding a finite $\mathcal{P} \subset \mathscr{C}$;
- $\mathscr{C}$ satisfies genericity assumptions w.r.t. $\mathcal{P}$

Output

A connectivity graph of $\mathscr{C} \cap \mathbb{R}^n$ capturing the points in $\mathcal{P} \cap \mathbb{R}^n$.

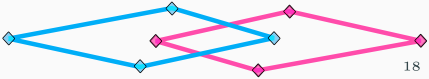
1. $\mathscr{D}, \mathscr{Q} \leftarrow \mathsf{Proj2D}(\mathscr{R}), \mathsf{Proj2D}(\mathscr{P})$;
2. $\mathscr{G} \leftarrow \mathsf{Topo2D}(\mathscr{D}, \mathscr{Q})$;
3. $\mathscr{Q}_{\mathrm{app}} \leftarrow \mathsf{ApparentSingularities}(\mathscr{R})$;
4. $\mathscr{G}' \leftarrow \mathsf{NodeResolution}(\mathscr{G}, \mathscr{Q}_{\mathrm{app}})$;
5. `return` $\mathsf{ConnectGraph}(\mathscr{Q}, \mathscr{G}')$;

$\mathscr{C}$

Overall Complexity

$\tilde{O}(\delta^5(\delta + \tau))$

$\mathscr{G}'$



Summary

Input

Polynomials in $\mathbb{Q}[x_1, \dots x_n]$ of max degree D defining a smooth algebraic set of dim. d

Connectivity reduction process - before

$$\boxed{\text{Arbitrary dimension}} \xrightarrow{\text{Roadmap}} \boxed{\begin{array}{l}\text{Dimension: } 1\\ \text{Size: } (nD)^{O(n\log(n))}\end{array}} \xrightarrow{\text{Topology}} \boxed{\text{Finite graph } \mathscr{G}}$$

$$\downarrow \quad \boxed{(nD)^{O(n\log^2(n))}} \qquad \downarrow \quad \boxed{(\text{Size})^{O(1)} = (nD)^{O(n\log(n))}}$$

[Basu, Roy; 2014] [Safey El Din, Schost; 2011]

Connectivity reduction process - now

$$\boxed{\text{Arbitrary dimension}} \xrightarrow{\text{Roadmap}} \boxed{\begin{array}{l}\text{Dimension: } 1\\ \text{Size: } (n^2D)^{4n\log_2(d)+O(n)}\end{array}} \xrightarrow{\text{Connectivity}} \boxed{\text{Finite graph } \mathscr{G}}$$

$$\downarrow \quad \boxed{(n^2D)^{6n\log_2(d)+O(n)}} \qquad \downarrow \quad \boxed{(\text{Size})^3 = (n^2D)^{12n\log_2(d)+O(n)}}$$

[P., Safey El Din, Schost; 2024] [Islam, Poteaux, P.; 2023]

Computing roadmaps in unbounded smooth real algebraic sets I: connectivity results, 2024
with M. Safey El Din and É. Schost

Computing roadmaps in unbounded smooth real algebraic sets II: algorithm and complexity, 2024
with M. Safey El Din and É. Schost

Algorithm for connectivity queries on real algebraic curves, 2023
with Md N. Islam and A. Poteaux

Perspectives

Algorithms

Roadmap algorithms:

- Adapt the algorithms to structured systems: quadratic case (J.A.K.Elliott, M.Safey El Din, É.Schost)
- Reduce the size of the roadmap by taking fewer fibers (M.Safey El Din, É.Schost)
- Generalize the connectivity result to semi-algebraic sets
- Design optimal roadmap algorithms with complexity exponential in $O(n)$

Connectivity of s.a. curves:

- Obtain a deterministic version of the algorithm (F.Bréhard, A.Poteaux)
- Adapt to algebraic curves given as union (A.Poteaux)
- Generalize to semi-algebraic curves
- Investigate the connectivity of plane curves

Applications

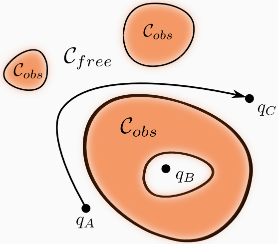
- Analyze challenging class of robots (D.Salunkhe, P.Wenger)
- Algorithms for rigidity and program verification problems (E.Bayarmagnai, F.Mohammadi)
- Obtain practical version of modern roadmap algorithms

Software

- Connectivity of curves: subresultant/GCD computations deg ~ 100 (now) $\to \sim 200$ (target)
- Build a Julia library for computational real algebraic geometry (C.Eder, R.Mohr)
- Implement a ready-to-use toolbox for roboticians

Motion planning with parameters and obstacles

Obstacles/Self-intersections

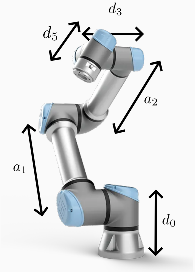


Inequalities⇒ semi-algebraic

Cuspidality

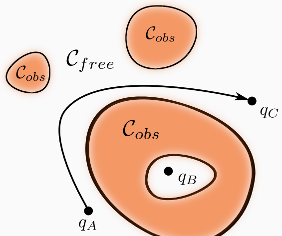
Parametric classification
↬design guidelines

