Symbolic numeric methods for certifcated computation

B. Mourrain,

INRIA, BP 93, 06902 Sophia Antipolis
mourrain@sophia.inria.fr

January 8, 2003

Long time ago

$\mathbb{Z}$: Addition, substraction as geometric operations.

$\mathbb{Q}$: Ratio of numbers. Thalès and the pyramids (-620-546).

The multiplication is commutative (Pappus theorem).

!! *all is number* (commensurable) for the Pythagore's school (-585-400).

?? *No*, dare to say Hippase de Métaponte, *not* $\sqrt{2}$.

$\mathbb{R}$: *It's not a problem*, says Dedekind, *the missing numbers are those which are inbetween*.

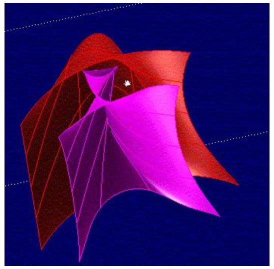
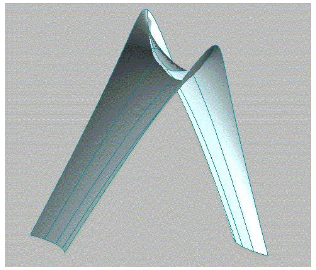
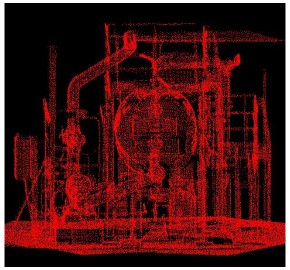
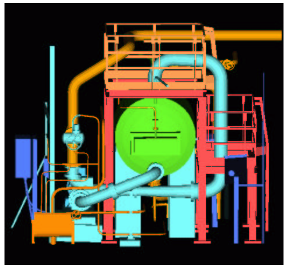
$\overline{\mathbb{Q}}$: *Yes, but* $\sqrt{2}$ *is special*, says Galois.

But today ?

Today, we are able to do 10^{10} floating point operations per second on a computer.

But dealing with algebraic numbers and solutions of polynomial equations is still critical in many situations.

In CAD

Constructions

- ☐ Intersection points of curves, surfaces.
- ☐ Approximation of intersection curves. Arrangements of patches.
- ☐ Offsets, filets, drafts, medial axis.
  $\Rightarrow$ **fast solvers, control of the error, refinement procedures.**

Detections

- ☐ Self-intersection, singularities, ray-tracing.
- ☐ Geometric approximation with simpler objects.
  $\Rightarrow$ **fast questers, isolation/sudvision/exclusion tools.**

Predicats

- ☐ Sorting points on a curve.
- ☐ Connectivity. Topological coherence.
- ☐ Geometric predicates on constructed points, curves, . . .

$\Rightarrow$ **fast tests (μs), filtering technics, datastructure for algebraic numbers.**

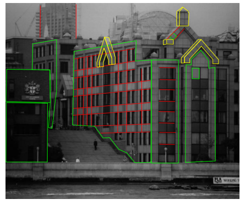
Representations

- ☐ Parametrised/implicit.
- ☐ Approximation of shape.
- ☐ Model reduction.

$\Rightarrow$ **global resultant methods, semi-algebraic geometry, multilevel, multiscale bases.**

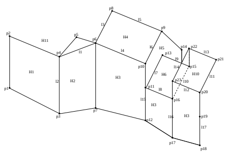
Modelisation from images

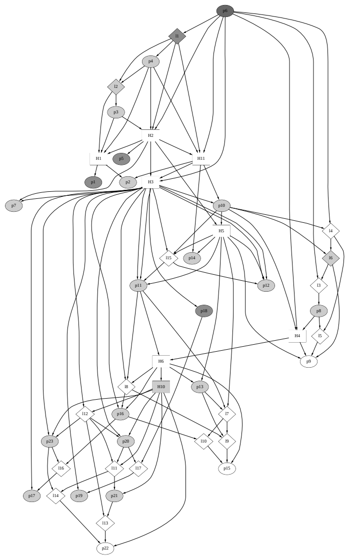
Extraction of points, lines, planes, . . .



Symbolic treatement of the geometric constraints.

