

# Computing points in connected components defined by a real inequation: algorithms, complexity and implementations.

**Edern Gillot**

JNCF 2026

Joint work with Jérémy Berthomieu and Mohab Safey El Din

6 March 2026



# Problem Description

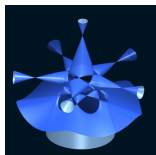
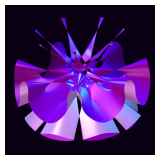
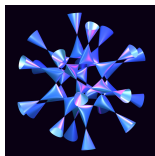
**Real polynomial** systems  
with **constraints**

$$\{\mathbf{x} \in \mathbb{R}^n : f_1(\mathbf{x}) = \dots = f_r(\mathbf{x}) = 0, \\ g_1(\mathbf{x}) > 0, \dots, g_s(\mathbf{x}) > 0\}, f_i, g_j \in \mathbb{R}[\mathbf{x}]$$

# Problem Description

**Real polynomial systems  
with constraints**

$$\{\mathbf{x} \in \mathbb{R}^n : f_1(\mathbf{x}) = \dots = f_r(\mathbf{x}) = 0, \\ g_1(\mathbf{x}) > 0, \dots, g_s(\mathbf{x}) > 0\}, f_i, g_j \in \mathbb{R}[\mathbf{x}]$$



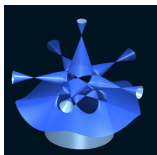
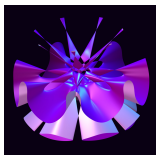
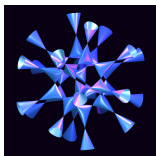
Source: **[IMAGINARY-Labs]**

# Problem Description

**Real polynomial systems  
with constraints**

$$\{\mathbf{x} \in \mathbb{R}^n : f_1(\mathbf{x}) = \dots = f_r(\mathbf{x}) = 0, \\ g_1(\mathbf{x}) > 0, \dots, g_s(\mathbf{x}) > 0\}, f_i, g_j \in \mathbb{R}[\mathbf{x}]$$

**Existence of solutions?**



Source: [IMAGINARY-Labs]

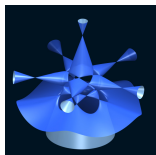
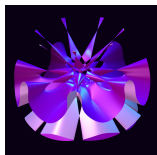
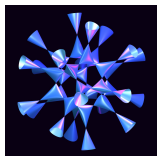
# Problem Description

**Real polynomial systems  
with constraints**

$$\{\mathbf{x} \in \mathbb{R}^n : f_1(\mathbf{x}) = \dots = f_r(\mathbf{x}) = 0, \\ g_1(\mathbf{x}) > 0, \dots, g_s(\mathbf{x}) > 0\}, f_i, g_j \in \mathbb{R}[\mathbf{x}]$$

**Existence of solutions?**

**NP Hard!** [Garey–Johnson 1979]



Source: [IMAGINARY-Labs]

# Problem Description

**Real polynomial systems  
with constraints**

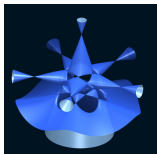
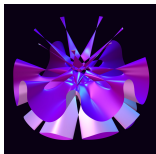
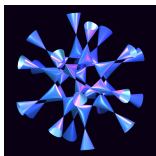
$$\{\mathbf{x} \in \mathbb{R}^n : f_1(\mathbf{x}) = \dots = f_r(\mathbf{x}) = 0, \\ g_1(\mathbf{x}) > 0, \dots, g_s(\mathbf{x}) > 0\}, f_i, g_j \in \mathbb{R}[\mathbf{x}]$$

**Existence of solutions?**

**NP Hard!** [Garey–Johnson 1979]

**Finite # of connected components**

[Tarski 1929, Whitney 1957, Łojasiewicz 1964]



Source: [IMAGINARY-Labs]

# Problem Description

**Real polynomial systems  
with constraints**

$$\{\mathbf{x} \in \mathbb{R}^n : f_1(\mathbf{x}) = \dots = f_r(\mathbf{x}) = 0, \\ g_1(\mathbf{x}) > 0, \dots, g_s(\mathbf{x}) > 0\}, f_i, g_j \in \mathbb{R}[\mathbf{x}]$$

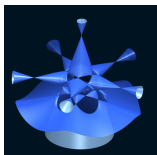
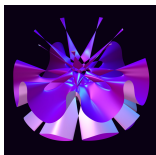
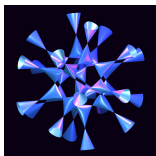
**Existence of solutions?**

**NP Hard!** [Garey–Johnson 1979]

**Finite # of connected components**

[Tarski 1929, Whitney 1957, Łojasiewicz 1964]

$s^n O(d)^n$  [Oleinik–Petrovski 1949,  
Milnor 1964, Thom 1965]  
 $n$  variables, max degree  $d$ ,  
 $s$  inequalities



Source: [IMAGINARY–Labs]

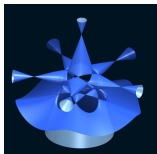
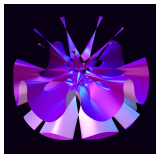
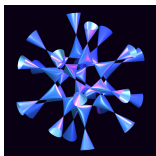
# Problem Description

**Real polynomial systems  
with constraints**

$$\{\mathbf{x} \in \mathbb{R}^n : f_1(\mathbf{x}) = \dots = f_r(\mathbf{x}) = 0, \\ g_1(\mathbf{x}) > 0, \dots, g_s(\mathbf{x}) > 0\}, f_i, g_j \in \mathbb{R}[\mathbf{x}]$$

**Given**  $f_i$ 's,  $g_j$ 's  
 $\in \mathbb{Q}[x_1, \dots, x_n]$ ,  $\rightarrow$  **at  
least one point per  
connected component**

$s^n O(d)^n$  [Oleinik–Petrovski 1949,  
Milnor 1964, Thom 1965]  
 $n$  variables, max degree  $d$ ,  
 $s$  inequalities



Source: [IMAGINARY-Labs]

# Problem Description

**Given**  $f_i$ 's,  $g_j$ 's  
 $\in \mathbb{Q}[x_1, \dots, x_n]$ ,  $\rightarrow$  **at**  
**least** one point per  
**connected component**

# Problem Description

Given  $f_i$ 's,  $g_j$ 's  
 $\in \mathbb{Q}[x_1, \dots, x_n]$ ,  $\rightarrow$  at  
least one point per  
connected component



## Applications:

**Robotics** [Chablat–Prébet–Safey El Din–Salunkhe–Wenger 2022]

[Capco–Safey El Din–Schicho 2023]

**Biology** [Feliu–Sadeghimanesh 2022]

[Yabo–Safey El Din–Caillaud–Gouzé 2023]

**Optimisation** [Greuet–Safey El Din 2014] [Ferguson 2022]

**Program Verification**

[Goharshady–Hitarth–Mohammadi–Motwani 2023]

[Bayarmagnai–Mohammadi–Prébet 2024] [Maaz–Strzeboński 2025]

**Combinatorics** [Kauers–Pillwein 2005] [Ibrahim–Salvy 2024]

# Problem Description

Given  $f_i$ 's,  $g_j$ 's  
 $\in \mathbb{Q}[x_1, \dots, x_n]$ ,  $\rightarrow$  at  
least one point per  
connected component



## Applications:

**Robotics** [Chablat–Prébet–Safey El Din–Salunkhe–Wenger 2022]

[Capco–Safey El Din–Schicho 2023]

**Biology** [Feliu–Sadeghimanesh 2022]

[Yabo–Safey El Din–Caillaud–Gouzé 2023]

**Optimisation** [Greuet–Safey El Din 2014] [Ferguson 2022]

**Program Verification**

[Goharshady–Hitarth–Mohammadi–Motwani 2023]

[Bayarmagnai–Mohammadi–Prébet 2024] [Maaz–Strzeboński 2025]

**Combinatorics** [Kauers–Pillwein 2005] [Ibrahim–Salvy 2024]

Make use of  
**inherent structure**

# Problem Description

Given  $f_i$ 's,  $g_j$ 's  
 $\in \mathbb{Q}[x_1, \dots, x_n]$ ,  $\rightarrow$  at  
least one point per  
connected component



## Applications:

**Robotics** [Chablat–Prébet–Safeq El Din–Salunkhe–Wenger 2022]

[Capco–Safeq El Din–Schicho 2023]

**Biology** [Feliu–Sadeghimanesh 2022]

[Yabo–Safeq El Din–Caillau–Gouzé 2023]

**Optimisation** [Greuet–Safeq El Din 2014] [Ferguson 2022]

**Program Verification**

[Goharshady–Hitarth–Mohammadi–Motwani 2023]