Family of robots



Parametric systems

Obstacles/Self-intersections



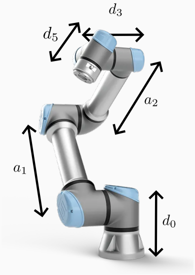
Inequalities$\Rightarrow$ semi-algebraic

Cuspidality

Parametric classification
$\hookrightarrow$design guidelines

Subroutines
efficient in practice

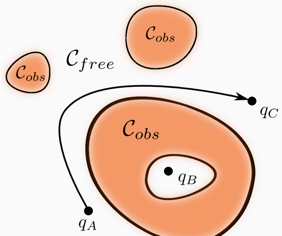
Family of robots



Parametric systems

Motion planning with parameters and obstacles

Obstacles/Self-intersections



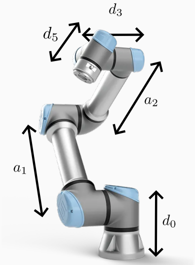
Inequalities$\Rightarrow$ semi-algebraic

Cuspidality

Parametric classification
$\leadsto$design guidelines

Subroutines
efficient in practice

Family of robots



Parametric systems

Roadmaps : structured semi-algebraic case

$$f_1 = \cdots = f_p = 0, \quad g_1 > 0, \ldots, g_s > 0$$

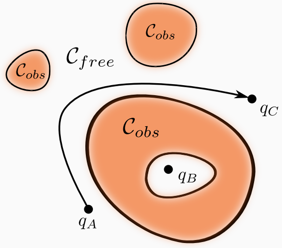
Structures: quadratic, multi/quasi-homogeneous

Current: $s^n(nD)^{O(n^2)}$ [Basu, Pollack, Roy ; 2000]

Goal: $s^n(n^2D)^{4n\log_2 n + O(n)}$

Motion planning with parameters and obstacles

Obstacles/Self-intersections



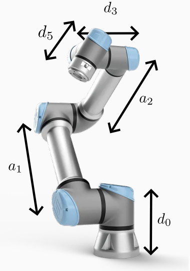
Inequalities ⇒ semi-algebraic

Cuspidality

Parametric classification
↪ design guidelines

Subroutines
efficient in practice

Family of robots



Parametric systems

Roadmaps : structured semi-algebraic case

$$f_1 = \cdots = f_p = 0, \quad g_1 > 0, \ldots, g_s > 0$$

Structures: quadratic, multi/quasi-homogeneous

Current: $s^n (nD)^{O(n^2)}$ [Basu, Pollack, Roy ; 2000]

Goal: $\boldsymbol{s^n} (n^2 D)^{4n \log_2 n + O(n)}$

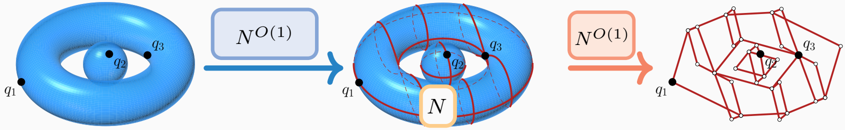
Implementation

AlgebraicSolving.jl
↙ `msolve` ↘
ℝ Isolation elimination

New situation

Analysing the road map becomes the most costly step

Dimension reduction

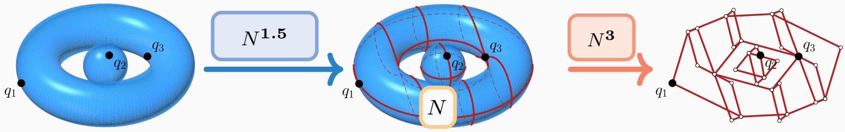


Geometry of curves

New situation

Analysing the road map becomes the most costly step

Dimension reduction

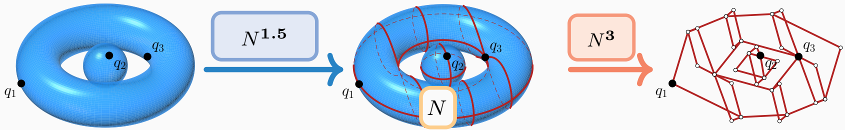


Geometry of curves

New situation

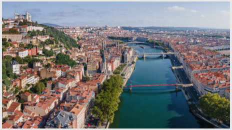
Analysing the road map becomes the most costly step

Dimension reduction



Bottleneck

Algebraic elimination (resultant)

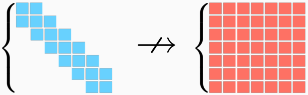


with G. Villard

Structures des cartes routières

Algebraic

Avoid change of variables :



with F. Bréhard, A. Poteaux Université de Lille

Geometric