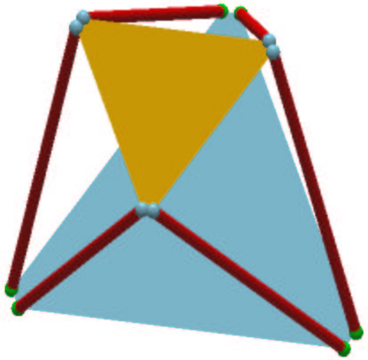
- **algebra**, rewritting, simplification.
- **proof**, **automatic** discovering of properties.

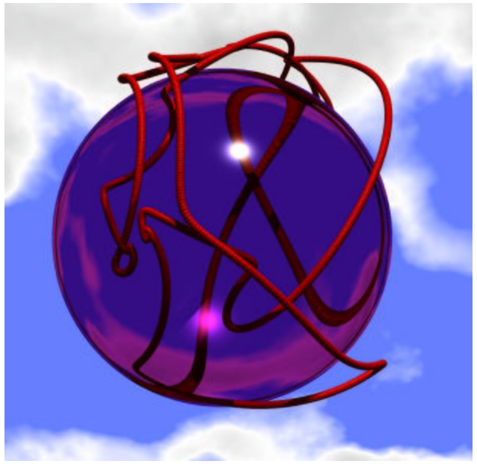




Numerical ajustment of the 3D model.

A robotic problem





Equations: $\|RY_i + T - X_i\|^2 - d_i^2 = 0, i = 1, \ldots, 6,$

$$R = \frac{1}{a^2+b^2+c^2+d^2}\begin{bmatrix} a^2-b^2-c^2+d^2 & 2\,ab-2\,cd & 2\,ac+2\,bd \\ 2\,ab+2\,cd & -a^2+b^2-c^2+d^2 & 2\,bc-2\,ad \\ 2\,ac-2\,bd & 2\,ad+2\,bc & -a^2-b^2+c^2+d^2 \end{bmatrix}, T = \begin{bmatrix} u/z \\ v/z \\ w/z \end{bmatrix}$$

Solutions: Generically 40 solutions: [RV93], [M94], [L93].

$$I_{\mathbb{P}^3\times\mathbb{P}^3} = \mathbf{P}_2^1 \cap \mathbf{Q}_2^8 \cap Q_1^{20} \cap \mathbf{Q}_0^{40} \cap \underbrace{Q_{-1,1}^{2\times 12} \cap Q_{1,-1}^{10} \cap Q_{-1}}_{\text{imbeddedcomponents}}$$

Solvers: fast and accurate; used intensively for several values of d_i and same geometry of the plateform; avoid singularities.

Solving

Equations with approximate coefficients

□ We consider the **neighbourhood** of a given system.

□ Family of systems depending on parameters, of the same "shape".

□ Around a **regular** value of the parameters,

- **continuity** of the solution set.
- **continuity** of the algebraic structure.

□ At a **singular** value of the parameters, all sort of bad things may happen.

How to proceed ?

- Analyse the class of systems that we have to solve.
- Apply tuned methods for generic systems of this class.

Solvers

- **Analytic solvers: exploit the value of f and its derivatives.**

Newton like methods, Minimisation methods, Weierstrass method.

- **Homotopic solvers: deform a system with known roots into the system to solve.**

Projective, toric, flat, deformation.

- **Subdivision solvers: use an exclusion criterion to isolate the roots.**

Taylor exclusion function, interval arithmetic, Descartes rule.

- **Algebraic solvers: exploit the known relation between the unkowns.**

Gröbner basis, normal form computations. Reduction to univariate or eigenvalue problems.

- **Geometric solvers: project the problem onto a smaller subspace.**

Resultant-based methods. Reduction to univariate or eigenvalue problems.

Subdivision

Subdvision solver

□ **Bernstein basis**: $f(x) = \sum_{i=0}^{d} b_i \, B_d^i(x)$, where $B_d^i(x) = \binom{d}{i} x^i (1-x)^{d-i}$.

$\mathbf{b} = [b_i]_{i=0,\ldots,d}$ are called the ***control coefficients***.

- $f(0) = b_0, f(1) = b_d$,
- $f'(x) = \sum_{i=0}^{d-1} \Delta(\mathbf{b})_i \, B_{d-1}^i(x)$ where $\Delta(\mathbf{b})_i = b_{i+1} - b_i$.