[Bayarmagnai–Mohammadi–Prébet 2024] [Maaz–Strzeboński 2025]

**Combinatorics** [Kauers–Pillwein 2005] [Ibrahim–Salvy 2024]

Make use of  
**inherent structure**

**Single** inequation:

$$S = \{x \in \mathbb{R}^n : f(x) \neq 0\}$$

# State-of-the-Art

$n$  variables, degree  $d$ , bitsize  $\tau$ , Thom–Milnor bound  $O(d)^n$

**First** algorithm: [Tarski 1951] not elementary recursive

# State-of-the-Art

$n$  variables, degree  $d$ , bitsize  $\tau$ , Thom–Milnor bound  $O(d)^n$

**First** algorithm: [Tarski 1951] not elementary recursive

**CAD** [Collins 1975]

# State-of-the-Art

$n$  variables, degree  $d$ , bitsize  $\tau$ , Thom–Milnor bound  $O(d)^n$

**First** algorithm: [Tarski 1951] not elementary recursive

In practice  
 $n \leq 4$



**CAD** [Collins 1975]

**Bit** complexity:  $\tau d^{2^{O(n)}}$

# State-of-the-Art

$n$  variables, degree  $d$ , bitsize  $\tau$ , Thom–Milnor bound  $O(d)^n$

First algorithm: [Tarski 1951] not elementary recursive

In practice  
 $n \leq 4$

CAD [Collins 1975]

Bit complexity:  $\tau d^{2^{O(n)}}$

Deterministic  
algorithms

[Grigoriev–Vorobjov 1988] [Renegar 1992]  
[Canny–Grigoriev–Vorobjov 1992]  
[Heintz–Roy–Solernó 1994]  
[Basu–Pollack–Roy 1996/2006]...

# State-of-the-Art

$n$  variables, degree  $d$ , bitsize  $\tau$ , Thom–Milnor bound  $O(d)^n$

First algorithm: [Tarski 1951] not elementary recursive

In practice  
 $n \leq 4$

**CAD** [Collins 1975]

Bit complexity:  $\tau d^{2^{O(n)}}$

**Deterministic**  
algorithms

[Grigoriev–Vorobjov 1988] [Renegar 1992]  
[Canny–Grigoriev–Vorobjov 1992]  
[Heintz–Roy–Solernó 1994]  
[Basu–Pollack–Roy 1996/2006]...

Bit complexity:  $\tau d^{O(n)}$   
**Constant** too large

**Critical** points & **Polar** varieties

# State-of-the-Art

$n$  variables, degree  $d$ , bitsize  $\tau$ , Thom–Milnor bound  $O(d)^n$

First algorithm: [Tarski 1951] not elementary recursive

In practice  
 $n \leq 4$

CAD [Collins 1975]

Bit complexity:  $\tau d^{2^{O(n)}}$

Deterministic  
algorithms

[Grigoriev–Vorobjov 1988] [Renegar 1992]  
[Canny–Grigoriev–Vorobjov 1992]  
[Heintz–Roy–Solernó 1994]  
[Basu–Pollack–Roy 1996/2006]...

Bit complexity:  $\tau d^{O(n)}$   
Constant too large

Critical points & Polar varieties

Probabilistic  
algorithms

[Bank–Giusti–Heintz–Mandel–Mbakop 1997] [Safey El Din–Schost 2003]  
[Bank–Giusti–Heintz–Pardo 2004/2009]  
[Bank–Giusti–Heintz–Safey El Din–Schost 2010]  
[Bank–Gisuti–Heintz–Lecref–Matera–Solerno 2015] [Le–Safey El Din 2021]  
[Elliott–Giesbrecht–Schost 2020/2023] [Harris–Hauenstein–Szanto 2023]...

# State-of-the-Art

$n$  variables, degree  $d$ , bitsize  $\tau$ , Thom–Milnor bound  $O(d)^n$

**Current arithmetic** complexity:

$$\tilde{O}\left(\mathcal{P}(n)8^n d^2 (d-1)^{2n}\right)$$

[Le–Safey El Din 2022]

**Current bit** complexity:

$$\tilde{O}\left(\tau \mathcal{P}(n) d^{3n+4}\right)$$

[Safey El Din–Schost 2003]

[Elliott–Giesbrecht–Schost 2023]

Do **not** involve degree structure of  $f$ , **general** bounds

# State-of-the-Art

$n$  variables, degree  $d$ , bitsize  $\tau$ , Thom–Milnor bound  $O(d)^n$

Current arithmetic complexity:

$$\tilde{O}\left(\mathcal{P}(n)8^n d^2 (d-1)^{2n}\right)$$

[Le–Safey El Din 2022]

Current bit complexity:

$$\tilde{O}\left(\tau \mathcal{P}(n) d^{3n+4}\right)$$

[Safey El Din–Schost 2003]  
[Elliott–Giesbrecht–Schost 2023]

Do **not** involve degree structure of  $f$ , **general** bounds

Taking **structure** into account for **bit** complexity

[Labahn–Safey El Din–Schost–Vu 2021] [Basu–Perrucci 2023]

[Faugère–Labahn–Safey El Din–Schost–Vu 2023]

# State-of-the-Art

$n$  variables, degree  $d$ , bitsize  $\tau$ , Thom–Milnor bound  $O(d)^n$

Current arithmetic complexity:

$$\tilde{O}\left(\mathcal{P}(n)8^n d^2 (d-1)^{2n}\right)$$

[Le–Safey El Din 2022]

Current bit complexity:

$$\tilde{O}\left(\tau \mathcal{P}(n) d^{3n+4}\right)$$

[Safey El Din–Schost 2003]  
[Elliott–Giesbrecht–Schost 2023]

Do **not** involve degree structure of  $f$ , **general** bounds

**Regularity** assumptions on  $f$

Taking **structure** into account for **bit** complexity

**Semi-algebraic** to **algebraic**

$$f \neq 0 \rightarrow \lambda f - 1 = 0$$

[Labahn–Safey El Din–Schost–Vu 2021] [Basu–Perrucci 2023]

[Faugère–Labahn–Safey El Din–Schost–Vu 2023]

**Extra** variable & **breaks** structure

# Contributions

**Input:**  $f \in \mathbb{Q}[x_1, \dots, x_n]$ ,  $0 < \epsilon < 1$

**Output:** **parametrisation** of  
finite set of points with  
probability **at least**  $1 - \epsilon$

# Contributions

**Input:**  $f \in \mathbb{Q}[x_1, \dots, x_n]$ ,  $0 < \epsilon < 1$

**New Bit Complexity:**

$$\tilde{O}\left(\tau \mathcal{P}(n, \log(1/\epsilon)) \mathfrak{D}^3\right)$$

Valid for  $V(f)$  **smooth** and **singular**

$\mathfrak{D}$  encapsulates **degree structure**

$\mathfrak{D}$  is a Bézout bound  
 $\implies$  analysis remains valid if  
better bound found!

**Output:** **parametrisation** of  
finite set of points with  
probability **at least**  $1 - \epsilon$

$$\mathfrak{D} := \deg(f) \prod_{i=2}^n \deg\left(\frac{\partial f}{\partial X_i}\right)$$
$$\mathfrak{D} \leq d(d-1)^{n-1}$$

**Multi-homogeneous** Bézout  
bound, **BKK** bound for **sparse**  
systems...

[Bernstein 1975] [Kouchnirenko 1976] [Khovanskii 1978]  
[van der Waerden 1927]...

