A Gröbner-Free Alternative to Solving and a Geometric Analogue to Cook's thesis

Marc Giusti

LIX
Computer Science Lab
CNRS–Polytechnique

ISSAC 2009, Seoul, Korea

MATHEMATICS:

The synthesis of the calculus of n-variables and of n-dimensional geometry is the basis of what Seldon once called "my little algebra of humanity" . . . "

MATHEMATICS:

The synthesis of the calculus of n-variables and of n-dimensional geometry is the basis of what Seldon once called "my little algebra of humanity" . . . "

Isaac Asimov, Second Foundation.

The affine zero dimensional elimination problem: basic step to "PoSSo (polynomial system solving)"

- $\mathbf{k}$ an effective field of characteristic zero, e.g. $\mathbb{Q}$

The affine zero dimensional elimination problem: basic step to "PoSSo (polynomial system solving)"

- $\mathbf{k}$ an effective field of characteristic zero, e.g. $\mathbb{Q}$
- $f_1, \ldots, f_s \in \mathbb{Q}[X_1, \ldots, X_n]$ input polynomials

The affine zero dimensional elimination problem: basic step to "PoSSo (polynomial system solving)"

- $\mathbf{k}$ an effective field of characteristic zero, e.g. $\mathbb{Q}$
- $f_1, \ldots, f_s \in \mathbb{Q}[X_1, \ldots, X_n]$ input polynomials
- the affine space $\mathbb{A}^n := \mathbb{A}^n(\mathbb{C})$
- the algebraic subvariety
  $V = V(f_1, \ldots, f_s) = \{x \in \mathbb{A}^n | f_1 = \cdots = f_s = 0\}$.

The affine zero dimensional elimination problem: basic step to "PoSSo (polynomial system solving)"

- $\mathbf{k}$ an effective field of characteristic zero, e.g. $\mathbb{Q}$
- $f_1, \ldots, f_s \in \mathbb{Q}[X_1, \ldots, X_n]$ input polynomials
- the affine space $\mathbb{A}^n := \mathbb{A}^n(\mathbb{C})$
- the algebraic subvariety
  $V = V(f_1, \ldots, f_s) = \{x \in \mathbb{A}^n | f_1 = \cdots = f_s = 0\}$.

Assume $\dim V = 0$ ($\#V$ finite)

The affine zero dimensional elimination problem: basic step to "PoSSo (polynomial system solving)"

- $\mathbf{k}$ an effective field of characteristic zero, e.g. $\mathbb{Q}$
- $f_1, \ldots, f_s \in \mathbb{Q}[X_1, \ldots, X_n]$ input polynomials
- the affine space $\mathbb{A}^n := \mathbb{A}^n(\mathbb{C})$
- the algebraic subvariety
  $V = V(f_1, \ldots, f_s) = \{x \in \mathbb{A}^n | f_1 = \cdots = f_s = 0\}$.

Assume $\dim V = 0$ ($\#V$ finite)

Problem.

- Given $\ell(X_1, \ldots, X_n)$ a non zero linear form

The affine zero dimensional elimination problem: basic step to "PoSSo (polynomial system solving)"

- $\mathbf{k}$ an effective field of characteristic zero, e.g. $\mathbb{Q}$
- $f_1, \ldots, f_s \in \mathbb{Q}[X_1, \ldots, X_n]$ input polynomials
- the affine space $\mathbb{A}^n := \mathbb{A}^n(\mathbb{C})$
- the algebraic subvariety
  $V = V(f_1, \ldots, f_s) = \{x \in \mathbb{A}^n | f_1 = \cdots = f_s = 0\}$.

Assume $\dim V = 0$ ($\#V$ finite)

Problem.

- Given $\ell(X_1, \ldots, X_n)$ a non zero linear form
- Find a non zero univariate polynomial p and s divisors $g_1, \ldots, g_s$ in $\mathbf{k}[X_1, \ldots, X_n]$ such that the following division holds:

$$p \circ \ell = f_1 g_1 + \cdots + f_s g_s.$$

PoSSo cont.

- The polynomial p is called an elimination polynomial
- Elimination of the variables $X_1, \ldots, X_n$ among the equations $l(X_1, \ldots, X_n) - Y, f_1, \ldots, f_s$

PoSSo cont.

- The polynomial p is called an elimination polynomial
- Elimination of the variables $X_1, \ldots, X_n$ among the equations $l(X_1, \ldots, X_n) - Y, f_1, \ldots, f_s$
- $p \circ \ell$ vanishes on V, hence the zeroes of p are the images of V by ℓ

PoSSo cont.

- The polynomial p is called an elimination polynomial
- Elimination of the variables $X_1, \ldots, X_n$ among the equations $l(X_1, \ldots, X_n) - Y, f_1, \ldots, f_s$
- $p \circ \ell$ vanishes on V, hence the zeroes of p are the images of V by ℓ
- Multivariate problem $\Longrightarrow$ a bunch of univariate problems.

PoSSo cont.

- The polynomial p is called an elimination polynomial
- Elimination of the variables $X_1, \ldots, X_n$ among the equations $l(X_1, \ldots, X_n) - Y, f_1, \ldots, f_s$
- $p \circ \ell$ vanishes on V, hence the zeroes of p are the images of V by ℓ
- Multivariate problem $\Longrightarrow$ a bunch of univariate problems.

- In our minds the problem is solved (at least theoretically ... ;) numerically or symbolically, by recombination of the univariate solvings.

PoSSo cont.

- The polynomial p is called an elimination polynomial
- Elimination of the variables $X_1, \ldots, X_n$ among the equations $l(X_1, \ldots, X_n) - Y, f_1, \ldots, f_s$
- $p \circ \ell$ vanishes on V, hence the zeroes of p are the images of V by ℓ
- Multivariate problem $\Longrightarrow$ a bunch of univariate problems.

- In our minds the problem is solved (at least theoretically ... ;) numerically or symbolically, by recombination of the univariate solvings.
- From a practical point of view, this is another story. How to avoid this recombination step?

PoSSo cont.

- The linear form ℓ is separating if it takes distinct values on distinct points of V
- This condition is generic in Thom's sense.
- Above a zero of p there is only one point of V.

PoSSo cont.

- The linear form ℓ is separating if it takes distinct values on distinct points of V
- This condition is generic in Thom's sense.
- Above a zero of p there is only one point of V.

The variety V is parametrized polynomially by the zeroes of p. This representation goes back at least to Kronecker [82]. It was rediscovered in computer algebra again and again under the names Shape Lemma or Rational Univariate Representation.

From dimension zero to positive one

Let's begin smoothly, as said Asimov.

From dimension zero to positive one

Let's begin smoothly, as said Asimov.

- *Generically* (again in Thom's sense), $f_1, \ldots, f_s$ is a reduced regular sequence of $k[X_1, \ldots, X_n]$ (hence $s \leq n$)

From dimension zero to positive one

Let's begin smoothly, as said Asimov.

- *Generically* (again in Thom's sense), $f_1, \ldots, f_s$ is a reduced regular sequence of $k[X_1, \ldots, X_n]$ (hence $s \leq n$)
- This is just saying that the variety V_i defined by $f_1, \ldots, f_i$, $1 \leq i \leq s$ is reduced, of codimension i, hence is a complete intersection, hence is equidimensional

From dimension zero to positive one

Let's begin smoothly, as said Asimov.

- *Generically* (again in Thom's sense), $f_1, \ldots, f_s$ is a reduced regular sequence of $k[X_1, \ldots, X_n]$ (hence $s \leq n$)
- This is just saying that the variety V_i defined by $f_1, \ldots, f_i$, $1 \leq i \leq s$ is reduced, of codimension i, hence is a complete intersection, hence is equidimensional
- V is of dimension zero if s equals n.

Geometric Resolution (static)

- Geometric version

  Leopold Kronecker: an equidimensional algebraic variety is

  - a (generally ramified) covering of a hyperplane of same dimension
  - birationally equivalent to an hypersurface of same dimension, contained in an affine subspace of dimension one more

Geometric Resolution (static)

- Geometric version

  Leopold Kronecker: an equidimensional algebraic variety is
  - a (generally ramified) covering of a hyperplane of same dimension
  - birationally equivalent to an hypersurface of same dimension, contained in an affine subspace of dimension one more

- Algebraic version

  Emmy Noether: normalization lemma
  - The algebra of functions on an equidimensional algebraic variety is an integral extension of a polynomial algebra

Geometric Resolution (static)

- Geometric version

  Leopold Kronecker: an equidimensional algebraic variety is
  - a (generally ramified) covering of a hyperplane of same dimension
  - birationally equivalent to an hypersurface of same dimension, contained in an affine subspace of dimension one more

- Algebraic version

  Emmy Noether: normalization lemma
  - The algebra of functions on an equidimensional algebraic variety is an integral extension of a polynomial algebra
  - This extension can be realized by a primitive element.

Geometric Resolution (static) cont.

Formally speaking, the Kronecker representation of a a variety of codimension i consists to give explicity the hypersurface and the birational equivalence

Geometric Resolution (static) cont.

Formally speaking, the Kronecker representation of a a variety of codimension i consists to give explicity the hypersurface and the birational equivalence

$$q(x_1,\ldots,x_{n-i},T)=0,\quad \begin{cases} \frac{\partial q}{\partial T}x_{n-i+1} & = & w_{n-i+1}(x_1,\ldots,x_{n-i},T), \\ & \vdots & \\ \frac{\partial q}{\partial T}x_n & = & w_n(x_1,\ldots,x_{n-i},T), \end{cases}$$

where

- $q, w_{n-i+1;},\ldots,w_n$ are polynomials in $\mathbb{Q}[x_1,\ldots,x_{n-i},T]$
- q is monic in T and square free.

TERA: Geometric Resolution (dynamic): TERA

What is important is to give an algorithm yielding the geometric resolution, and not its the result, i.e. the static mathematical form. We can speak of geometric solving

TERA: Geometric Resolution (dynamic): TERA

What is important is to give an algorithm yielding the geometric resolution, and not its the result, i.e. the static mathematical form. We can speak of geometric solving

Moreover of course with a good complexity!

TERA: Geometric Resolution (dynamic): TERA

What is important is to give an algorithm yielding the geometric resolution, and not its the result, i.e. the static mathematical form. We can speak of geometric solving

Moreover of course with a good complexity!

Why not optimal?

TERA: Geometric Resolution (dynamic): TERA

What is important is to give an algorithm yielding the geometric resolution, and not its the result, i.e. the static mathematical form. We can speak of geometric solving
Moreover of course with a good complexity!
Why not optimal?

- Late eighties . . . early nineties [G–Heintz], [G–Heintz–Pardo et al.], . . .

TERA: Geometric Resolution (dynamic): TERA

What is important is to give an algorithm yielding the geometric resolution, and not its the result, i.e. the static mathematical form. We can speak of geometric solving
Moreover of course with a good complexity!
Why not optimal?

- Late eighties . . . early nineties [G–Heintz], [G–Heintz–Pardo et al.], . . .
- Design of efficient algorithms [G–Lecerf–Salvy], [Durvye–Lecerf], [Durvye], [Heintz–Matera–Waissbein]
- TERA group (= MEGA^2 . . . ;)

TERA: Geometric Resolution (dynamic): TERA

What is important is to give an algorithm yielding the geometric resolution, and not its the result, i.e. the static mathematical form. We can speak of geometric solving
Moreover of course with a good complexity!
Why not optimal?

- Late eighties . . . early nineties [G–Heintz], [G–Heintz–Pardo et al.], . . .
- Design of efficient algorithms [G–Lecerf–Salvy], [Durvye–Lecerf], [Durvye], [Heintz–Matera–Waissbein]
- TERA group (= MEGA^2 . . . ;)
  Aldaz, Bank, Beltran, Bostan, Cafure, Castaño, Castro, Colin D'Alfonso, Durvye, Fitchas, G, Grimson, Hägele, Heintz, Jeronimo, Krick, Lecerf, Lehmann, Llovet, Mandel, Marchand, Massaccesi, Matera, M'bakop, Montaña, Morais, Morgenstern, Ollivier, Pardo, Puddu, Sabia, Salvy, Schost, Sedoglavic, Smietanski, Solernó, Turull, Wachenchauzer, Waissbein, . . .

Classical, implicit and naive representation of polynomials and integers

- Polynomials
- Dense representation: coded through the vector of its coefficients
- Size dominated by the total degree

Classical, implicit and naive representation of polynomials and integers

- Polynomials
- Dense representation: coded through the vector of its coefficients
- Size dominated by the total degree

- Integers
- Binary expansion $\longrightarrow$ univariate polynomial
- Degree = bit length = (logarithmic) height

Classical, implicit and naive representation of polynomials and integers

- Polynomials
- Dense representation: coded through the vector of its coefficients
- Size dominated by the total degree

- Integers
- Binary expansion $\longrightarrow$ univariate polynomial
- Degree = bit length = (logarithmic) height

- Height of a polynomial

Geometric Resolution (dynamic): First Main Idea

Don't separate data structures from algorithms!