□ **Subdivision by De Casteljau algorithm:**

$$b_i^0 = b_i, \;\; i = 0, \ldots, d,$$
$$b_i^r(t) = (1-t)\, b_i^{r-1}(t) + t\, b_{i+1}^{r-1}(t), \;\; i = 0, \ldots, d-r.$$

- The control coefficients $\mathbf{b}^-(t) = (b_0^0(t), b_0^1(t), \ldots, b_0^d(t))$ and $\mathbf{b}^+(t) = (b_0^d(t), b_1^{d-1}(t), \ldots, b_d^0(t))$ describe f on $[0,t]$ and $[t,1]$.

- For $t = \frac{1}{2}$, $b_i^r = \frac{1}{2}(b_i^{r-1} + b_{i+1}^{r-1})$.; use of adapted arithmetic.
- Number of arithmetic operations bounded by $\mathcal{O}(d^2)$, memory space $\mathcal{O}(d)$. Indeed, asymptotic complexity $\mathcal{O}(d \log(d))$.

□ Isolation of real roots

Proposition: **(Descartes rule)** $\#\{f(x)=0; x\in[0,1]\} = V(\mathbf{b}) - 2p, p \in \mathbb{N}$.

Algorithm: isolation of the roots of f on the interval $[a,b]$

INPUT: *A representation $(\mathbf{b},[a,b])$ associate with f and ϵ.*
- *If $V(\mathbf{b})>1$ and $|b-a|>\epsilon$, subdivide;*
- *If $V(\mathbf{b})=0$, remove the interval.*
- *If $V(\mathbf{b})=1$, output interval containing one and only one root.*
- *If $|b-a|\le\epsilon$ and $V(\mathbf{b})>0$ output the interval and the multiplicity.*

OUTPUT: *list of isolating intervals in $[a,b]$ for the real roots of f or the ϵ-multiple root.*

- Multiple roots (and their multiplicity) computed within a precision ϵ.
- $x := t/(1-t)$: Uspensky method.
- Complexity: $\mathcal{O}(\frac{1}{2}d(d+1)\,r\,\left(\lceil\log_2\left(\frac{1+\sqrt{3}}{2s}\right)\rceil - \log_2(r)+4\right))$ [MVY02+]
- Natural extension to B-splines.

Benchmarks

Pentium III 933Mhz.

The number of equations per s. (C++ with 64-bit floats; $\epsilon = 0.000001$):

degree	8	9	12	16	18	20
	25 000	20-22 000	12-13 000	7.5-8 000	5.9-6.2 000	5.4 000

Equations per s. (precision bits vs. degree; $\epsilon = 0.000001$) using GMP library:

	16	20	30	40	60	80	100
128-bit	96	62.5	25.4	12.5	–	–	–
192-bit	83.3	53.2	21.5	10.8	4.0	–	–
256-bit	73.5	47.2	18.9	9.5	3.6	1.8	–
384-bit	60.2	37.7	15.2	7.6	2.9	1.4	0.8
512-bit	51	31.2	12.2	6.1	2.3	1.2	0.7

Compare favorably with other efficient solvers (Aberth method, `mpsolve`).

Algebraic method

The quotient algebra $\mathcal{A}$

- The polynomial ring $R = \mathbb{K}[x_1, \ldots, x_n]$.
- The equations $f_1 = 0, \ldots, f_m = 0$ to solve, with $f_i \in R$.
- The ideal $I = (f_1, \ldots, f_m) = \{\sum_i h_i f_i; h_i \in R\}$.
- The quotient algebra $\mathcal{A} = R/I$ of polynomials modulo I: $a \equiv a'$ iff $a - a' \in I$.

  (cf. polynomial functions on the set of solutions.)