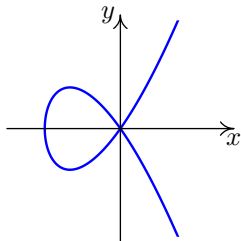
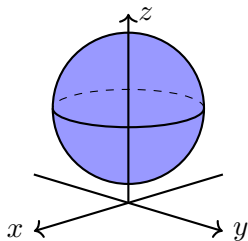
# Definitions

**Smoothness**

# Definitions

**Smoothness**

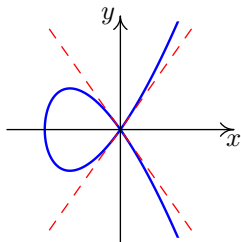
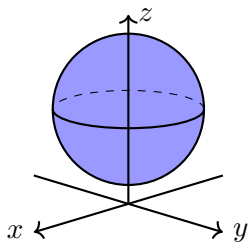
**Tangent space** has expected **dimension**



# Definitions

**Smoothness**

**Tangent space** has expected **dimension**

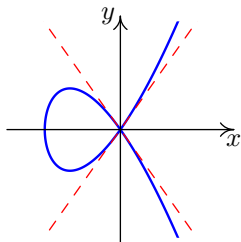
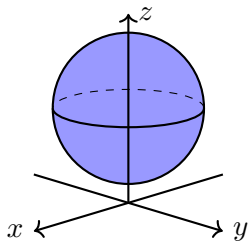


# Definitions

Smoothness

Tangent space has  
expected **dimension**

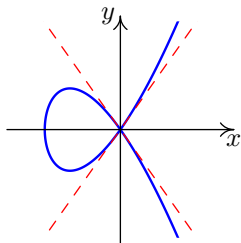
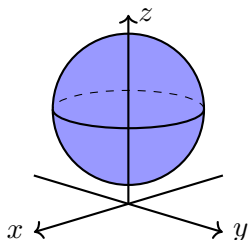
Critical points



# Definitions

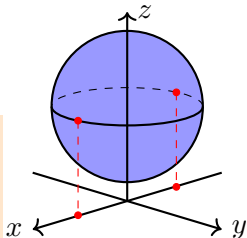
## Smoothness

Tangent space has expected **dimension**



## Critical points

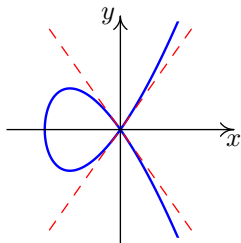
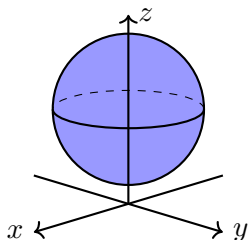
Smooth points for which tangent space projection has **lower dimension**



# Definitions

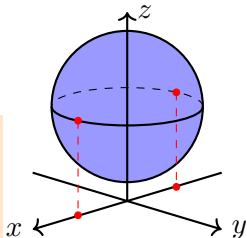
## Smoothness

Tangent space has expected **dimension**



## Critical points

Smooth points for which tangent space projection has **lower dimension**

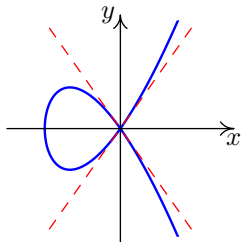
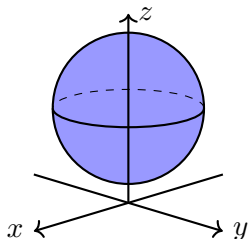


**Critical** points are defined by **partial derivatives**; **Singular** points **always** satisfy critical equations

# Definitions

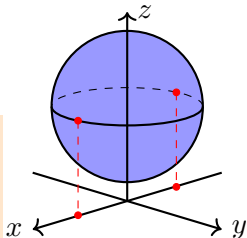
## Smoothness

Tangent space has expected **dimension**



## Critical points

Smooth points for which tangent space projection has **lower dimension**



**Critical** points are defined by **partial derivatives**; **Singular** points **always** satisfy critical equations

**Generic:** true in a non-empty Zariski open set

# Smooth Case — Change of Variables

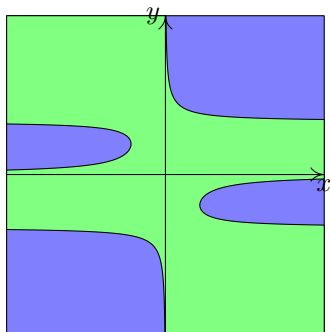
**Random** change of variables

**Properness** of projection on  $x$ -axis

# Smooth Case — Change of Variables

**Random** change of variables

**Properness** of projection on  $x$ -axis

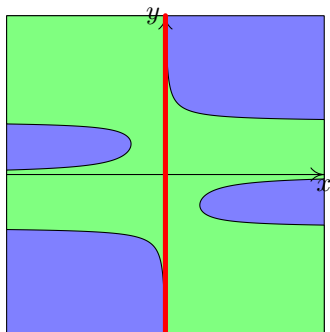


$$f = 4x(y^3 - y) - 1 \neq 0$$

# Smooth Case — Change of Variables

**Random** change of variables

**Properness** of projection on  $x$ -axis

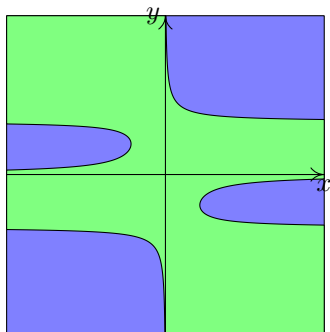


$$f = 4x(y^3 - y) - 1 \neq 0$$

# Smooth Case — Change of Variables

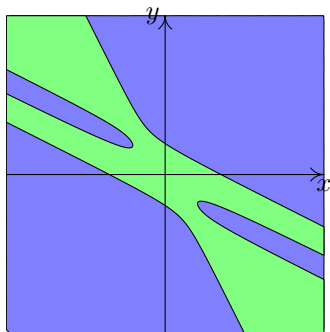
**Random** change of variables

**Properness** of projection on  $x$ -axis



$$f = 4x(y^3 - y) - 1 \neq 0$$

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

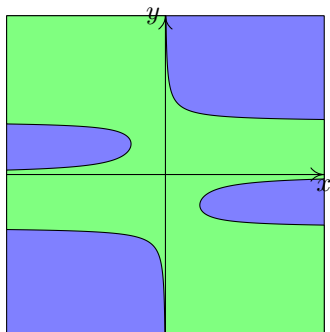


$$f^{\mathbf{A}} := f(\mathbf{A}(x, y)^T) \neq 0$$

# Smooth Case — Change of Variables

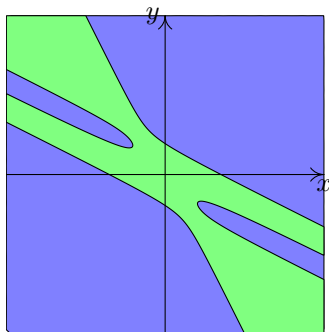
**Random** change of variables

**Properness** of projection on  $x$ -axis



$$f = 4x(y^3 - y) - 1 \neq 0$$

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$



$$f^{\mathbf{A}} := f(\mathbf{A}(x, y)^T) \neq 0$$

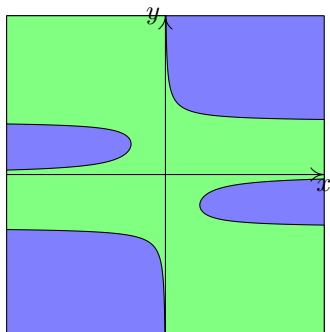
**Generic**  $\mathbf{A} \implies$  projection of **any** component of  $V(f^{\mathbf{A}})$  on  $x$ -axis is **closed** and has **finite** fibres

[Safey El Din-Schost 2003]

# Smooth Case — Change of Variables

**Random** change of variables

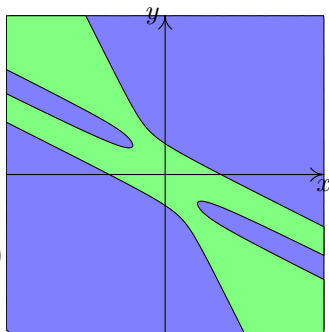
**Properness** of projection on  $x$ -axis



$$f = 4x(y^3 - y) - 1 \neq 0$$

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\log(\mathbf{A}_{ij}) \in \tilde{O}(n + \log(d)) + \log(1/\epsilon)$$



$$f^{\mathbf{A}} := f(\mathbf{A}(x, y)^T) \neq 0$$

**Generic  $\mathbf{A} \implies$  projection of any component of  $V(f^{\mathbf{A}})$  on  $x$ -axis is closed and has finite fibres**

[Safey El Din–Schost 2003]

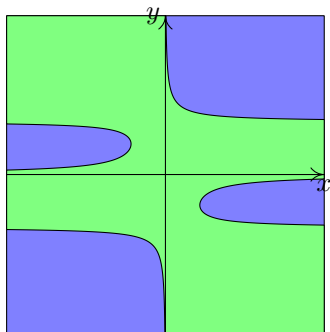
**Probability of choosing a good  $\mathbf{A}$ :  $\geq 1 - \epsilon/3$**

[Elliott–Giesbrecht–Schost 2020/2023]

# Smooth Case — Change of Variables

**Random** change of variables

**Properness** of projection on  $x$ -axis

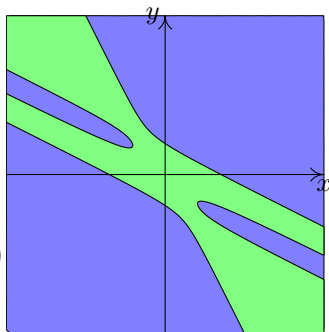


$$f = 4x(y^3 - y) - 1 \neq 0$$

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\log(\mathbf{A}_{ij}) \in \tilde{O}(n + \log(d) + \log(1/\epsilon))$$