Geometric Resolution (dynamic): First Main Idea

Don't separate data structures from algorithms!

Elimination polynomials are not hard to evaluate.

Don't separate data structures from algorithms!

Elimination polynomials are not hard to evaluate.

- How many variables they ever may contain, their evaluation complexity is always polynomial in their degree

Don't separate data structures from algorithms!

Elimination polynomials are not hard to evaluate.

- How many variables they ever may contain, their evaluation complexity is always polynomial in their degree
- Whereas their number of monomials may be exponential in the number of their variables.

Don't separate data structures from algorithms!

Elimination polynomials are not hard to evaluate.

- How many variables they ever may contain, their evaluation complexity is always polynomial in their degree
- Whereas their number of monomials may be exponential in the number of their variables.

Evaluation of polynomials can be formalized by the notion of Directed Acyclic Graph (DAG), neither with equality testing nor branching, and executed by a Straight-Line Program (SLP) or arithmetic circuit.

Don't separate data structures from algorithms!

Elimination polynomials are not hard to evaluate.

- How many variables they ever may contain, their evaluation complexity is always polynomial in their degree
- Whereas their number of monomials may be exponential in the number of their variables.

Evaluation of polynomials can be formalized by the notion of Directed Acyclic Graph (DAG), neither with equality testing nor branching, and executed by a Straight-Line Program (SLP) or arithmetic circuit.
Only addition, substraction and products (no divisions) of non scalars are allowed, scalars can be used freely.

Don't separate data structures from algorithms!

Elimination polynomials are not hard to evaluate.

- How many variables they ever may contain, their evaluation complexity is always polynomial in their degree
- Whereas their number of monomials may be exponential in the number of their variables.

Evaluation of polynomials can be formalized by the notion of Directed Acyclic Graph (DAG), neither with equality testing nor branching, and executed by a Straight-Line Program (SLP) or arithmetic circuit.
Only addition, substraction and products (no divisions) of non scalars are allowed, scalars can be used freely.

Extension to the representation of integers [Hägele–Montaña].

Geometric Resolution (dynamic): Complexity Measures

More generally, an algorithm will be described by a general DAG, executed by an arithmetic network.

Geometric Resolution (dynamic): Complexity Measures

More generally, an algorithm will be described by a general DAG, executed by an arithmetic network.

- The sequential complexity of the network is the size of the corresponding DAG, i.e the number of vertices.

Geometric Resolution (dynamic): Complexity Measures

More generally, an algorithm will be described by a general DAG, executed by an arithmetic network.

- The sequential complexity of the network is the size of the corresponding DAG, i.e the number of vertices.
- For SLP's we speak of length, i.e. the number of arithmetic operations

Geometric Resolution (dynamic): Complexity Measures

More generally, an algorithm will be described by a general DAG, executed by an arithmetic network.

- The sequential complexity of the network is the size of the corresponding DAG, i.e the number of vertices.
- For SLP's we speak of length, i.e. the number of arithmetic operations
- Each arithmetic processor can be become a boolean circuit manipulating bits. But we have to take into account the of the height of the scalars involved, to define in the same way the complexity.

Geometric Resolution (dynamic): Complexity of the Input

Input $f_1, \ldots, f_s \in \mathbb{Q}[X_1, \ldots, X_n]$

Geometric Resolution (dynamic): Complexity of the Input

Input $f_1, \ldots, f_s \in \mathbb{Q}[X_1, \ldots, X_n]$

- Dimension n of ambiant space
- Maximal total degree d
- Maximal height h
- Length L of a straight–line program evaluating the input

Geometric Resolution (dynamic): Complexity of the Input

Input $f_1, \ldots, f_s \in \mathbb{Q}[X_1, \ldots, X_n]$

- Dimension n of ambiant space
- Maximal total degree d
- Maximal height h
- Length L of a straight–line program evaluating the input

Input Complexity: the 4-uple (n, d, h, L)

Geometric Resolution (dynamic): Second Main Idea

Intrinsic invariants dominating the complexity of algorithms!

Geometric Resolution (dynamic): Second Main Idea

Intrinsic invariants dominating the complexity of algorithms!

Decreasing sequence $V = V_1 \supset V_2 \supset \cdots \supset V_s$

Geometric Resolution (dynamic): Second Main Idea

Intrinsic invariants dominating the complexity of algorithms!

Decreasing sequence $V = V_1 \supset V_2 \supset \cdots \supset V_s$

- the degree of the polynomial system $\delta := \max(\deg V_i, i = 1, 2, \ldots s)$

Geometric Resolution (dynamic): Second Main Idea

Intrinsic invariants dominating the complexity of algorithms!

Decreasing sequence $V = V_1 \supset V_2 \supset \cdots \supset V_s$

- the degree of the polynomial system $\delta := \max(\deg V_i, i = 1, 2, \ldots s)$
- the height η of the polynomial system defined informally as the maximum height of polynomials occuring during the geometric solving.

Geometric Resolution (dynamic): Second Main Idea

Intrinsic invariants dominating the complexity of algorithms!

Decreasing sequence $V = V_1 \supset V_2 \supset \cdots \supset V_s$

- the degree of the polynomial system $\delta := \max(\deg V_i, i = 1, 2, \ldots s)$
- the height η of the polynomial system defined informally as the maximum height of polynomials occuring during the geometric solving.
- Intrinsic Complexity of the Output: (δ, η)

Geometric Resolution (dynamic): Second Main Idea

Intrinsic invariants dominating the complexity of algorithms!

Decreasing sequence $V = V_1 \supset V_2 \supset \cdots \supset V_s$

- the degree of the polynomial system $\delta := \max(\deg V_i, i = 1, 2, \ldots s)$
- the height η of the polynomial system defined informally as the maximum height of polynomials occuring during the geometric solving.
- Intrinsic Complexity of the Output: (δ, η)
- Extrinsic Complexity of the Output
  - (Ordinary) Bézout Theorem: $\delta \leq d^n$
  - (Arithmetical) Bézout Theorem: $\eta \leq hd^{An}$, A universal constant
  - Degree and height can be arbitrarily smaller.

Complexity of Geometric Resolution

Theorem [GHHMMP 97]. *There exists a bounded error probabilistic Turing machine that from a (reduced) regular sequence of $i \leq n$ polynomials with integer coefficients, of size dominated by (n, d, h, L), outputs a geometric solution.*

Complexity of Geometric Resolution

Theorem [GHHMMP 97]. *There exists a bounded error probabilistic Turing machine that from a (reduced) regular sequence of $i \leq n$ polynomials with integer coefficients, of size dominated by (n, d, h, L), outputs a geometric solution.*
The time complexity (number of bit operations executed) is linear in L and polynomial in $ndh\delta\eta$, if intermediate and output polynomials are coded by straight-line programs.

Complexity of Geometric Resolution

Theorem [GHHMMP 97]. *There exists a bounded error probabilistic Turing machine that from a (reduced) regular sequence of $i \leq n$ polynomials with integer coefficients, of size dominated by (n, d, h, L), outputs a geometric solution.*
The time complexity (number of bit operations executed) is linear in L and polynomial in $ndh\delta\eta$, if intermediate and output polynomials are coded by straight-line programs.
Furthermore if this representation by evaluation is extended to integers, the complexity is no longer dependent on η.

Complexity of Geometric Resolution

Theorem [GHHMMP 97]. *There exists a bounded error probabilistic Turing machine that from a (reduced) regular sequence of $i \leq n$ polynomials with integer coefficients, of size dominated by (n, d, h, L), outputs a geometric solution.*
The time complexity (number of bit operations executed) is linear in L and polynomial in $ndh\delta\eta$, if intermediate and output polynomials are coded by straight-line programs.
Furthermore if this representation by evaluation is extended to integers, the complexity is no longer dependent on η.

This is a *theoretical* theorem. Anyway we refer from now on to evaluation techniques if SLP encoding is used.

Improvement and Extensions of Evaluation Techniques

- Hypothesis: reduced regular sequence

Improvement and Extensions of Evaluation Techniques

- Hypothesis: reduced regular sequence
- Enough if fulfilled outside a given hypersurface $g = 0$.
  - Complexity measures attached to the sequence $f_1, \ldots, f_s, g$ and the Zariski closure of $V_i \setminus V(g)$.
  - Same class of complexity for the algorithm

Improvement and Extensions of Evaluation Techniques

- Hypothesis: reduced regular sequence
- Enough if fulfilled outside a given hypersurface $g = 0$.
  - Complexity measures attached to the sequence $f_1, \ldots, f_s, g$ and the Zariski closure of $V_i \setminus V(g)$.
  - Same class of complexity for the algorithm
- Linear in L, quasi-quadratic in δ

Improvement and Extensions of Evaluation Techniques

- Hypothesis: reduced regular sequence
- Enough if fulfilled outside a given hypersurface $g = 0$.
  - Complexity measures attached to the sequence $f_1, \ldots, f_s, g$ and the Zariski closure of $V_i \setminus V(g)$.
  - Same class of complexity for the algorithm
- Linear in L, quasi-quadratic in δ
- Drop the hypothesis $\Longrightarrow$ general case, arbitrary input
  Equidimensional decomposition, irreducible decomposition modulo factorization

Improvement and Extensions of Evaluation Techniques

- Hypothesis: reduced regular sequence
- Enough if fulfilled outside a given hypersurface $g = 0$.
  - Complexity measures attached to the sequence $f_1, \ldots, f_s, g$ and the Zariski closure of $V_i \setminus V(g)$.
  - Same class of complexity for the algorithm
- Linear in L, quasi-quadratic in δ
- Drop the hypothesis $\Longrightarrow$ general case, arbitrary input
  Equidimensional decomposition, irreducible decomposition modulo factorization
- Primary decomposition of an arbitrary variety of dimension 0 outside an hypersurface.
  Multiplicities and even more: description of the local algebras.

Implementation

- Drop the straight–line programs

Implementation

- Drop the straight–line programs
- Replaced by efficient use of specialization

Implementation

- Drop the straight–line programs
- Replaced by efficient use of specialization
- Evaluation techniques reminiscent of deforestation techniques [Wadler]

Implementation

- Drop the straight–line programs
- Replaced by efficient use of specialization
- Evaluation techniques reminiscent of deforestation techniques [Wadler]

– Software `KRONECKER`

– Package written in `MAGMA` by G. Lecerf, improvements by Schost and Lehmann

– Home page of Lecerf:
`http://www.math.uvsq.fr/`

Miscelleanous

– Polynomial Invariants under finite groups [Colin–G]

- Evaluation techniques are well adapted to handle polynomials invariants under the action of a finite subgroup H of the general linear group.
- Primary invariants are besser handled with an evaluation data structure

Miscelleanous

– Polynomial Invariants under finite groups [Colin–G]

- Evaluation techniques are well adapted to handle polynomials invariants under the action of a finite subgroup H of the general linear group.
- Primary invariants are besser handled with an evaluation data structure
- Permutation subgroup of the symmetric group:
- Lagrange resolvents can be computed in polynomial time in the index [C–G] + [Dahan, Schost, Wu]

Miscelleanous

– Polynomial Invariants under finite groups [Colin–G]

- Evaluation techniques are well adapted to handle polynomials invariants under the action of a finite subgroup H of the general linear group.
- Primary invariants are besser handled with an evaluation data structure
- Permutation subgroup of the symmetric group:
- Lagrange resolvents can be computed in polynomial time in the index [C–G] + [Dahan, Schost, Wu]

– Ritt's resolvent of a prime differential ring [Solerno et al.]

Lower Bounds and Optimality Considerations

[Heintz–Matera–Pardo–Wachenchauzer], [G–Heintz], [Castro–G–Heintz–Matera–Pardo])

Lower Bounds and Optimality Considerations

[Heintz–Matera–Pardo–Wachenchauzer], [G–Heintz], [Castro–G–Heintz–Matera–Pardo])

Folklore: Elimination and Cook's question are related. Here is a precise statement due to [Heintz–Morgenstern 1993]:

Lower Bounds and Optimality Considerations

[Heintz–Matera–Pardo–Wachenchauzer], [G–Heintz], [Castro–G–Heintz–Matera–Pardo])

Folklore: Elimination and Cook's question are related. Here is a precise statement due to [Heintz–Morgenstern 1993]:

Theorem *If $\mathcal{P} \neq \mathcal{NP}$ holds, there exists no uniform well parallelizable algorithm such that:*

- *the input $f_1, \ldots, f_s \in \mathbf{k}[X_1, \ldots, X_n]$ is given by a well parallelizable division–free straight–line program of length L*
- *it solves the affine zero dimensional problem in sequential time $(Ld)^{O(1)}$*
- *the output polynomial $Q(Y)$ belonging to $(f_1, \ldots, f_s)$ is of degree $\deg V$ and is given by a well parallelizable division–free straight–line program of length $(Ld)^{O(1)}$.*

What do we expect from elimination theory?

Elimination procedure:

$$\varepsilon : W \longrightarrow \mathbb{C}^D$$

What do we expect from elimination theory?