- **How to represent and exploit effectively the structure of $\mathcal{A}$?**
  - ☐ **A basis for $\mathcal{A}$.**
  - ☐ **The multiplicative tables.**

Multiplication operators

We assume that $\mathcal{Z}(I) = \{\zeta_1, \ldots, \zeta_d\} \Leftrightarrow \mathcal{A}$ of finite dimension D over $\mathbb{K}$.

$$\begin{array}{rcl} M_a : \mathcal{A} & \rightarrow & \mathcal{A} \\ u & \mapsto & a\,u \end{array} \qquad \begin{array}{rcl} M_a^{\mathtt{t}} : \widehat{\mathcal{A}} & \rightarrow & \widehat{\mathcal{A}} \\ \Lambda & \mapsto & a \cdot \Lambda = \Lambda \circ M_a \end{array}$$

Theorem:

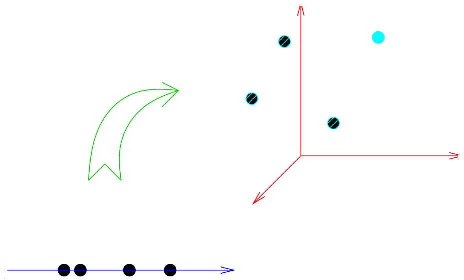
- **The eigenvalues of M_a are $\{a(\zeta_1), \ldots, a(\zeta_d)\}$.**
- **The eigenvectors of all $(M_a^{\mathtt{t}})_{a \in \mathcal{A}}$ are (up to a scalar) $\mathbf{1}_{\zeta_i} : p \mapsto p(\zeta_i)$.**

Theorem: **In a basis of $\mathcal{A}$, all the matrices M_a $(a \in \mathcal{A})$ are of the form**

$$\mathtt{M}_a = \begin{bmatrix} \mathtt{N}_a^1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathtt{N}_a^d \end{bmatrix} \text{ with } \mathtt{N}_a^i = \begin{bmatrix} a(\zeta_i) & & \star \\ & \ddots & \\ \mathbf{0} & & a(\zeta_i) \end{bmatrix}$$

Corollary: (Chow form)
$\Delta(\mathbf{u}) = \det(u_0 + u_1\,\mathtt{M}_{x_1} + \cdots + u_n\,\mathtt{M}_{x_n}) = \prod_{\zeta \in \mathcal{Z}(I)} (u_0 + u_1\zeta_1 + \cdots + u_n\zeta_n)^{\mu_\zeta}$.

Rational Univariate Representation of the roots



Algorithm: Rational Univariate Representation.

1. *Compute a* multiple *of the Chow form* $\Delta(\mathbf{u})$ *and its square free part* $d(\mathbf{u})$.
2. *Choose a generic* $t \in \mathbb{K}^{n+1}$ *and compute the first coefficients of*

$$d(t+u) = d_0(u_0) + u_1\, d_1(u_0) + \cdots + u_n\, d_n(u_0) + \cdots$$

3. *A non minimal rational univariate representation of the roots is given by* $\zeta_1 = \frac{d_1(u_0)}{d_0'(u_0)}, \ldots,$ $\zeta_n = \frac{d_n(u_0)}{d_0'(u_0)}$, $d_0(u_0) = 0$.
4. *Factorize* $d_0(u_0)$ *and keep the good factors for a minimal representation.*

Remark: t is generic iff $gcd(d_0(u_0), d_0'(u_0)) = 1$.

Normal form computation

Compute the projection of $\mathbb{K}[\mathbf{x}]$ onto a vector space B, modulo the ideal $I = (f_1, \ldots, f_m)$.

⇨ Grobner basis [CLO92, F99].

Compatibility with a monomial ordering but numerical instability.

⇨ Generalisation [M99, MT00, MT02].

No monomial ordering required. Linear algebra *with column pivoting* ; better numerical behavior of the basis.

Linear algebra on sparse matrices. Generic Sparse LU decomposition.

Application in Geometry

Solving by subdivision methods

Rectangular patches: $f(x,y) = \sum_{i=0}^{d_1} \sum_{j=0}^{d_2} b_{j,i} B_{d_1}^i(x) B_{d_2}^j(y)$ associated with the box $[0,1] \times [0,1]$.

• **Subdivision** by row or by column, similar to the univariate case.

• Arithmetic **complexity** of a subdivision bounded by $\mathcal{O}(d^3)$ ($d = max(d_1, d_2)$), memory space $\mathcal{O}(d^2)$.

Triangular patches: $f(x,y) = \sum_{i+j+k=d} b_{i,j,k} \frac{d!}{i!j!k!} x^i\, y^j\, (1-x-y)^k$ associated with the representation on the 2d **simplex**.