**No  $\tau$**



$$f^A := f(\mathbf{A}(x, y)^T) \neq 0$$

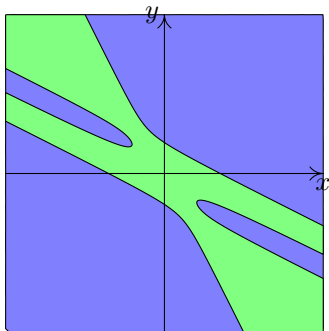
**Generic  $\mathbf{A} \implies$  projection of any component of  $V(f^A)$  on  $x$ -axis is closed and has finite fibres**

[Safey El Din–Schost 2003]

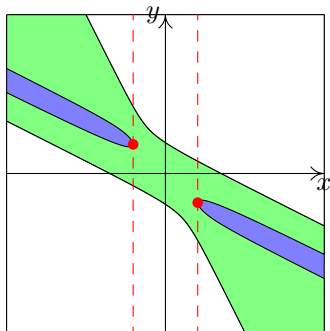
**Probability of choosing a good  $\mathbf{A}$ :  $\geq 1 - \epsilon/3$**

[Elliott–Giesbrecht–Schost 2020/2023]

## Smooth Case — Critical Points

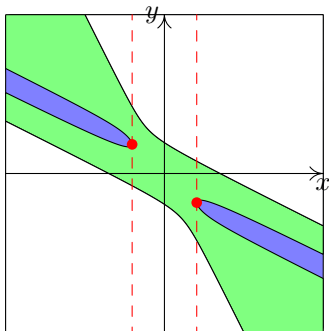


## Smooth Case — Critical Points



$$f^A = \frac{\partial f^A}{\partial x_2} = \dots = \frac{\partial f^A}{\partial x_n} = 0$$

## Smooth Case — Critical Points



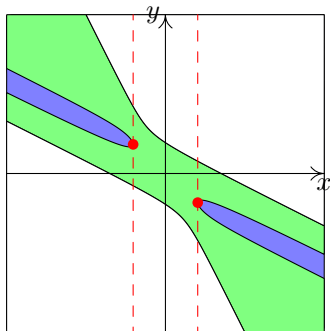
$$f^A = \frac{\partial f^A}{\partial x_2} = \dots = \frac{\partial f^A}{\partial x_n} = 0$$

**Bit cost** for computing  
**parametrisation** of solutions:

**quadratic** in  $\deg(f^A) \prod_{i=2}^n \deg\left(\frac{\partial f^A}{\partial X_i}\right)$

[Safey El Din–Schost 2018]

# Smooth Case — Critical Points



$$f^A = \frac{\partial f^A}{\partial x_2} = \dots = \frac{\partial f^A}{\partial x_n} = 0$$

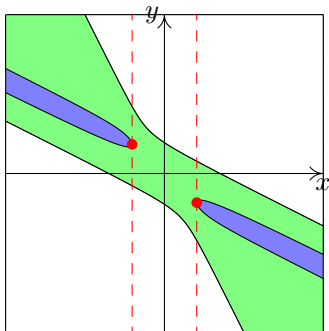
**Bit cost** for computing  
**parametrisation** of solutions:

**quadratic** in  $\deg(f^A) \prod_{i=2}^n \deg\left(\frac{\partial f^A}{\partial X_i}\right)$

[Safey El Din–Schost 2018]

$= d(d-1)^{n-1}$ : **A breaks structure!**

# Smooth Case — Critical Points



$$f^A = \frac{\partial f^A}{\partial x_2} = \dots = \frac{\partial f^A}{\partial x_n} = 0$$

**Bit cost** for computing  
**parametrisation** of solutions:

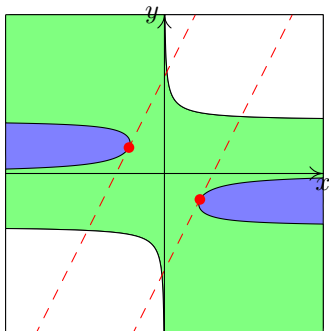
$$\text{quadratic in } \deg(f^A) \prod_{i=2}^n \deg\left(\frac{\partial f^A}{\partial X_i}\right)$$

[Safey El Din–Schost 2018]

$$= d(d-1)^{n-1}: \mathbf{A \text{ breaks structure!}}$$

$$\exists g_2, \dots, g_n \in \mathbb{Q}[x_1, \dots, x_n] : \deg(g_i) = \deg\left(\frac{\partial f}{\partial X_i}\right) \text{ such that}$$
$$\xi \text{ critical} \iff \mathbf{A\xi} \in V(f, g_2, \dots, g_n)$$

# Smooth Case — Critical Points



$$f^A = \frac{\partial f^A}{\partial x_2} = \dots = \frac{\partial f^A}{\partial x_n} = 0$$

**Bit cost** for computing  
**parametrisation** of solutions:

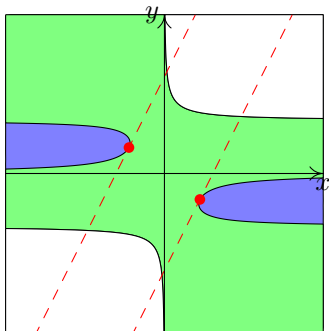
**quadratic** in  $\deg(f^A) \prod_{i=2}^n \deg\left(\frac{\partial f^A}{\partial X_i}\right)$

[Safey El Din–Schost 2018]

$= d(d-1)^{n-1}$ : **A breaks structure!**

$\exists g_2, \dots, g_n \in \mathbb{Q}[x_1, \dots, x_n] : \deg(g_i) = \deg\left(\frac{\partial f}{\partial X_i}\right)$  such that  
 $\xi$  critical  $\iff \mathbf{A}\xi \in V(f, g_2, \dots, g_n)$

## Smooth Case — Critical Points



$$f^A = \frac{\partial f^A}{\partial x_2} = \dots = \frac{\partial f^A}{\partial x_n} = 0$$

**Bit cost** for computing  
**parametrisation** of solutions:  
**quadratic** in  $\deg(f^A) \prod_{i=2}^n \deg\left(\frac{\partial f^A}{\partial X_i}\right)$

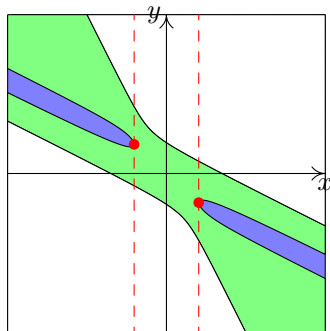
[Safey El Din–Schost 2018]

$= d(d-1)^{n-1}$ : **A breaks structure!**

$\exists g_2, \dots, g_n \in \mathbb{Q}[x_1, \dots, x_n] : \deg(g_i) = \deg\left(\frac{\partial f}{\partial X_i}\right)$  such that  
 $\xi$  critical  $\iff \mathbf{A}\xi \in V(f, g_2, \dots, g_n)$

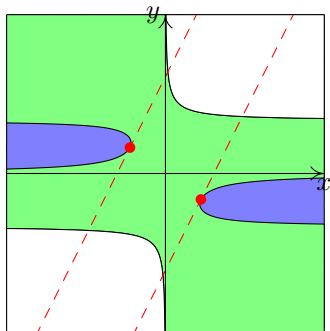
**Bit cost** for computing these points:  $\tilde{O}((\tau + \log(1/\epsilon))\mathcal{P}(n)\mathcal{D}^2)$   
**Probability** of success  $\geq 1 - \epsilon/3$  after  $O(\log(1/\epsilon))$  attempts