Elimination procedure:

$$\varepsilon : W \longrightarrow \mathbb{C}^D$$

- W is a constructible subset of $\mathbb{C}^m$ admitting a short syntactical description of length polynomial in $\dim W$.
- $\mathbb{C}^D$ semantical objects.

What do we expect from elimination theory?

Elimination procedure:

$$\varepsilon : W \longrightarrow \mathbb{C}^D$$

- W is a constructible subset of $\mathbb{C}^m$ admitting a short syntactical description of length polynomial in $\dim W$.
- $\mathbb{C}^D$ semantical objects.

Data base and queries:

$$\Theta : \mathbb{C}^D \longrightarrow \mathbb{C}^\ell$$

- Θ represents questions

What do we expect from elimination theory?

Elimination procedure:

$$\varepsilon : W \longrightarrow \mathbb{C}^D$$

- W is a constructible subset of $\mathbb{C}^m$ admitting a short syntactical description of length polynomial in $\dim W$.
- $\mathbb{C}^D$ semantical objects.

Data base and queries:

$$\Theta : \mathbb{C}^D \longrightarrow \mathbb{C}^\ell$$

- Θ represents questions
- C^ℓ space of answers

Example

Conditions:

- $f_1, \ldots, f_n$ reduced regular sequence
- $q \in \mathbb{C}[X_1, \ldots, X_n]$
- $\deg f_i, \deg q \leq d$
- $degV := V(f_1, \ldots, f_n) \leq D$

Example

Conditions:

- $f_1, \ldots, f_n$ reduced regular sequence
- $q \in \mathbb{C}[X_1, \ldots, X_n]$
- $\deg f_i, \deg q \leq d$
- $degV := V(f_1, \ldots, f_n) \leq D$

yield a constructible $W \subset \mathbb{C}^m$ with $m = (n+1)\binom{d+n}{n}$.

Eliminating Polynomial

Generalizing the case we begin with, where q was linear

- Image of $q : V(f_1, \ldots, f_n) \longrightarrow \mathbb{C}$?

Eliminating Polynomial

Generalizing the case we begin with, where q was linear

- Image of $q : V(f_1, \ldots, f_n) \longrightarrow \mathbb{C}$?
- $\prod_{x \in V}(y - q(x))$

Eliminating Polynomial

Generalizing the case we begin with, where q was linear

- Image of $q : V(f_1, \ldots, f_n) \longrightarrow \mathbb{C}$?
- $\prod_{x \in V}(y - q(x))$
- Multiplication by q induces an endomorphism $\mathbb{C}[X_1, \ldots, X_n]$

Eliminating Polynomial

Generalizing the case we begin with, where q was linear

- Image of $q : V(f_1, \ldots, f_n) \longrightarrow \mathbb{C}$?
- $\prod_{x \in V}(y - q(x))$
- Multiplication by q induces an endomorphism $\mathbb{C}[X_1, \ldots, X_n]$
- χ_q its characteristic polynomial.

Eliminating Polynomial

Generalizing the case we begin with, where q was linear

- Image of $q : V(f_1, \ldots, f_n) \longrightarrow \mathbb{C}$?
- $\prod_{x \in V}(y - q(x))$
- Multiplication by q induces an endomorphism $\mathbb{C}[X_1, \ldots, X_n]$
- χ_q its characteristic polynomial.

Questions?

Questions

Nullstellensatz

- $V(f_1, \ldots, f_n, q) = \emptyset$?
- q vanishes on some point of V?
- Constant term of χ_q is 0?
- $pr : \mathbb{C}^D \longrightarrow \mathbb{C}$ zero or not?

Questions

Nullstellensatz

- $V(f_1, \ldots, f_n, q) = \emptyset$?
- q vanishes on some point of V?
- Constant term of χ_q is 0?
- $pr : \mathbb{C}^D \longrightarrow \mathbb{C}$ zero or not?

Ideal Membership

- $q \in (f_1, \ldots, f_n)$?
- q vanishes on V?
- All non leading terms χ_q are 0?
- $Id : \mathbb{C}^D \longrightarrow \mathbb{C}^D$ zero or not?

Size Remarks

- Assume d small compared to n: typically $d = 2$

Size Remarks

- Assume d small compared to n: typically $d = 2$
- The source dimension m of the syntactical input is typically polynomial in n: $(n+1)\binom{d+n}{d}$

Size Remarks

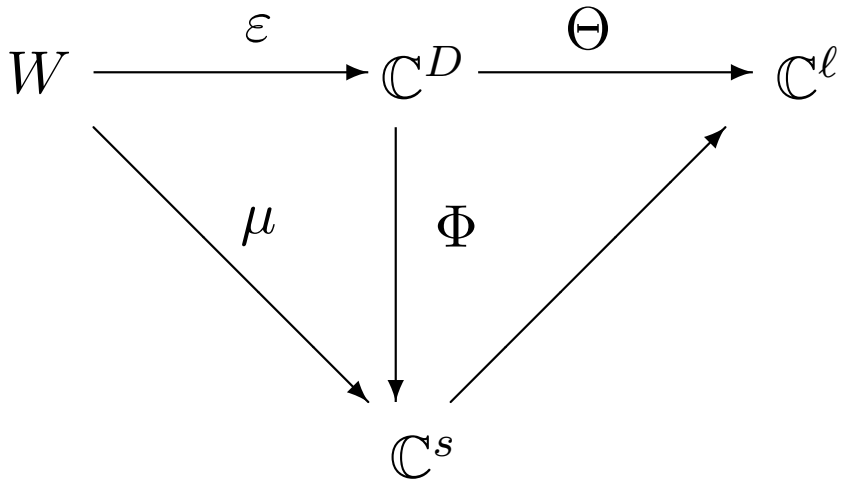
- Assume d small compared to n: typically $d = 2$
- The source dimension m of the syntactical input is typically polynomial in n: $(n+1)\binom{d+n}{d}$
- the target dimension D of the semantical output is typically exponential in n (Bézout: 2^n)

Size Remarks

- Assume d small compared to n: typically $d = 2$
- The source dimension m of the syntactical input is typically polynomial in n: $(n+1)\binom{d+n}{d}$
- the target dimension D of the semantical output is typically exponential in n (Bézout: 2^n)

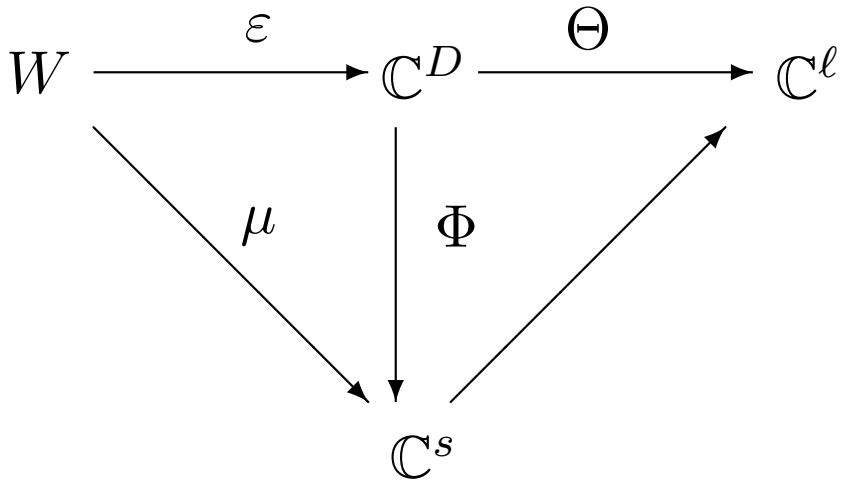
From now on we will be in this situation, and D will be d^n.

Alternative: Change the Data Structure



Dense representation : explicit writing of ε too large. Try:

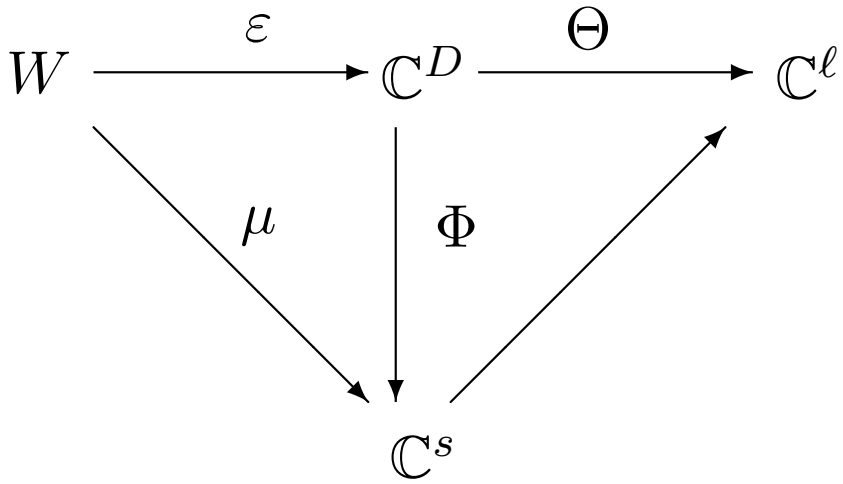
Alternative: Change the Data Structure



Dense representation : explicit writing of ε too large. Try:

- More efficient data structure?

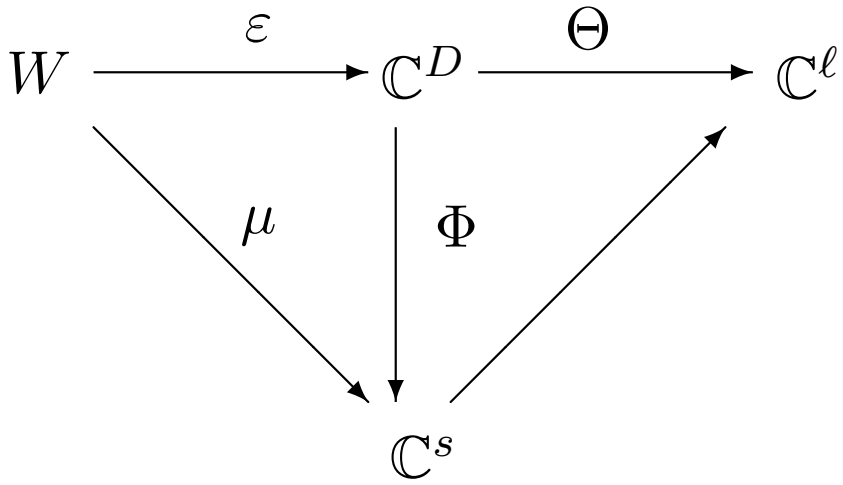
Alternative: Change the Data Structure



Dense representation : explicit writing of ε too large. Try:

- More efficient data structure?
- Encoding space $\mathbb{C}^s$, data structure Φ, s length of output encoding

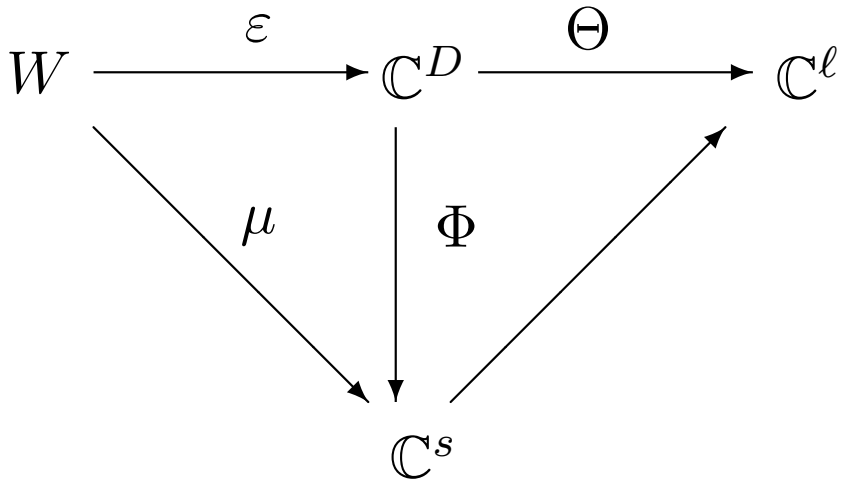
Alternative: Change the Data Structure



Dense representation : explicit writing of ε too large. Try:

- More efficient data structure?
- Encoding space $\mathbb{C}^s$, data structure Φ, s length of output encoding
- Certified data structure: Φ restricted to $\mathrm{Im}\ \varepsilon$ is injective

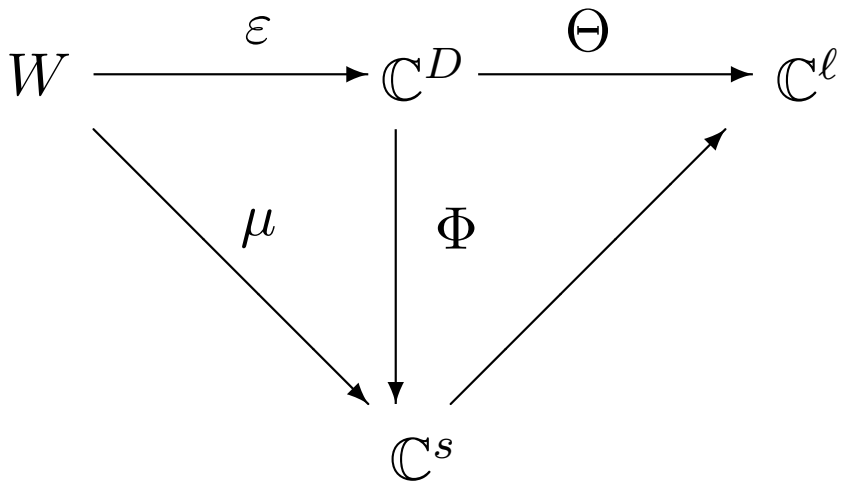
Alternative: Change the Data Structure



Dense representation : explicit writing of ε too large. Try:

- More efficient data structure?
- Encoding space $\mathbb{C}^s$, data structure Φ, s length of output encoding
- Certified data structure: Φ restricted to $\mathrm{Im}\ \varepsilon$ is injective
- Interpolation map $\mathrm{Im}\ \Phi \longrightarrow \mathrm{Im}\ \varepsilon$

Alternative: Change the Data Structure



Dense representation : explicit writing of ε too large. Try:

- More efficient data structure?
- Encoding space $\mathbb{C}^s$, data structure Φ, s length of output encoding
- Certified data structure: Φ restricted to $\mathrm{Im}\ \varepsilon$ is injective
- Interpolation map $\mathrm{Im}\ \Phi \longrightarrow \mathrm{Im}\ \varepsilon$
- Black–box $\mu = \Phi \circ \varepsilon$

Correct Test Sequences [Heintz–Schnorr 82]

- W is the set of polynomials in n variables, of bounded given degree and bounded given evaluation length

Correct Test Sequences [Heintz–Schnorr 82]

- W is the set of polynomials in n variables, of bounded given degree and bounded given evaluation length
- Encoding space $\mathbb{C}^s$, $s = 8 + 16 \dim W$

Correct Test Sequences [Heintz–Schnorr 82]

- W is the set of polynomials in n variables, of bounded given degree and bounded given evaluation length
- Encoding space $\mathbb{C}^s$, $s = 8 + 16 \dim W$
- Certified data structure: Φ is the restriction to $\mathrm{Im}\ \varepsilon$ of an injective map

Correct Test Sequences [Heintz–Schnorr 82]

- W is the set of polynomials in n variables, of bounded given degree and bounded given evaluation length
- Encoding space $\mathbb{C}^s$, $s = 8 + 16 \dim W$
- Certified data structure: Φ is the restriction to $\mathrm{Im}\ \varepsilon$ of an injective map
- Such maps are everywhere dense!

Question answered: when $\varepsilon(w) = 0$?

- There are elimination procedures, suitably encoded, whose output length is linear in the input length

Correct Test Sequences [Heintz–Schnorr 82]

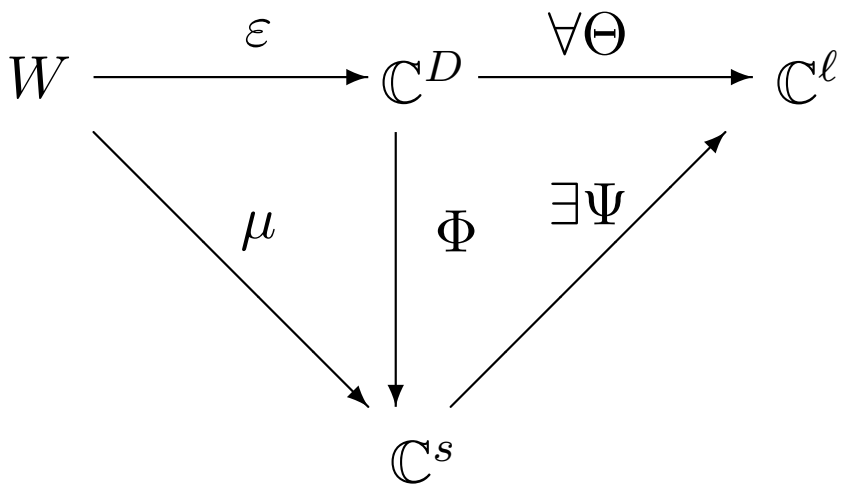
- W is the set of polynomials in n variables, of bounded given degree and bounded given evaluation length
- Encoding space $\mathbb{C}^s$, $s = 8 + 16 \dim W$
- Certified data structure: Φ is the restriction to $\mathrm{Im}\ \varepsilon$ of an injective map
- Such maps are everywhere dense!

Question answered: when $\varepsilon(w) = 0$?

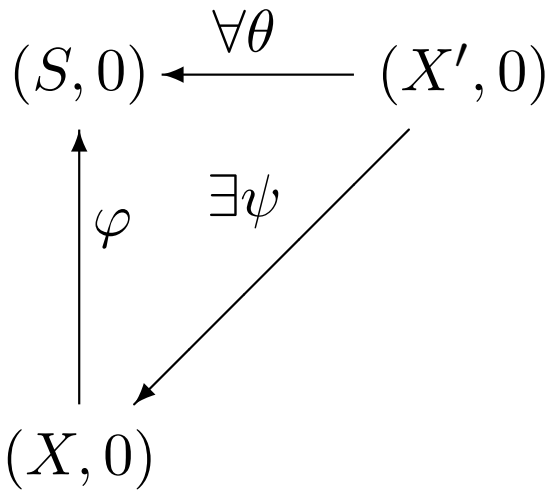
- There are elimination procedures, suitably encoded, whose output length is linear in the input length
- No exponential time lower bound for Elimination Theory can be found only by means of output length

Universal Elimination Procedures

Their output, suitably encoded, contains information enough to answer ANY question you may ask in the future on any instance of W.



Convincing the pure mathematician



- Distinguished point 0
- $\chi_g = T^D$
- Thom's organizing center: worst singularity!
- This corresponds to the worst problem, i.e. the ideal membership!!
- The dimension of the basis of the (almost) universal unfolding of T^D is $D-1$, hence the length of the output encoding s is exponential in the input length.

Conclusion: Universal Elimination Requires Exponential Time

- Efficient Elimination Methods must look for either partial answers discarding some questions
- Or accept to run in exponential time!

Conclusion: Universal Elimination Requires Exponential Time

- Efficient Elimination Methods must look for either partial answers discarding some questions
- Or accept to run in exponential time!

Even no further hope in direction of numerical analysis data structures.

Convincing the computer scientist

Any elimination procedure which is:

- parametric
- parsimonious with respect to *branchings* and *divisions*
- must necessarily have a non–polynomial sequential time complexity
- even if highly efficient data structures (as e.g. the arithmetic circuit encoding of polynomials) are used.

Convincing the numerical analyst: the hardness of Polynomial System Solving

- Let be given an arbitrary continuous data structure encoding systems of polynomials
- together with this data structure a universal elimination algorithm, say $\mathcal{P}$,
- solving arbitrary parametric polynomial equation systems.

Convincing the numerical analyst: the hardness of Polynomial System Solving

- Let be given an arbitrary continuous data structure encoding systems of polynomials
- together with this data structure a universal elimination algorithm, say $\mathcal{P}$,
- solving arbitrary parametric polynomial equation systems.
- Suppose that the algorithm $\mathcal{P}$ avoids "unnecessary" branchings and that $\mathcal{P}$ admits the efficient computation of certain natural limit objects (as e.g. the Zariski closure of a given constructible algebraic set or the parametric greatest common divisor of two given algebraic families of univariate polynomials).

Convincing the numerical analyst: the hardness of Polynomial System Solving

- Let be given an arbitrary continuous data structure encoding systems of polynomials
- together with this data structure a universal elimination algorithm, say $\mathcal{P}$,
- solving arbitrary parametric polynomial equation systems.
- Suppose that the algorithm $\mathcal{P}$ avoids "unnecessary" branchings and that $\mathcal{P}$ admits the efficient computation of certain natural limit objects (as e.g. the Zariski closure of a given constructible algebraic set or the parametric greatest common divisor of two given algebraic families of univariate polynomials).
- Then $\mathcal{P}$ cannot be a polynomial time algorithm.

REAL SOLVING [Bank–G–H–Lehmann–P]

Motivation: Wavelet construction via algorithmic real algebraic geometry

REAL SOLVING [Bank–G–H–Lehmann–P]

Motivation: Wavelet construction via algorithmic real algebraic geometry

During the last three decades discrete wavelet transforms arose as an important tool in signal analysis and in data compression (e.g. of picture or audio signals).

REAL SOLVING [Bank–G–H–Lehmann–P]

Motivation: Wavelet construction via algorithmic real algebraic geometry

During the last three decades discrete wavelet transforms arose as an important tool in signal analysis and in data compression (e.g. of picture or audio signals).

The wavelet transforms we consider shall possess the practical important properties of symmetry and orthogonality. The specification of such a wavelet transform depends on a finite number of real parameters. Those parameters have to obey certain polynomial equations.

REAL SOLVING [Bank–G–H–Lehmann–P]

Motivation: Wavelet construction via algorithmic real algebraic geometry

During the last three decades discrete wavelet transforms arose as an important tool in signal analysis and in data compression (e.g. of picture or audio signals).

The wavelet transforms we consider shall possess the practical important properties of symmetry and orthogonality. The specification of such a wavelet transform depends on a finite number of real parameters. Those parameters have to obey certain polynomial equations.

If the system of those equations has real solutions at all, the solution set can consist of a finite number of points or can be a variety of positive dimension.

In the literature published on this topic, only example problems with a finite solution set were presented. For the computation of those examples it was sufficient to solve quadratic equations in one or two variables. This is easily done with the help of the tools of common computer algebra systems.

In the literature published on this topic, only example problems with a finite solution set were presented. For the computation of those examples it was sufficient to solve quadratic equations in one or two variables. This is easily done with the help of the tools of common computer algebra systems.

Examples with real solution sets of positive dimension have the advantage that one can search for optimal solutions for some given, desired properties.

In the literature published on this topic, only example problems with a finite solution set were presented. For the computation of those examples it was sufficient to solve quadratic equations in one or two variables. This is easily done with the help of the tools of common computer algebra systems.

Examples with real solution sets of positive dimension have the advantage that one can search for optimal solutions for some given, desired properties.

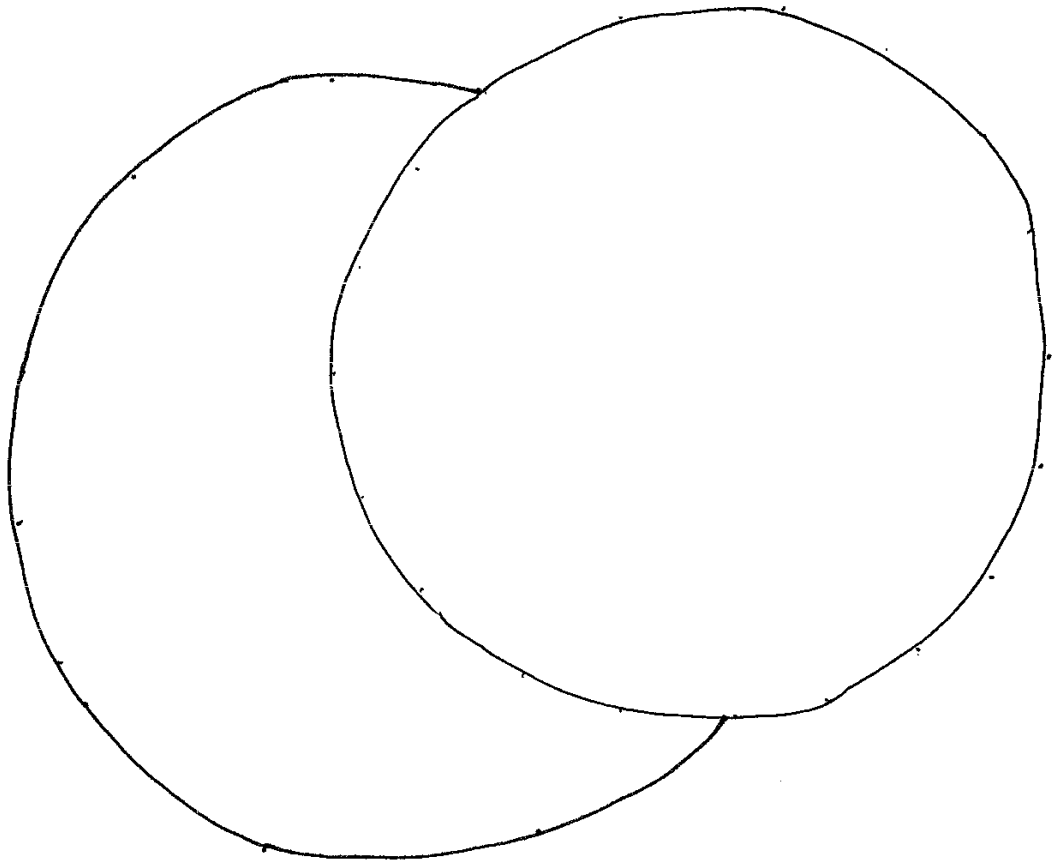
To characterize the set of real solutions of a system of polynomial equations it is a first step to find at least one point in each connected component.