• Subdivision at a **new point**. Arithmetic complexity $\mathcal{O}(d^3)$, memory space $\mathcal{O}(d^2)$.

• Combined with **Delaunay triangulations**.

• Extension to A-patches.

Approximating an implicit curve

Algorithm: Representation of the implicit curve $f(x, y) = 0$

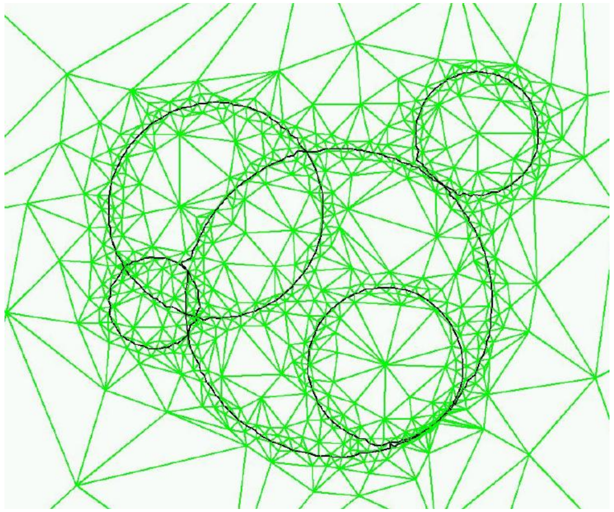
INPUT: *A triangular representation of f $L := ((A, B, C), \mathbf{b})$ and a precision ϵ.*
• *If at least one of the triangle edges is bigger that ϵ, split the triangle and insert the new triangles in L:*
- when the number of sign changes of some row (column or diagonal) is ≥ 2,
- or when the coefficients of f'_x (or f'_y, f'_z) have not the same sign.

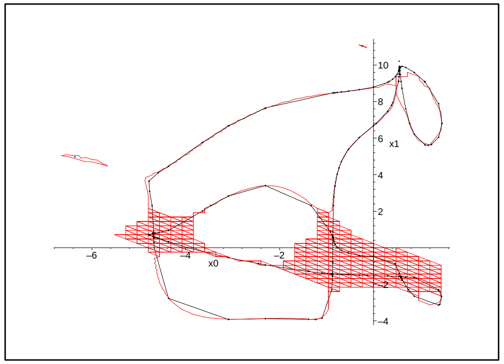
• *Remove the triangle from L if the coefficients of f have the same sign.*

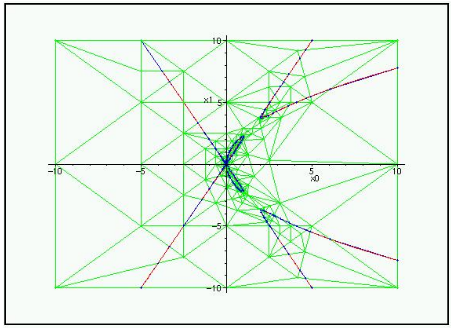
• *Save it*
- when all the edges of the triangle are smaller than ϵ,
- or when the total number of sign changes on the border sides is 2 and f'_x or f'_y, f'_z, has a constant sign. Isolate the roots.
OUTPUT: *A list of segments approximating the curve $f(x, y) = 0$.*

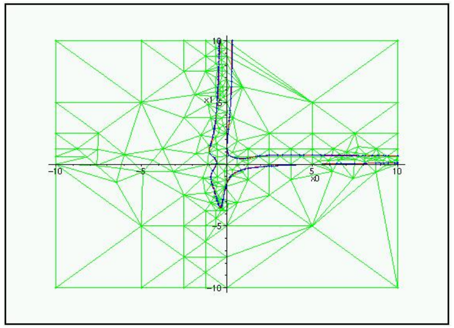
- Insertion of the circumcenter (barycenter), in order to break the *bad triangle*.
- No specific directions/axes used.
- New edges are constructed, no tangency problem.
- Number of triangles related to the complexe local feature size.
- Application to the intersection of curves, surfaces.