## Smooth Case — Transverse Line



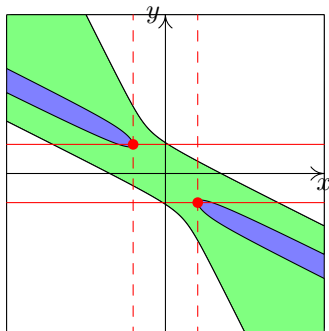
$f^A \neq 0$

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$



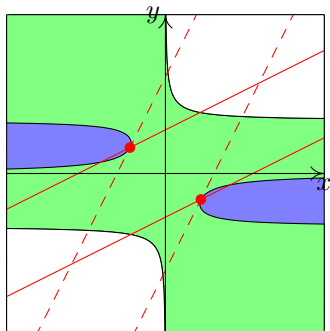
$f \neq 0$

## Smooth Case — Transverse Line



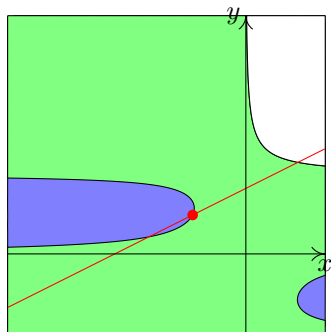
$f^A \neq 0$

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$



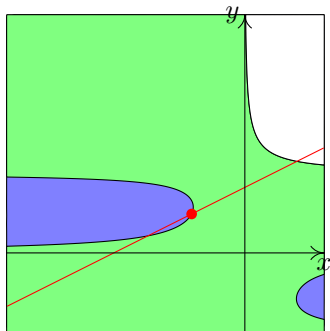
$f \neq 0$

## Smooth Case — Transverse Line



$$f \neq 0$$

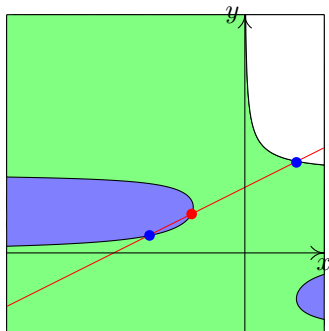
## Smooth Case — Transverse Line



$$f \neq 0$$

Minimal distance between  
**computed solution point** and other  
intersection points

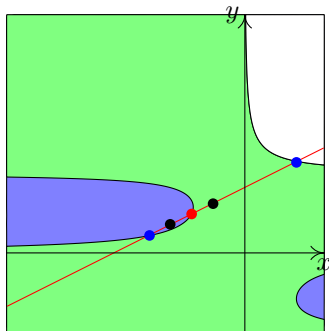
## Smooth Case — Transverse Line



$$f \neq 0$$

Minimal distance between  
**computed solution point** and other  
intersection points

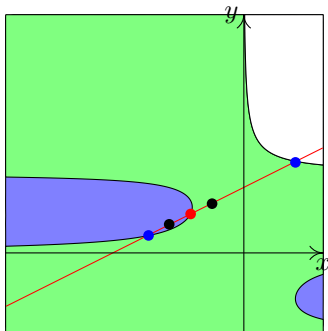
## Smooth Case — Transverse Line



$$f \neq 0$$

Minimal distance between  
**computed solution point** and other  
intersection points

# Smooth Case — Transverse Line



$f \neq 0$

Minimal distance between  
**computed solution point** and other  
intersection points

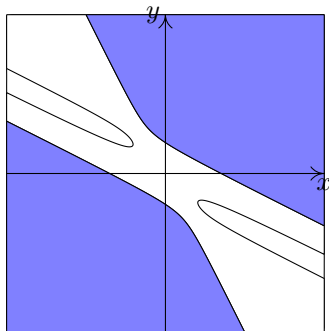
Number of real critical points  
 $\mathfrak{R} \leq \mathfrak{D}$

Bit cost of computing **minimal distance** and  
**parametrisations** of the **black points**:

$$\tilde{O}((\tau + \log(1/\epsilon))\mathcal{P}(n)\mathfrak{R}\mathfrak{D}^2)$$

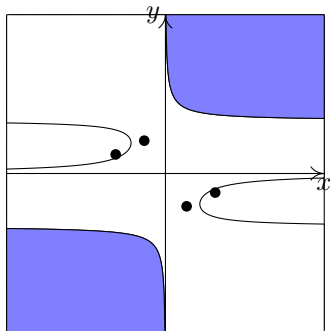
[Strzebonski–Tsigaridas 2019]

## Smooth Case — Specification



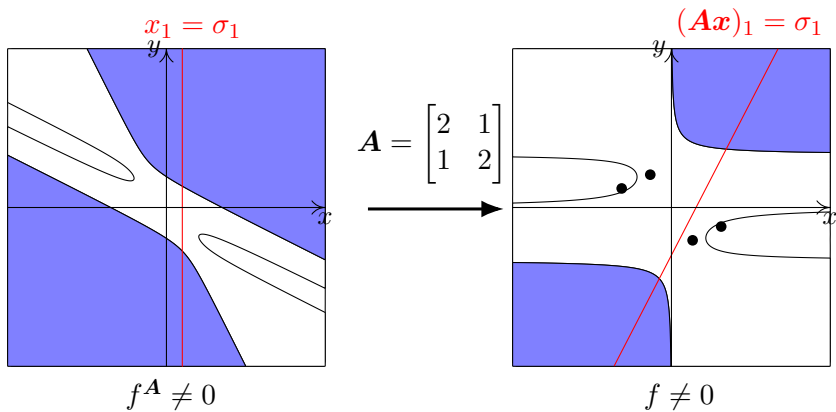
$f^A \neq 0$

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

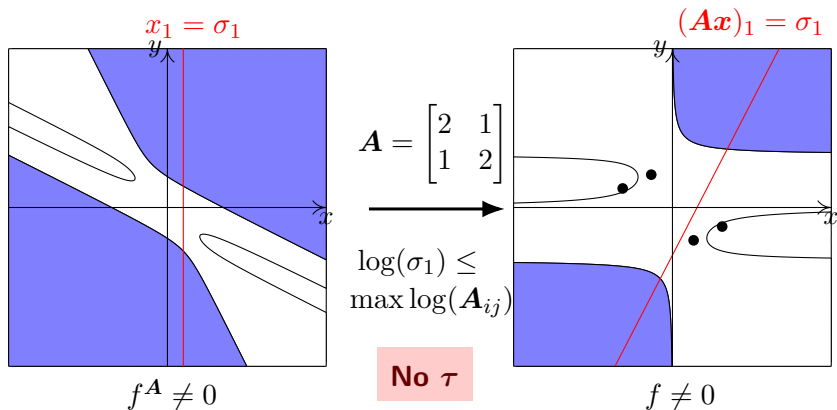


$f \neq 0$

# Smooth Case — Specification



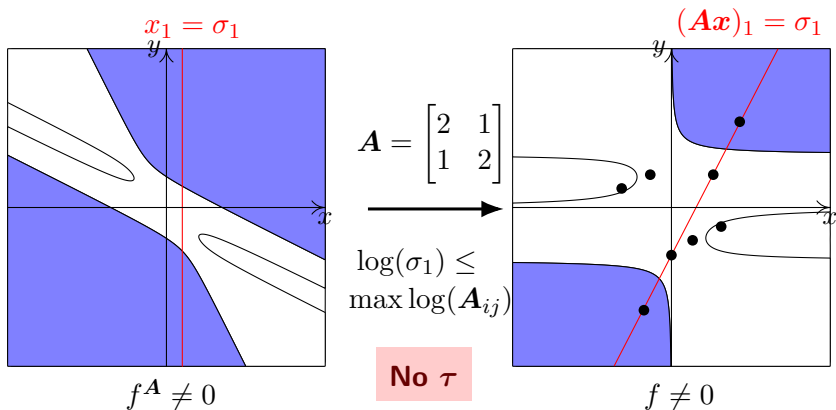
# Smooth Case — Specification



**Probability** of choosing  $\sigma_1$  such that  $V((Ax)_1 - \sigma_1, f)$  is **smooth** is  $\geq 1 - \epsilon/3$

[Elliott–Giesbrecht–Schost 2020/2023]

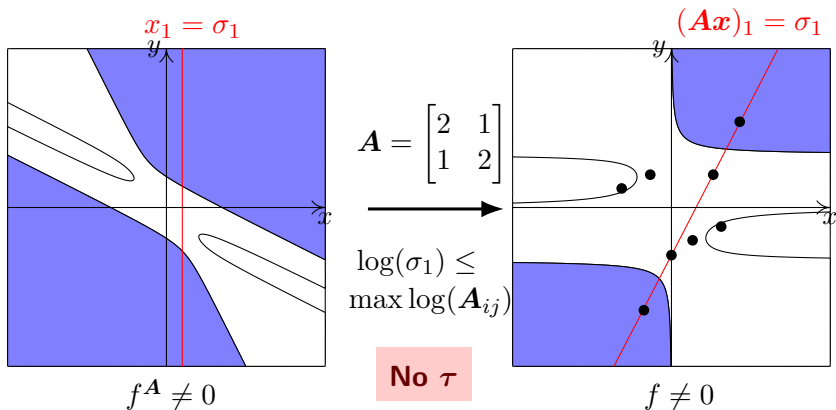
# Smooth Case — Specification



**Probability** of choosing  $\sigma_1$  such that  $V((Ax)_1 - \sigma_1, f)$  is **smooth** is  $\geq 1 - \epsilon/3$

[Elliott-Giesbrecht-Schost 2020/2023]