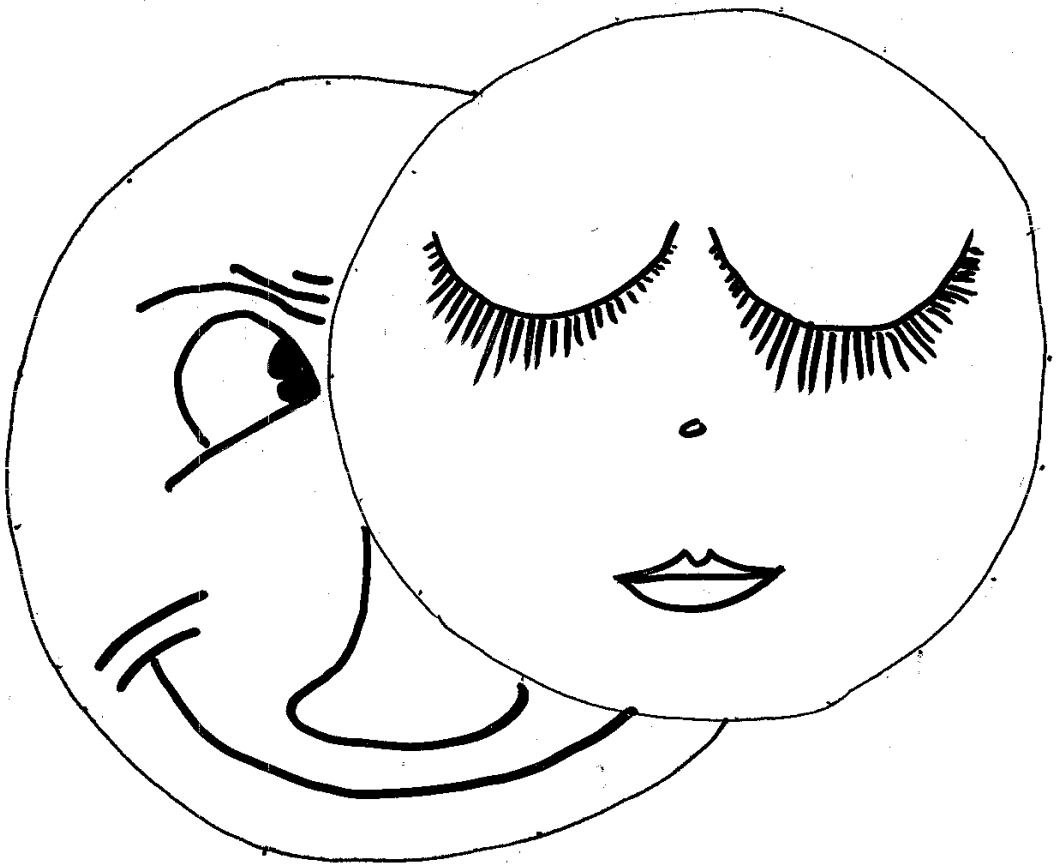
In the literature published on this topic, only example problems with a finite solution set were presented. For the computation of those examples it was sufficient to solve quadratic equations in one or two variables. This is easily done with the help of the tools of common computer algebra systems.

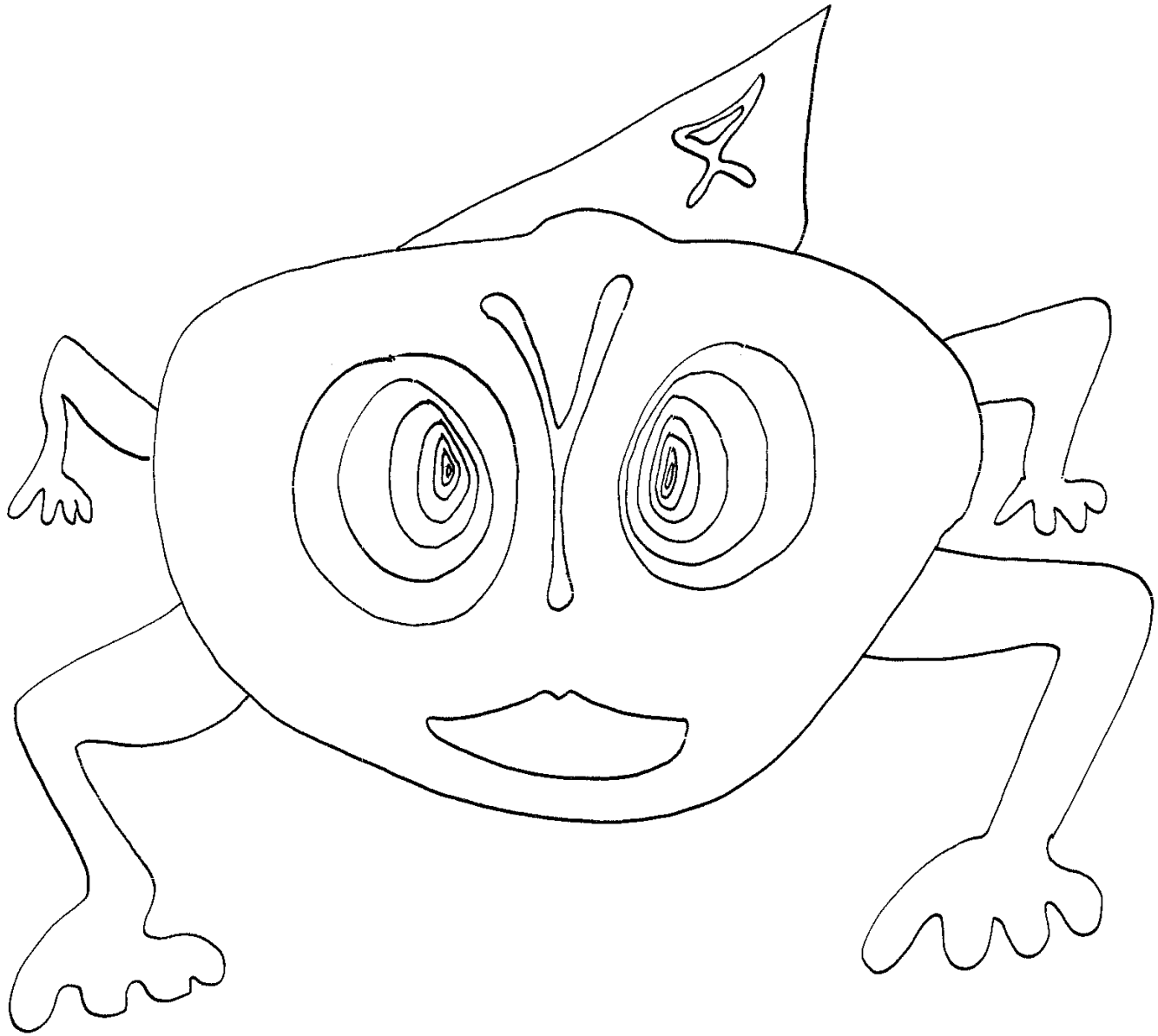
Examples with real solution sets of positive dimension have the advantage that one can search for optimal solutions for some given, desired properties.

To characterize the set of real solutions of a system of polynomial equations it is a first step to find at least one point in each connected component.

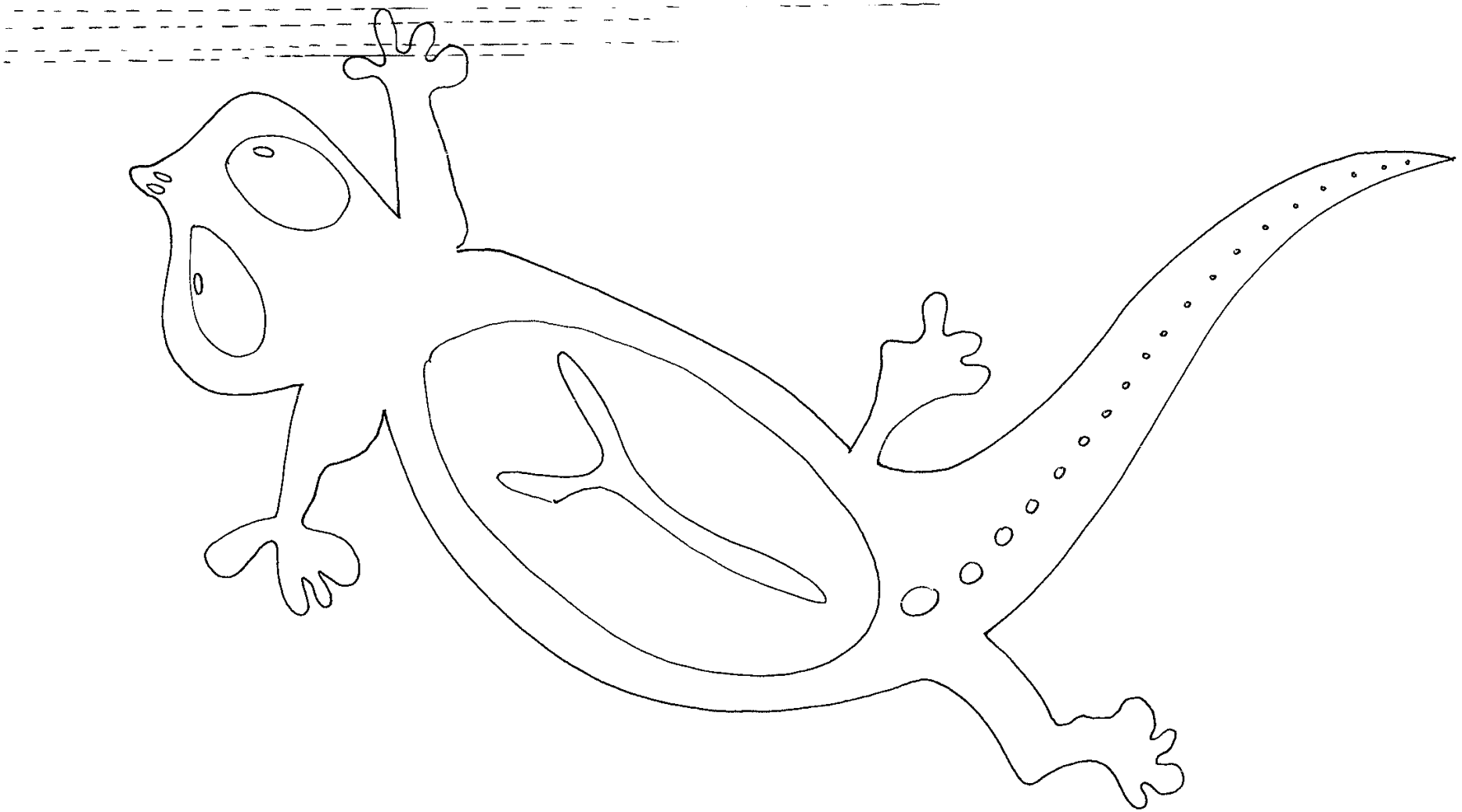
It turns out that the algorithm of the TERA–project performes very well with this task and is able to solve a larger number of examples than the best known commercial polynomial solvers.

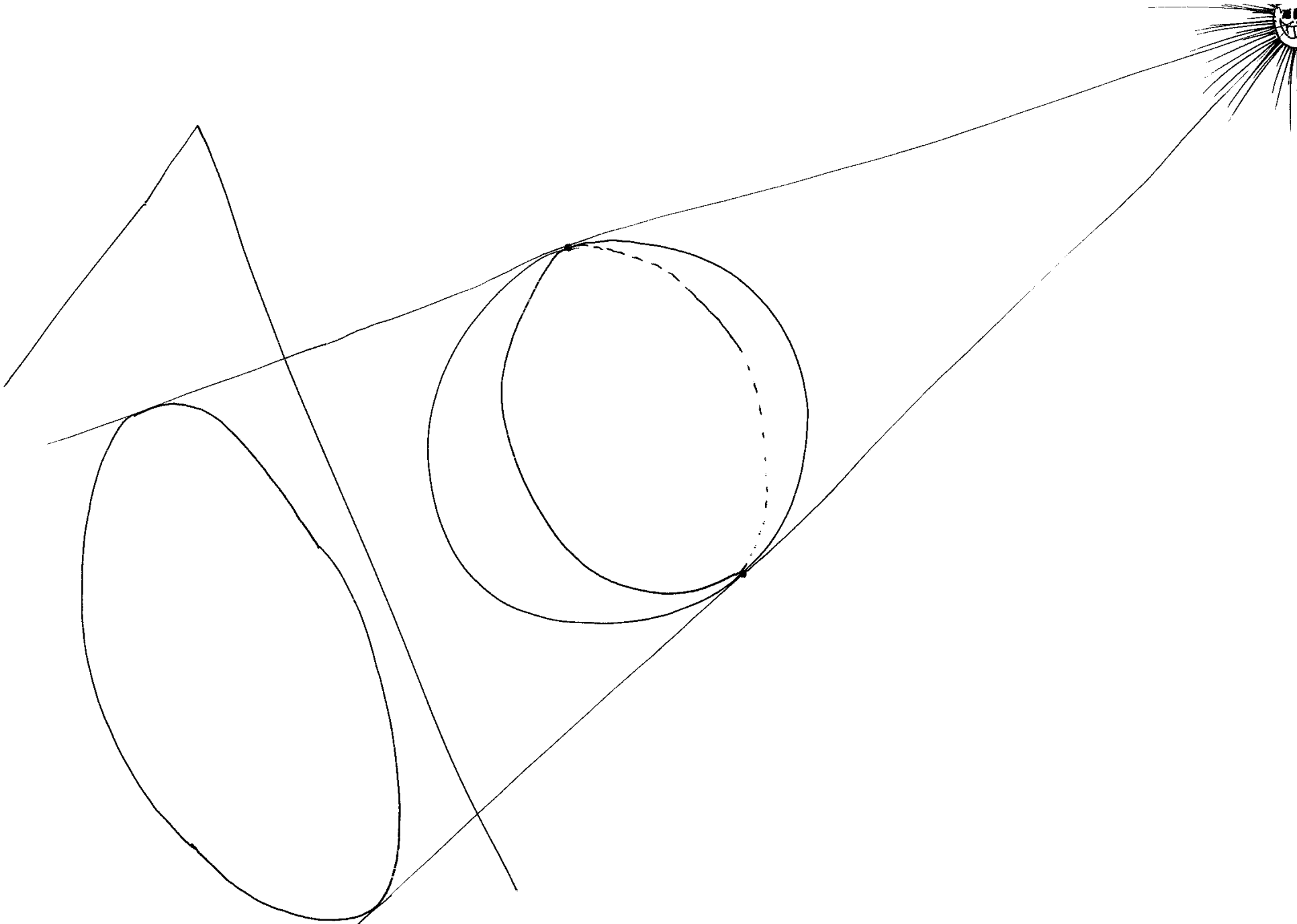






An algebraic curve





Introduction

- Polar varieties are objects coming from classical algebraic geometry. When generic they provide a tool for the design of algorithms in real algebraic geometry that exhibit an

  intrinsic complexity.

Introduction

- Polar varieties are objects coming from classical algebraic geometry. When generic they provide a tool for the design of algorithms in real algebraic geometry that exhibit an

  intrinsic complexity.

- In particular to find efficiently one representative point in every connected component of a real algebraic variety constitutes a fundamental problem in real algebraic geometry. We tackle this problem by exploiting the

  geometric solving of generic or (sufficiently generic) polar varieties (by the algorithm `Kronecker`)

Introduction

- Polar varieties are objects coming from classical algebraic geometry. When generic they provide a tool for the design of algorithms in real algebraic geometry that exhibit an

  intrinsic complexity.

- In particular to find efficiently one representative point in every connected component of a real algebraic variety constitutes a fundamental problem in real algebraic geometry. We tackle this problem by exploiting the

  geometric solving of generic or (sufficiently generic) polar varieties (by the algorithm `Kronecker`)

- Several generalisations of the notion of polar variety allows us to drop successively the assumptions of

  hypersurface, compactness and smoothness.

Polar varieties: Notations

Let A be a **generic** $((n-1)\times n)$–matrix.

For an index i, fixed between 1 and $n-1$, we denote by

$$A_i := \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n-i,1} & a_{n-i,2} & \cdots & a_{n-i,n} \end{bmatrix},$$

the sub–matrix of A formed by the lines $1, 2, \ldots, n-i$ of A.

- $f \in \mathbb{Q}[x_1, \ldots, x_n]$
- $\mathcal{V} := \mathcal{V}_{\mathbb{C}}(f) := \{x \in \mathbb{C}^n \mid f(x) = 0\}$
- $\mathcal{V}_{\mathbb{R}}$ its real trace $\mathcal{V}_{\mathbb{C}} \cap {\mathbb{A}_{\mathbb{R}}}^n$

Polar varieties: Definitions

The i–th (open) polar variety of $\mathcal{V}_{\mathbb{R}}$

is defined by

$$\Delta_i(A) := \{x \in \mathcal{V}_{\mathbb{R}} \setminus Sing\ \mathcal{V}_{\mathbb{R}} \mid T_x\ \mathcal{V}_{\mathbb{R}} \text{ not transversal to } \ker A_i\}.$$

$\Delta_i(A)$ is the set of all points in $\mathcal{V}_{\mathbb{R}}$,
where the $((1+n-i)\times n)$–matrix

$$\mathcal{A}_i(x) := \begin{bmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_n} \\ & A_i & \end{bmatrix} \text{ is not of maximal rank, i.e.,}$$

$$rank\ \mathcal{A}_i(x) \leq n-i.$$

Polar varieties: Definitions

The i–th (open) polar variety of $\mathcal{V}_{\mathbb{R}}$

is defined by

$$\Delta_i(A) := \{x \in \mathcal{V}_{\mathbb{R}} \setminus Sing\ \mathcal{V}_{\mathbb{R}} \mid T_x\ \mathcal{V}_{\mathbb{R}} \text{ not transversal to } \ker A_i\}.$$

$\Delta_i(A)$ is the set of all points in $\mathcal{V}_{\mathbb{R}}$,
where the $((1+n-i)\times n)$–matrix

$$\mathcal{A}_i(x) := \begin{bmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_n} \\ & A_i & \end{bmatrix} \text{ is not of maximal rank, i.e.,}$$

$$rank\ \mathcal{A}_i(x) \leq n-i.$$

The i–th (closed) polar variety of $\mathcal{V}_{\mathbb{R}}$

is defined as the Zariski–closure of the i–th open polar variety.

Polar varieties: Determinantal Equations in the Smooth Case

- If the hypersurface $\mathcal{V}_{\mathbb{R}}$ is smooth, one obtains natural equations by adjoining to $f = 0$ all maximal minors of the matrix $\mathcal{A}_i(x)$.
- According to the maximal codimension of determinantal varieties, the expected codimension of the i–th polar variety is i,

$$n - (1 + n - i) + 1 = i \ .$$

- Corresponding to the flag

$$\ker A_{n-1} \subset \ldots \subset \ker A_1$$

the polar varieties form a decreasing sequence of sub–varieties of $\mathcal{V}$ with expected codimension $1, \ldots, n-1$.

Notations and problem

- $f \in \mathbb{Q}[x_1, \ldots, x_n]$
- $\mathcal{V} := \mathcal{V}_{\mathbb{C}}(f) := \{x \in \mathbb{C}^n \mid f(x) = 0\}$
- $\mathcal{V}_{\mathbb{R}}$ its real trace $\mathcal{V}_{\mathbb{C}} \cap {\mathbb{A}^n}_{\mathbb{R}}$
- The equation of the hypersurface $f = 0$ is assumed to be regular, i.e. : $\mathcal{V}_{\mathbb{R}}$ is non empty and $grad(f)$ does not vanish on any of its connected component.
- Degree of $f \leq d$

FIND AT LEAST A POINT IN EVERY CONNECTED COMPONENT OF $\mathcal{V}_{\mathbb{R}}$

Small history of the problem

- Grigoriev, Grigoriev/Vorobjov '87 , '88
- Complexity $d^{O(n)}$:
  Heintz/Roy/Solerno '89, '90
  Barvinok '91
  Renegar '92, ,'98
  Canny/Emiris, Canny '95, '98
  Blum/Cucker/Shub/Smale '97
  Cucker/Smale '98
  Basu/Pollack/Roy '95, '98
- More recent papers:
  Rouillier/Roy/Safey el Din
  Aubry/Rouillier/Safey el Din
  Safey el Din/Schost, Safey el Din
  ...
- Our contributions '97, '98, '00, '02, '05, '07, '08

Polar Varieties: Quantification of Data Complexity

- $f \in \mathbb{Q}[x_1, \ldots x_n]$ regular equation of a real smooth hypersurface

$$\mathcal{V}_{\mathbb{R}} = f^{-1}(0) \subset \mathbb{R}^n, \quad \mathcal{V}_{\mathbb{R}} \text{ compact or not}$$

- Degree of $f \leq d$
- Evaluation complexity of $f \leq L$
- Maximal **degree of the complex polar varieties** δ;
  (always holds $\delta \leq d^n$, $\quad d^n$ is the **Bézout** number)
- If $\mathcal{V}_{\mathbb{R}}$ is not compact we use **generalized** polar varieties (see later).

The smooth past: find a point in every connected component of $\mathcal{V}_{\mathbb{R}}$ when smooth

: Complexity Theorem

- **Intrinsic Complexity** (number of arithmetic operations in $\mathbb{Q}$)

$$L(nd\delta)^{O(1)}$$

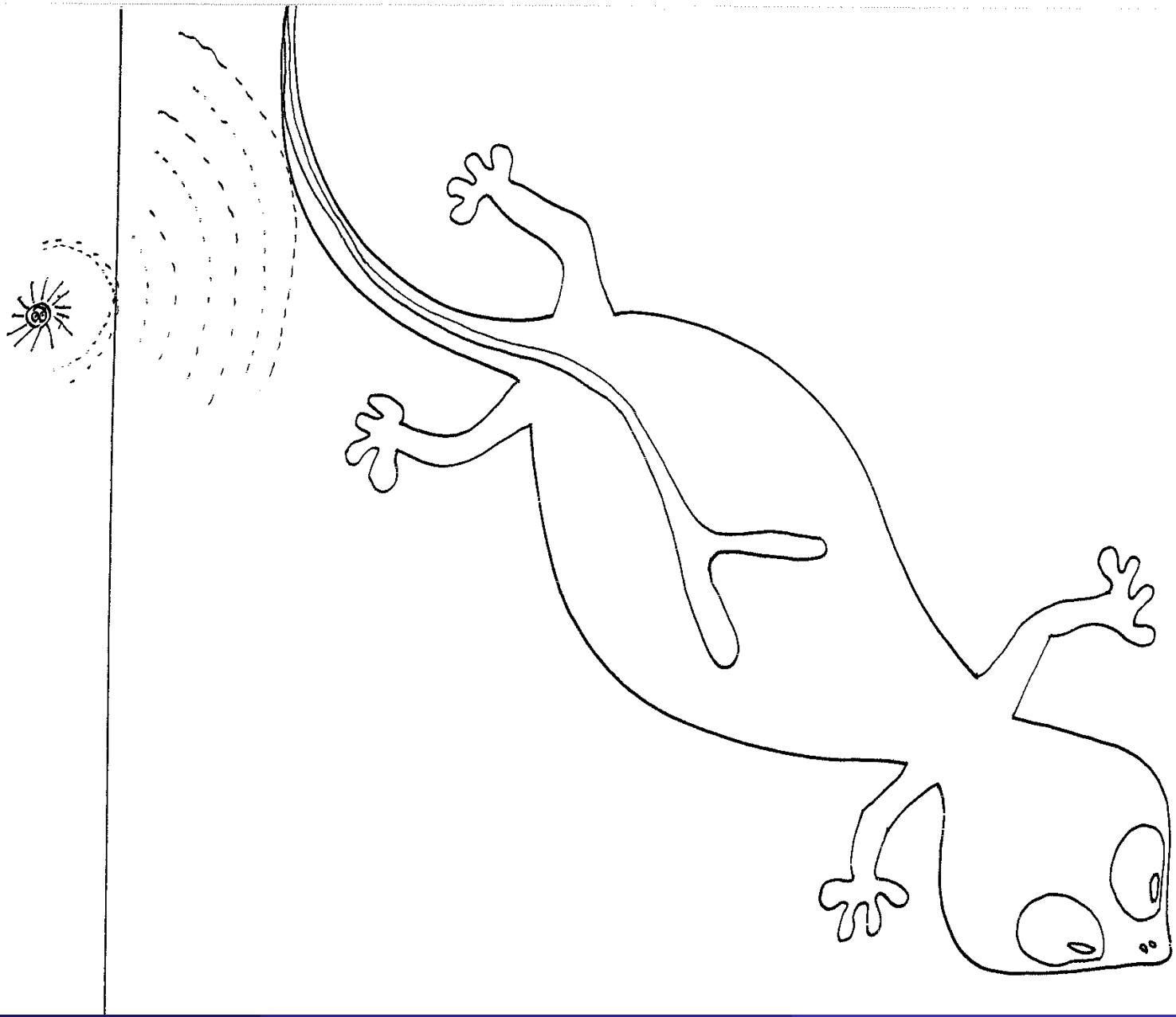
- Existence theorem
- Probabilistic version
- **Extrinsic complexity:**

Linear in $\mathbf{L}$

Polynomial in the Bézout number $\mathbf{d^n}$

We meet the best known extrinsic complexity bounds.

A non compact algebraic curve



The non compact case

We consider a larger parameter space, the product of

- a flag variety,
- a variety of hyperquadrics and
- a variety of hyperplanes.

The RADAR technology.

Again we arrive in the same complexity class!