# Smooth Case — Specification

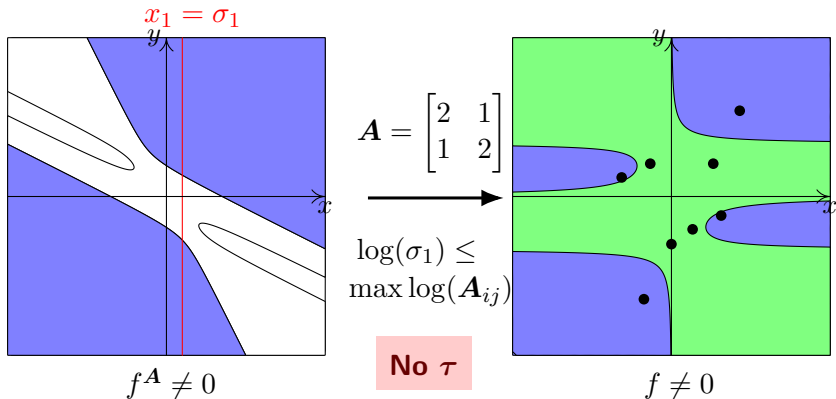


**Probability** of choosing  $\sigma_1$  such that  
 $V((Ax)_1 - \sigma_1, f)$  is **smooth** is  $\geq 1 - \epsilon/3$

[Elliott–Giesbrecht–Schost 2020/2023]

**Total bit complexity**  $\tilde{O}(\tau \mathcal{P}(n, \log(1/\epsilon)) \mathfrak{R} \mathfrak{D}^2)$ ; **Probability**  $\geq 1 - \epsilon$

# Smooth Case — Specification



**Probability** of choosing  $\sigma_1$  such that  
 $V((\mathbf{A}\mathbf{x})_1 - \sigma_1, f)$  is **smooth** is  $\geq 1 - \epsilon/3$

[Elliott-Giesbrecht-Schost 2020/2023]

**Total bit complexity**  $\tilde{O}(\tau \mathcal{P}(n, \log(1/\epsilon)) \mathfrak{R} \mathfrak{D}^2)$ ; **Probability**  $\geq 1 - \epsilon$

## Singular Case – Curve

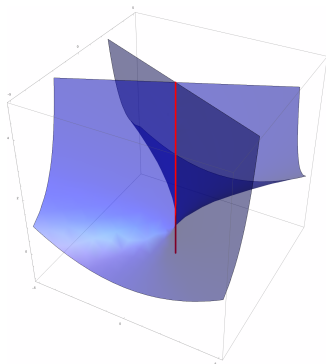
**No** guarantee of  
**finiteness** of solutions

$$f^A = \frac{\partial f^A}{\partial x_2} = \dots = \frac{\partial f^A}{\partial x_n} = 0$$

## Singular Case – Curve

**No** guarantee of  
**finiteness** of solutions

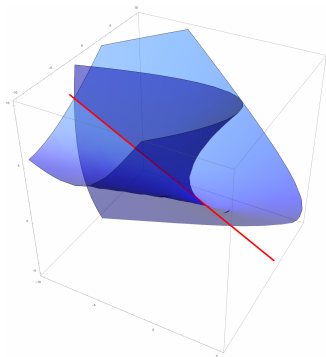
$$f^A = \frac{\partial f^A}{\partial x_2} = \dots = \frac{\partial f^A}{\partial x_n} = 0$$



## Singular Case – Curve

**No** guarantee of  
**finiteness** of solutions

$$f^A = \frac{\partial f^A}{\partial x_2} = \dots = \frac{\partial f^A}{\partial x_n} = 0$$

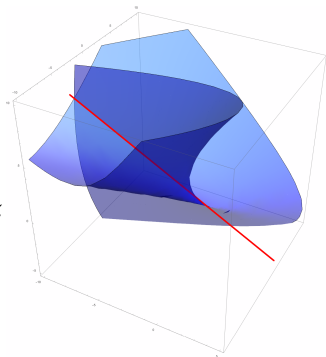


## Singular Case – Curve

No guarantee of  
**finiteness** of solutions

$$f^A = \frac{\partial f^A}{\partial x_2} = \dots = \frac{\partial f^A}{\partial x_n} = 0$$

$$\mathcal{C}^A := V \left( \frac{\partial f^A}{\partial x_2}, \dots, \frac{\partial f^A}{\partial x_n} \right) \setminus V \left( \frac{\partial f^A}{\partial x_1} \right) \quad \mathcal{Z}$$

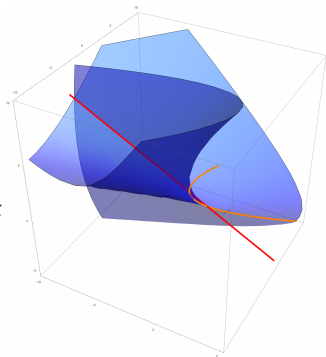


## Singular Case – Curve

No guarantee of  
**finiteness** of solutions

$$f^A = \frac{\partial f^A}{\partial x_2} = \dots = \frac{\partial f^A}{\partial x_n} = 0$$

$$\mathcal{C}^A := V \left( \frac{\partial f^A}{\partial x_2}, \dots, \frac{\partial f^A}{\partial x_n} \right) \setminus V \left( \frac{\partial f^A}{\partial x_1} \right)^z$$

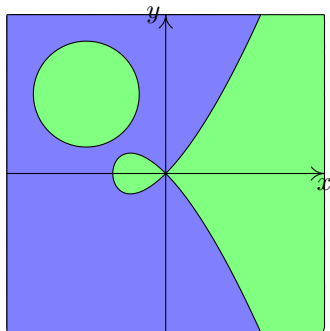


## Singular Case – Curve

No guarantee of  
**finiteness** of solutions

$$f^A = \frac{\partial f^A}{\partial x_2} = \dots = \frac{\partial f^A}{\partial x_n} = 0$$

$$C^A := V \left( \frac{\partial f^A}{\partial x_2}, \dots, \frac{\partial f^A}{\partial x_n} \right) \setminus V \left( \frac{\partial f^A}{\partial x_1} \right)$$



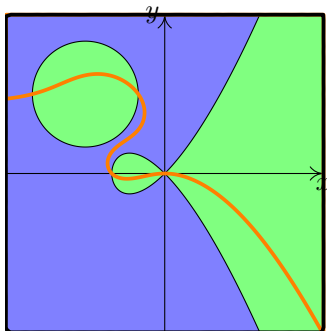
Singular  $f^A \neq 0$

## Singular Case – Curve

No guarantee of  
**finiteness** of solutions

$$f^A = \frac{\partial f^A}{\partial x_2} = \dots = \frac{\partial f^A}{\partial x_n} = 0$$

$$C^A := V \left( \frac{\partial f^A}{\partial x_2}, \dots, \frac{\partial f^A}{\partial x_n} \right) \setminus V \left( \frac{\partial f^A}{\partial x_1} \right)^z$$



Singular  $f^A \neq 0$

# Singular Case – Curve

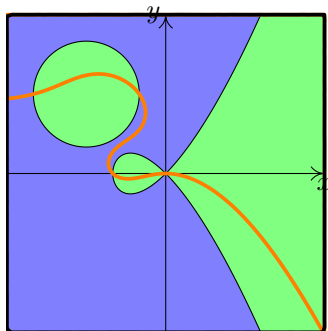
No guarantee of  
**finiteness** of solutions

$$f^A = \frac{\partial f^A}{\partial x_2} = \dots = \frac{\partial f^A}{\partial x_n} = 0$$

$$\mathcal{C}^A := V \left( \frac{\partial f^A}{\partial x_2}, \dots, \frac{\partial f^A}{\partial x_n} \right) \setminus V \left( \frac{\partial f^A}{\partial x_1} \right)^z$$

**Generic  $A \implies \mathcal{C}^A$  is a curve**

[Safey El Din 2005] [Lairez–Safey El Din 2021]



Singular  $f^A \neq 0$

# Singular Case – Curve

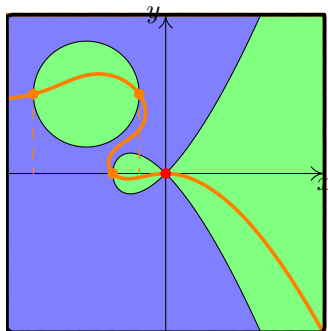
No guarantee of  
**finiteness** of solutions

$$f^A = \frac{\partial f^A}{\partial x_2} = \dots = \frac{\partial f^A}{\partial x_n} = 0$$

$$\mathcal{C}^A := V \left( \frac{\partial f^A}{\partial x_2}, \dots, \frac{\partial f^A}{\partial x_n} \right) \setminus V \left( \frac{\partial f^A}{\partial x_1} \right)$$