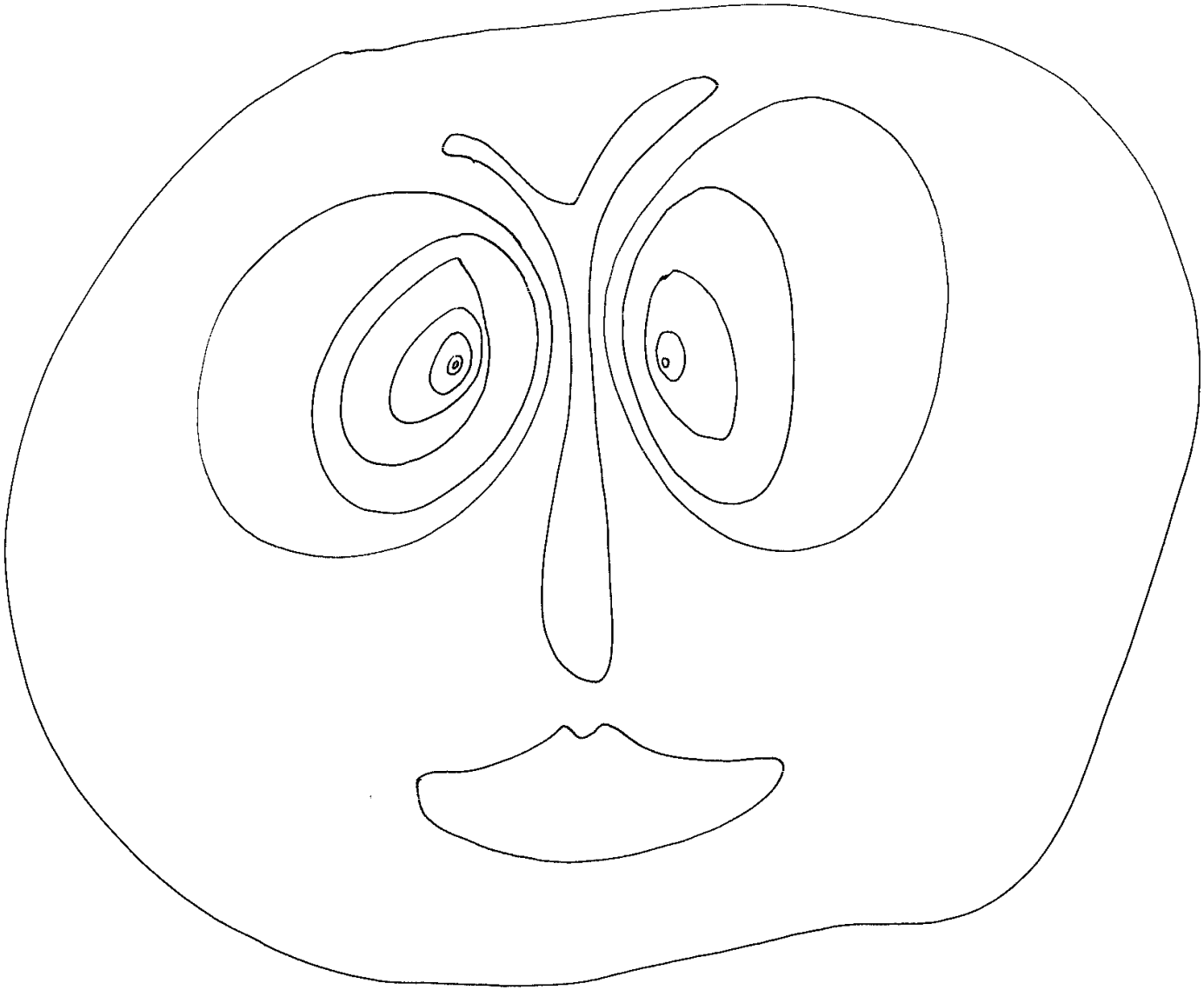
Complete intersection case; still smooth

- $f_1, \ldots, f_p$ a reduced regular sequence
- $\mathcal{V} := \{f_1 = \cdots = f_p = 0\}$

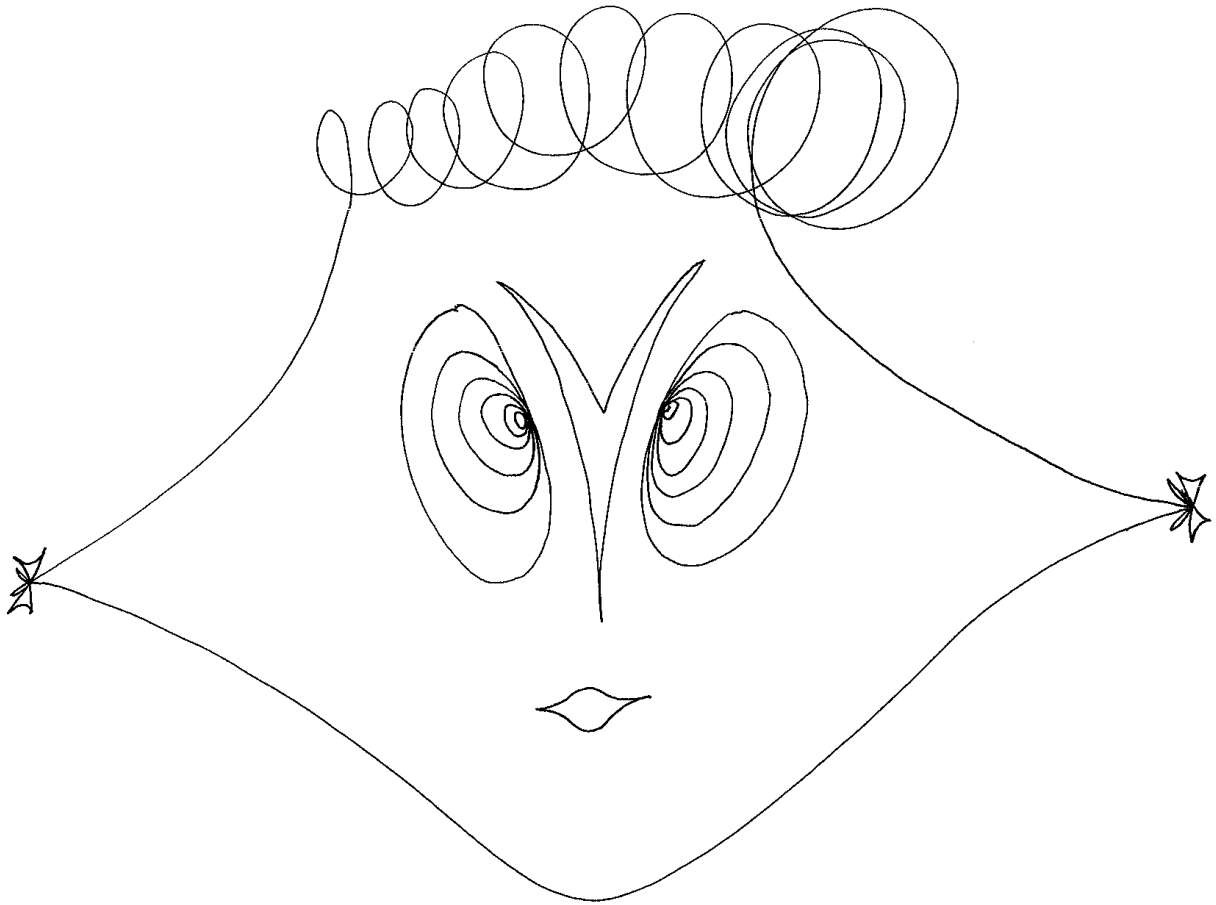
Up to a combinatorial factor we obtain the

same complexity class.

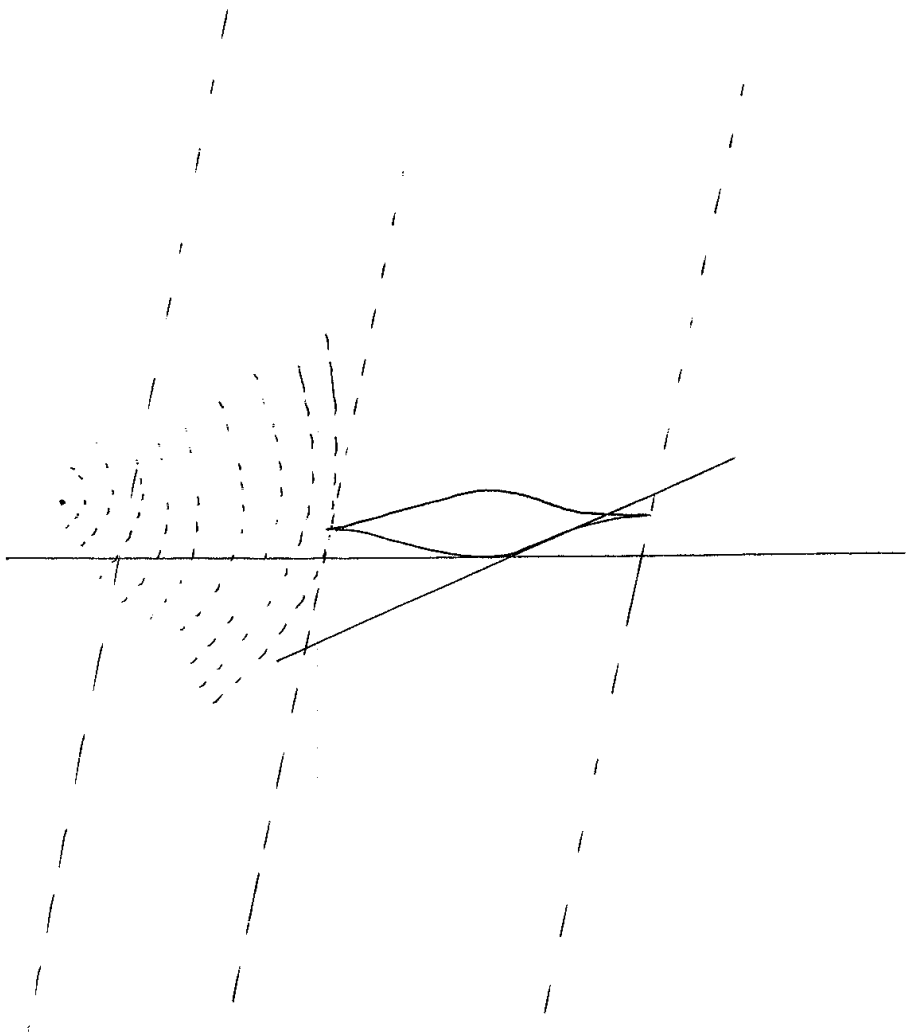
Again the old algebraic curve!



A new fresh one!!

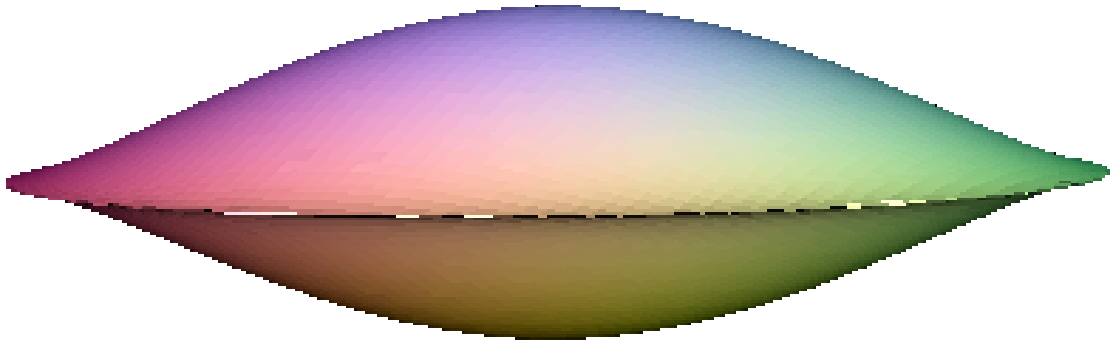


Thom's lips



The hat of the bishop of Jerusalem

$$f(x,y,z) := z^2 - \varepsilon^2(x^2+y^2-1)^3 = 0, \quad \varepsilon \quad \text{small}$$



The singular case

The (open) polar varieties

are not always non–empty.

The singular case

The (open) polar varieties

are not always non–empty.

The (closed) polar varieties

do not move anymore freely if the directions defined by A vary.
This is because they contain fixed parts (from the singular locus).

The singular case

The (open) polar varieties

are not always non–empty.

The (closed) polar varieties

do not move anymore freely if the directions defined by A vary.
This is because they contain fixed parts (from the singular locus).

Worse

We no longer have natural equations since ...

The singular case

The (open) polar varieties

are not always non–empty.

The (closed) polar varieties

do not move anymore freely if the directions defined by A vary.
This is because they contain fixed parts (from the singular locus).

Worse

We no longer have natural equations since . . .

. . . The determinantal description

defined above by the maximal minors is no longer valid since it contains all the singular locus.

LISSIFICATION AND STRATIFICATION VERSUS DESINGULARIZATION

Canonical desingularization of determinantal varieties – à la ROOM–KEMPF

- In the following let i be a fixed index between 1 and $n-1$. We consider the linear system

$$J(f)(x)^T\lambda + A^T\mu = 0$$

- with x in $\mathcal{V}$ and (λ, μ) in $\mathbb{A}^1 \times \mathbb{A}^{n-i}$.

LISSIFICATION AND STRATIFICATION VERSUS DESINGULARIZATION

Canonical desingularization of determinantal varieties – à la ROOM–KEMPF

- In the following let i be a fixed index between 1 and $n-1$. We consider the linear system

$$J(f)(x)^T\lambda + A^T\mu = 0$$

- with x in $\mathcal{V}$ and (λ, μ) in $\mathbb{A}^1 \times \mathbb{A}^{n-i}$.

Fundamental property: x singular implies $\mu = 0$

The singular case

The parameter space E_i

$$E = E_i := \{(A, \lambda, \mu) \in \mathbb{A}^{n(n-i)} \times \mathbb{A}^1 \times \mathbb{A}^{n-i} \,|\, rang\, A_i = n - i,\ \mu \neq 0\}$$

The singular case

The parameter space E_i

$$E = E_i := \{(A, \lambda, \mu) \in \mathbb{A}^{n(n-i)} \times \mathbb{A}^1 \times \mathbb{A}^{n-i} \,|\, rang\, A_i = n - i,\ \mu \neq 0\}$$

Group action

On E there is a natural right action of the group $G := Gl(n-i) \times Gl(1)$:
Let $g := (B, t) \in G$ and $e := (A, \lambda, \mu) \in E$

$$E \times G \to E, \qquad (e, g) \mapsto e \cdot g := (B^T \cdot A, t\lambda, tB^{-1} \cdot \mu)).$$

The singular case

The parameter space E_i

$$E = E_i := \{(A, \lambda, \mu) \in \mathbb{A}^{n(n-i)} \times \mathbb{A}^1 \times \mathbb{A}^{n-i} \,|\, rang\, A_i = n - i,\ \mu \neq 0\}$$

Group action

On E there is a natural right action of the group $G := Gl(n-i) \times Gl(1)$:
Let $g := (B, t) \in G$ and $e := (A, \lambda, \mu) \in E$

$$E \times G \to E, \qquad (e, g) \mapsto e \cdot g := (B^T \cdot A, t\lambda, tB^{-1} \cdot \mu)).$$

Quotient space

G induces on E_i an equivalence relation $\sim$.
We denote by $E_{/\sim}$ the set of G–orbits.

The singular case: small miracle

- The categorical quotient $E_i/\sim$ is a geometric quotient, i.e., is endowed with an algebraic variety structure – G is a " good " group (linearly reductive).

The singular case: small miracle

- The categorical quotient $E_i/\sim$ is a geometric quotient, i.e., is endowed with an algebraic variety structure – G is a " good " group (linearly reductive).
- This quotient is closed!
- Moreover it is a differentiable manifold.The points in a typical chart are of the form

$$A = \left[I_{n-i}\tilde{A}\right], \quad \mu = (1, \tilde{\mu})$$

where I_{n-i} is the identity matrix, $\tilde{A}$ a $(i \times (n-i))$–matrix, and $\tilde{\mu} := (\mu_2, \ldots, \mu_{n-i})$.

The singular case: medium miracle

- Let us consider for a fixed point x in $\mathcal{V}$ the subset defined by

$$\begin{array}{l} \dfrac{\partial f}{\partial x_1}(x)\lambda + \displaystyle\sum_{k=1}^{n-i} a_{k,1}\,\mu_k = 0, \\ \qquad\qquad \vdots \\ \dfrac{\partial f}{\partial x_n}(x)\lambda + \displaystyle\sum_{k=1}^{n-i} a_{k,n}\,\mu_k = 0, \end{array} \tag{1}$$

The singular case: medium miracle

- Let us consider for a fixed point x in $\mathcal{V}$ the subset defined by

$$
\begin{aligned}
\frac{\partial f}{\partial x_1}(x)\lambda + \sum_{k=1}^{n-i} a_{k,1}\,\mu_k &= 0,\\
&\vdots \qquad (1)\\
\frac{\partial f}{\partial x_n}(x)\lambda + \sum_{k=1}^{n-i} a_{k,n}\,\mu_k &= 0,
\end{aligned}
$$

- It is invariant under G.

The singular case: medium miracle

- Let us consider for a fixed point x in $\mathcal{V}$ the subset defined by

$$\begin{array}{l} \dfrac{\partial f}{\partial x_1}(x)\lambda + \displaystyle\sum_{k=1}^{n-i} a_{k,1}\,\mu_k = 0, \\ \qquad\qquad \vdots \\ \dfrac{\partial f}{\partial x_n}(x)\lambda + \displaystyle\sum_{k=1}^{n-i} a_{k,n}\,\mu_k = 0, \end{array} \tag{1}$$

- It is invariant under G.
- Therefore, we can consider the incident variety S_i of $\mathbb{A}^{n(n-i)} \times \mathbb{A}^n \times E_{i/\sim}$ defined by the same system (1) and of course the equation $f = 0$. It is closed!

The singular case: great miracle

The bipolar varieties

- The canonical projection

$$\mathbb{A}^n \times E_{i/\sim} \longrightarrow \mathbb{A}^n$$

maps S_i surjectively onto the open set of smoothness $\mathcal{V} \setminus Sing\mathcal{V}$ of smooth points (recall: $\mathcal{V} := \{x \in \mathbb{A}^n \,|f(x) = 0\}$).

The singular case: great miracle

The bipolar varieties

- The canonical projection

$$\mathbb{A}^n \times E_{i/\sim} \longrightarrow \mathbb{A}^n$$

maps S_i surjectively onto the open set of smoothness $\mathcal{V} \setminus Sing\mathcal{V}$ of smooth points (recall: $\mathcal{V} := \{x \in \mathbb{A}^n \,|f(x) = 0\}$).

- S_i is a smooth complete intersection variety.

The singular case: great miracle

The bipolar varieties

- The canonical projection

$$\mathbb{A}^n \times E_{i/\sim} \longrightarrow \mathbb{A}^n$$

  maps S_i surjectively onto the open set of smoothness $\mathcal{V} \setminus Sing\mathcal{V}$ of smooth points (recall: $\mathcal{V} := \{x \in \mathbb{A}^n \,|f(x) = 0\}$).
- S_i is a smooth complete intersection variety.
- S_i is **not** compact as soon as $Sing\mathcal{V} \neq \emptyset$.

The singular case: great miracle

The bipolar varieties

- The canonical projection

$$\mathbb{A}^n \times E_{i/\sim} \longrightarrow \mathbb{A}^n$$

  maps S_i surjectively onto the open set of smoothness $\mathcal{V} \setminus Sing\mathcal{V}$ of smooth points (recall: $\mathcal{V} := \{x \in \mathbb{A}^n \,|f(x) = 0\}$).
- S_i is a smooth complete intersection variety.
- S_i is **not** compact as soon as $Sing\mathcal{V} \neq \emptyset$.
- We can apply the technology developped previously for this case and consider of course its polar varieties (*generalized*). If sufficiently generic they form the bipolar varieties of de $\mathcal{V}$.
- The fibers $S_i(A)$ are (closed), generically smooth subvarieties dominating surjectively the (open) polar variety $\Delta_i(A)$.

Smoothness of the incident variety

On a typical chart S_i is defined by:

$$f(x) = 0,$$

$$\lambda \frac{\partial f}{\partial x_1}(x) + 1 = 0,$$

$$\lambda \frac{\partial f}{\partial x_l}(x) + \mu_l = 0, \qquad 2 \leq l \leq n - i$$

$$\lambda \frac{\partial f}{\partial x_l}(x) + a_{1,l} + \sum_{k=2}^{n-i} \mu_k \, \tilde{a}_{k,l} = 0, \qquad n - i + 1 \leq l \leq n.$$

At a non singular point of $\mathcal{V}$, analysing the corresponding jacobian matrix leads to:

- It is of full rank
- We can describe bipolar varieties with as many equations as its codimension (i.e. complete intersection)
- Applying the (weak) Thom's Transversality Theorem yields the generic smoothness of the fibers.

The bipolar lattice

- The bipolar varieties are organized by decreasing codimension in strictly ascending dimension:

$$B_{i,D_i} \subset \cdots \subset B_{i,j} \subset \cdots B_{i,1} \subset B_{i,0} = S_i$$

- If we let run also the index i from $n-1$ to 1, we obtain a tow–dimensional lattice of bipolar varieties.
- A walk in this lattice is a path of length at most n, starting from a zero–dimensional bipolar variety $B_{i_1,D_{i_1}}$ to end up at an orbit variety $B_{i_2,0} = S_{i_2}$
- At each step either the index i or the codimension j decreases.
- The bipolar varieties found under way along the walk, modulo suitable sections and indentifications, form an ascending sequence along which the dimension increases exactly by one. It is important to observe that their real traces are non empty, hence dense.

Algorithmic strategies

- Reversing a walk yields us an algorithmic strategy, which as soon as it finds regular points on bipolars hastens to project them on smooth real points of V
- As a bonus we obtain suitable choices of matrices A favourables to our aims
- There exists a particular walk boiling down to the previously known algorithms treating the smooth case.

The complexity theorem

Theorem.
Let $f(x_1, \ldots, x_n)$ a polynomial of degree $d \geq 2$ defining as above complex and real hypersurfaces V and $V_{\mathbb{R}}$. Suppose that f is given by a straight–line program of size L.

The complexity theorem

Theorem.
Let $f(x_1, \ldots, x_n)$ a polynomial of degree $d \geq 2$ defining as above complex and real hypersurfaces V and $V_{\mathbb{R}}$. Suppose that f is given by a straight–line program of size L.
Each walk $\mathcal{W}$ yields a procedure $\mathcal{R}_{\mathcal{W}}$ which exhibits at least one real algebraic point in each connected component of $V_{\mathbb{R}}$.

The complexity theorem

Theorem.
Let $f(x_1, \ldots, x_n)$ a polynomial of degree $d \geq 2$ defining as above complex and real hypersurfaces V and $V_{\mathbb{R}}$. Suppose that f is given by a straight–line program of size L.
Each walk $\mathcal{W}$ yields a procedure $\mathcal{R}_{\mathcal{W}}$ which exhibits at least one real algebraic point in each connected component of $V_{\mathbb{R}}$.
Its sequential complexity is linear in L and polynomial in d, n and an appropriate geometric quantity $\delta_{\mathcal{W}}$.

The complexity theorem

Theorem.
Let $f(x_1, \ldots, x_n)$ a polynomial of degree $d \geq 2$ defining as above complex and real hypersurfaces V and $V_{\mathbb{R}}$. Suppose that f is given by a straight–line program of size L.
Each walk $\mathcal{W}$ yields a procedure $\mathcal{R}_{\mathcal{W}}$ which exhibits at least one real algebraic point in each connected component of $V_{\mathbb{R}}$.
Its sequential complexity is linear in L and polynomial in d, n and an appropriate geometric quantity $\delta_{\mathcal{W}}$.
This quantity is the maximal degree of the (complex) bipolar varieties of V found along the walk. It is an intrinsic invariant of V and $\mathcal{W}$, which bounds also the number and the degree of the representative points exhibited by $\mathcal{R}_{\mathcal{W}}$.

Extrinsic Complexity for Finding One Point in Every Connected Component

It is enough to consider the last $n-1$–bipolar variety, of maximal codimension, i.e. zero-dimensional.
The system defining it satisfies:

- $d' = d$
- $n' \leq 2n$
- $\delta' \leq d^{2n}$

Conclusion:

THE SAME COMPLEXITY CLASS!

KRONECKER vs. GRÖBNER using MAGMA 2.13
Polynomial equations from the design of wavelet filters
PhD Thesis of Lutz Lehmann, advisor Bernd Bank

δ	δ^*	ktime	kmem	gtime	gmem
12	6	$1.5\,s$	$3\,MB$	$1.2\,s$	$1\,MB$
12	8	$7\,s$	$3\,MB$	$0.3\,s$	$6\,MB$
54	22	$1\,m\ 20\,s$	$7\,MB$	$3\,m\ 10s$	$38\,MB$
28	10	$16\,s$	$9\,MB$	$6.5s$	$9\,MB$
28	10	$60\,s$	$10\,MB$	$15\,s$	$10\,MB$
136	24	$30\,m$	$50\,MB$	$> 5\,h$	$> 800\,MB$
136	26	$1h\ 5\,m$	$75\,MB$	$> 5\,h$	$> 300\,MB$
32	6	$17\,s$	$7\,MB$	$2\,m\ 30\,s$	$17\,MB$
32	10	$45\,s$	$7\,MB$	$67\,s$	$21\,MB$
168	36	$1h\ 40\,m$	$98\,MB$	$> 5\,h$	$> 300\,MB$

δ: number of complex solutions
δ^*: number of real solutions
k ... : Kronecker (Lecerf, Lehmann)
g ... : Gröbner (Steel, F4 Faugère)

THANK YOU FOR YOUR ATTENTION!