**Generic  $A \implies \mathcal{C}^A$  is a curve**

[Safey El Din 2005] [Lairez–Safey El Din 2021]



Singular  $f^A \neq 0$

# Singular Case – Curve

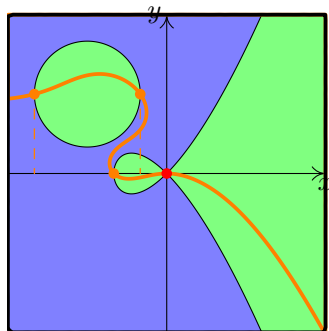
No guarantee of  
**finiteness** of solutions

$$f^A = \frac{\partial f^A}{\partial x_2} = \dots = \frac{\partial f^A}{\partial x_n} = 0$$

$$\mathcal{C}^A := V \left( \frac{\partial f^A}{\partial x_2}, \dots, \frac{\partial f^A}{\partial x_n} \right) \setminus V \left( \frac{\partial f^A}{\partial x_1} \right)$$

**Generic  $A \implies \mathcal{C}^A$  is a curve**

[Safey El Din 2005] [Lairez–Safey El Din 2021]



Singular  $f^A \neq 0$

**Probability**  $\geq 1 - \epsilon/3$ ; **equivalent** polynomials  $g_1, \dots, g_n$  defining  $\mathcal{C}$

# Singular Case – Curve

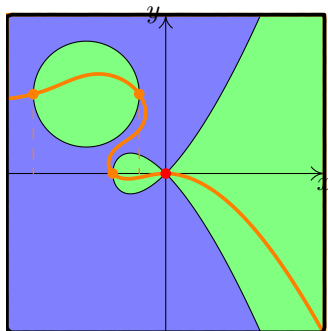
No guarantee of  
**finiteness** of solutions

$$f^A = \frac{\partial f^A}{\partial x_2} = \dots = \frac{\partial f^A}{\partial x_n} = 0$$

$$\mathcal{C}^A := V \left( \frac{\partial f^A}{\partial x_2}, \dots, \frac{\partial f^A}{\partial x_n} \right) \setminus V \left( \frac{\partial f^A}{\partial x_1} \right)$$

**Generic  $A \implies \mathcal{C}^A$  is a curve**

[Safey El Din 2005] [Lairez–Safey El Din 2021]



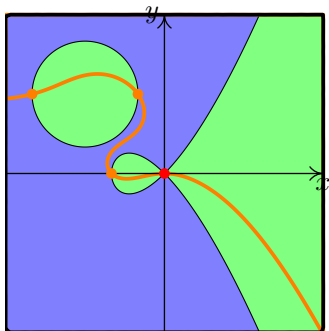
Singular  $f^A \neq 0$

**Probability**  $\geq 1 - \epsilon/3$ ; **equivalent** polynomials  $g_1, \dots, g_n$  defining  $\mathcal{C}$

**Bit cost** of computing a **parametrisation** of  $\mathcal{C}$ :

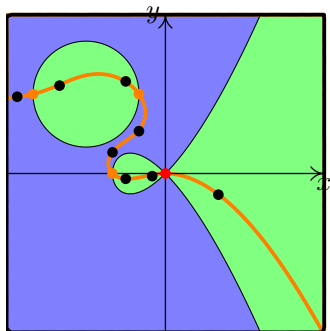
$$\tilde{O}(\tau \mathcal{P}(n, \log(1/\epsilon)) \mathfrak{D}^2)$$

## Singular Case – Optimisation



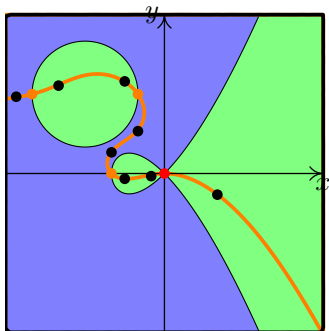
Compute  $e \in \mathbb{Q}$  small enough, and  
then compute  $\mathcal{C} \cap V(f \pm e)$

## Singular Case – Optimisation



Compute  $e \in \mathbb{Q}$  small enough, and  
then compute  $\mathcal{C} \cap V(f \pm e)$

# Singular Case – Optimisation

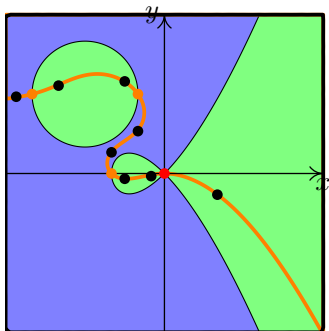


Compute  $e \in \mathbb{Q}$  small enough, and  
then compute  $\mathcal{C} \cap V(f \pm e)$

... smaller than the asymptotic  
values of  $f$  on  $\mathcal{C}$ . Bit cost:  
 $\sim \tilde{O}(\mathcal{D}^2)$ , Bit size:  $\sim \tilde{O}(\mathcal{D})$

[Krick–Pardo–Sombra 2001] [Safey El Din–Schost 2004]  
[Dahan–Kadri–Schost 2012]

# Singular Case – Optimisation



Compute  $e \in \mathbb{Q}$  small enough, and  
then compute  $\mathcal{C} \cap V(f \pm e)$

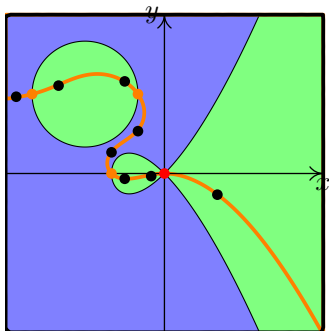
... smaller than the asymptotic  
values of  $f$  on  $\mathcal{C}$ . Bit cost:  
 $\sim \tilde{O}(\mathfrak{D}^2)$ , Bit size:  $\sim \tilde{O}(\mathfrak{D})$

[Krick–Pardo–Sombra 2001] [Safey El Din–Schost 2004]  
[Dahan–Kadri–Schost 2012]

... smaller than the critical and  
singular values of  $f$  on  $\mathcal{C}$ . Bit cost:  
 $\sim \tilde{O}(\mathfrak{D}^3)$ , Bit size:  $\sim \tilde{O}(\mathfrak{D})$

[Giusti–Lecerf–Salvy 2001] [Giménez–Matera 2019]  
[Giménez–Heintz–Matera–Pardo–Pérez–Privitelli 2026]

# Singular Case – Optimisation



Bit cost of computing the  
black points:

$$\tilde{O}(\tau \mathcal{P}(n, \log(1/\epsilon)) \mathfrak{D}^3)$$

[Safey El Din–  
Schost 2018]

Compute  $e \in \mathbb{Q}$  small enough, and  
then compute  $\mathcal{C} \cap V(f \pm e)$

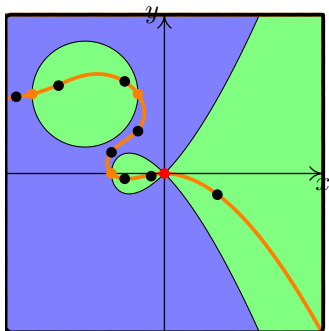
... smaller than the asymptotic  
values of  $f$  on  $\mathcal{C}$ . Bit cost:  
 $\sim \tilde{O}(\mathfrak{D}^2)$ , Bit size:  $\sim \tilde{O}(\mathfrak{D})$

[Krick–Pardo–Sombra 2001] [Safey El Din–Schost 2004]  
[Dahan–Kadri–Schost 2012]

... smaller than the critical and  
singular values of  $f$  on  $\mathcal{C}$ . Bit cost:  
 $\sim \tilde{O}(\mathfrak{D}^3)$ , Bit size:  $\sim \tilde{O}(\mathfrak{D})$

[Giusti–Lecerf–Salvy 2001] [Giménez–Matera 2019]  
[Giménez–Heintz–Matera–Pardo–Pérez–Privitelli 2026]

# Singular Case – Optimisation



Bit cost of computing the black points:

$$\tilde{O}(\tau \mathcal{P}(n, \log(1/\epsilon)) \mathfrak{D}^3)$$

[Safey El Din–Schost 2018]

Compute  $e \in \mathbb{Q}$  small enough, and then compute  $\mathcal{C} \cap V(f \pm e)$

... smaller than the asymptotic values of  $f$  on  $\mathcal{C}$ . Bit cost:  $\sim \tilde{O}(\mathfrak{D}^2)$ , Bit size:  $\sim \tilde{O}(\mathfrak{D})$

[Krick–Pardo–Sombra 2001] [Safey El Din–Schost 2004]  
[Dahan–Kadri–Schost 2012]

... smaller than the critical and singular values of  $f$  on  $\mathcal{C}$ . Bit cost:  $\sim \tilde{O}(\mathfrak{D}^3)$ , Bit size:  $\sim \tilde{O}(\mathfrak{D})$

[Giusti–Lecerf–Salvy 2001] [Giménez–Matera 2019]  
[Giménez–Heintz–Matera–Pardo–Pérez–Privitelli 2026]

Repeat with  $x_1 = \sigma_1, \dots$  Total bit complexity  $\tilde{O}(\tau \mathcal{P}(n, \log(1/\epsilon)) \mathfrak{D}^3)$ ; Probability of success  $\geq 1 - \epsilon$

# Experiments

**SageMath** implementation, using **msolve** for zero dimensional polynomial system solving [Berthomieu–Eder–Safey El Din 2021]

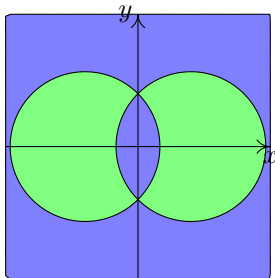
# Experiments

**SageMath** implementation, using **msolve** for zero dimensional polynomial system solving [Berthomieu–Eder–Safey El Din 2021]

**CAD:** SAMPLEPOINTS from Maple;  
[Safey El Din–Schost 2003]: state-of-the-art bit complexity;  
**Numerical:** HYPERSURFACE REGIONS from Julia

**Numerical** algorithm  
has no **certification**

$$f = ((x - 1)^2 + y^2 - 1 - 1/2^{16}) \\ \times ((x + 1)^2 + y^2 - 1 - 1/2^{16})$$



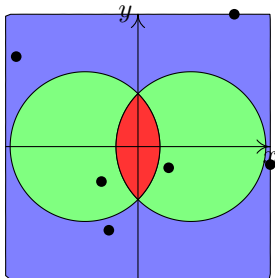
# Experiments

**SageMath** implementation, using **msolve** for zero dimensional polynomial system solving [Berthomieu–Eder–Safey El Din 2021]

**CAD:** SAMPLEPOINTS from Maple;  
[Safey El Din–Schost 2003]: state-of-the-art bit complexity;  
**Numerical:** HYPERSURFACE REGIONS from Julia

**Numerical** algorithm  
has no **certification**

$$f = ((x - 1)^2 + y^2 - 1 - 1/2^{16}) \\ \times ((x + 1)^2 + y^2 - 1 - 1/2^{16})$$



# Experiments

**SageMath** implementation, using **msolve** for zero dimensional polynomial system solving [Berthomieu–Eder–Safey El Din 2021]

**Single thread**;  $\infty$  indicates  $> 25$  days; **OOM** indicates  $> 1.5$ TB of memory requirement; **Err** indicates an error

$n$	CAD SAMPLEPOINTS.MM		[Safey El Din–Schost 2003]		New Algo		HYPERSURFACEREGIONS.JL	
	Time	# of points	Time	# of points	Time	# of points	Time	# of points
4	11 000	20607	14	23	3	13	84	19
5	$\infty$	—	500	61	10	65	110	33
6	$\infty$	—	37 000	97	120	73	230	53
7	$\infty$	—	$\infty$	—	8 800	177	1 100	117
8	$\infty$	—	$\infty$	—	1 800 000	183	7 400	113
9	$\infty$	—	$\infty$	—	$\infty$	—	38 000	269
10	$\infty$	—	$\infty$	—	$\infty$	—	170 000	349

Comparisons on **generic, dense** degree 4 examples of fixed bitsize 8, in **seconds**

# Experiments

**SageMath** implementation, using **msolve** for zero dimensional polynomial system solving [Berthomieu–Eder–Safei El Din 2021]

**Single thread**;  $\infty$  indicates  $> 25$  days; **OOM** indicates  $> 1.5$ TB of memory requirement; **Err** indicates an error

$n, k$	CAD SAMPLEPOINTS.MM		[Safei El Din–Schost 2003]		New Algo		HYPERSURFACEREGIONS.JL	
	Time	# of points	Time	# of points	Time	# of points	Time	# of points
8, 2	260	3105	19 000	49	24	117	220	41
8, 3	64 000	21343	$\infty$	—	11 000	237	10 000	105
8, 4	$\infty$	—	$\infty$	—	$\infty$	—	480 000	243
12, 2	20 000	25181	OOM	—	31	89	850	37
12, 3	$\infty$	—	OOM	—	20 000	241	19 000	95
20, 2	$\infty$	—	OOM	—	110	205	3 300	45
20, 3	$\infty$	—	OOM	—	56 000	345	140 000	Err
50, 1	$\infty$	—	OOM	—	210	261	5 900	9
50, 2	$\infty$	—	OOM	—	960	325	200 000	43
100, 0	1	4	22 000	247	1 200	257	15 000	13
150, 0	Err	—	260 000	387	7 300	397	100 000	17

Comparisons on degree 12 examples with  $n - k$  partial derivatives of degree 1, in **seconds**

# Experiments

**SageMath** implementation, using **msolve** for zero dimensional polynomial system solving [Berthomieu–Eder–Safei El Din 2021]

**Single thread**;  $\infty$  indicates  $> 25$  days; **OOM** indicates  $> 1.5$ TB of memory requirement; **Err** indicates an error

Polynomial	CAD SAMPLEPOINTS.MM		[Safei El Din–Schost 2003]		New Algo		HYPERSURFACEREGIONS.JL	
	Time	# of points	Time	# of points	Time	# of points	Time	# of points
Vor1	4	48	230	47	17	53	200	Err
Vor2	$\infty$	—	830 000	87	17 000	149	220 000	23
P3PGen	$\infty$	—	$\infty$	—	80 000	709	8 100	234
K1	OOM	—	11 000	643	280	381	8 400	Err
K2	1	112	9 600	693	290	345	3 400	Err
K3	$\infty$	—	560	355	56	201	1 600	Err
K4	2	224	11 000	717	340	373	6 700	Err
Kalto1–4	$\infty$	—	$\infty$	—	$\infty$	—	$\infty$	—

Computations on **deformed** ( $1/2^{32}$  subtracted) **singular** examples arising from applications, in **seconds**

Vor from **computational geometry**; P3PGen from **computer vision**; K and Kalto from **proofs of conjectures in linear algebra**

# Experiments

**SageMath** implementation, using **msolve** for zero dimensional polynomial system solving [Berthomieu–Eder–Safei El Din 2021]

**Single thread**;  $\infty$  indicates  $> 25$  days; **OOM** indicates  $> 1.5\text{TB}$  of memory requirement; **Err** indicates an error

Polynomial	CAD SAMPLEPOINTS.MM		[Safei El Din–Schost 2003]		New Algo		HYPERSURFACE REGIONS.JL	
	Time	# of points	Time	# of points	Time	# of points	Time	# of points
Vor1	4	48	230	47	17	53	200	Err
Vor2	$\infty$	—	830 000	87	17 000	149	220 000	23
P3PGen	$\infty$	—	$\infty$	—	80 000	709	8 100	234
K1	OOM	—	11 000	643	280	381	8 400	Err
K2	1	112	9 600	693	290	345	3 400	Err
K3	$\infty$	—	560	355	56	201	1 600	Err
K4	2	224	11 000	717	340	373	6 700	Err
Kalto1–4	$\infty$	—	$\infty$	—	$\infty$	—	$\infty$	—

Computations on **deformed** ( $1/2^{32}$  subtracted) **singular** examples arising from applications, in **seconds**  
Vor from **computational geometry**; P3PGen from **computer vision**; K and Kalto from **proofs of conjectures in linear algebra**

**Pull request** to **SageMath** soon; **Smooth case** available at:  
<https://gillot.perso.lip6.fr/smooth.sage>

# Conclusions & Further Works

- **New algorithms with bit complexity**

$$\tilde{O}(\tau \mathcal{P}(n, \log(1/\epsilon)) \mathfrak{D}^3)$$

where  $\mathfrak{D} \leq \deg(f) \prod_{i=2}^n \deg\left(\frac{\partial f}{\partial X_i}\right)$  reflects **structure**

- **Probability** of success  $\geq 1 - \epsilon$
- **Practical Improvements!**

## Aims:

- **Generalise** to **several** inequalities with **cubic** bit complexity
- **Apply** to problems from **other scientific fields**
- **Study** other **structures**
- **Compare** to satisfiability (**SMT**) problem solvers