Examples

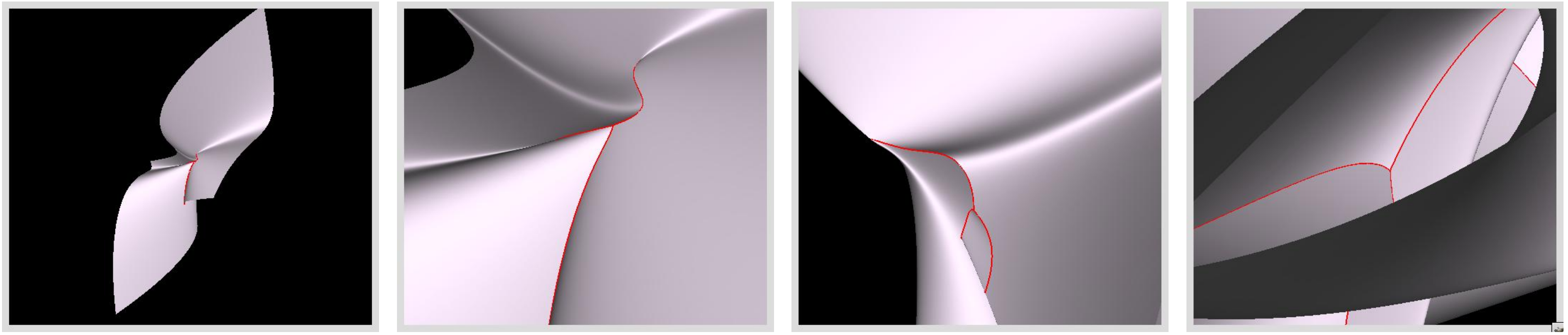


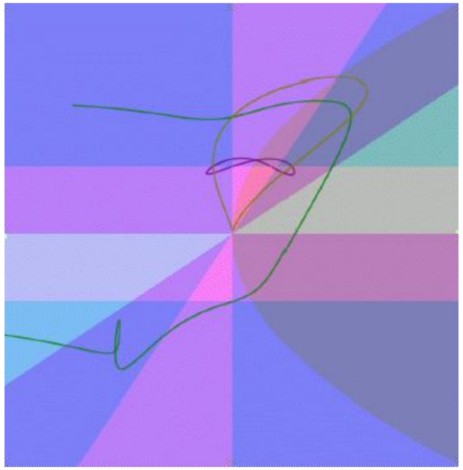






Self-intersection points

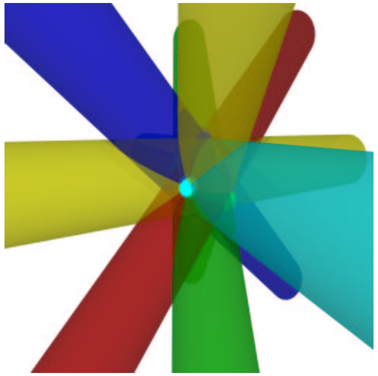




• Sample the surface.
• Segment it according to diff. information.
• Bound the regions with the same coding.
• Intersection the image of these regions by subdivision.

$(3,3)$	Sampling	Segmentation	Intersections
1000×1000	0.15 s	0.02s	0.5 s

Cylinders throught 4 and 5 points

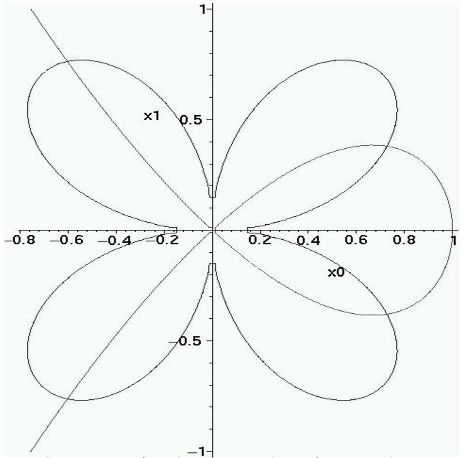


- Cylinders throught 4 points: curve of degree 3.
- Cylinders throught 5 points: $6 = 3 \times 3 - 3$.
- Cylinders throught 4 points and fixed radius: $12 = 3 \times 4$.
- Line tangent to 4 unit balls: 12.
- Cylinders throught 4 points and extremal radius: $18 = 3 \times 10 - 3 \times 4$.

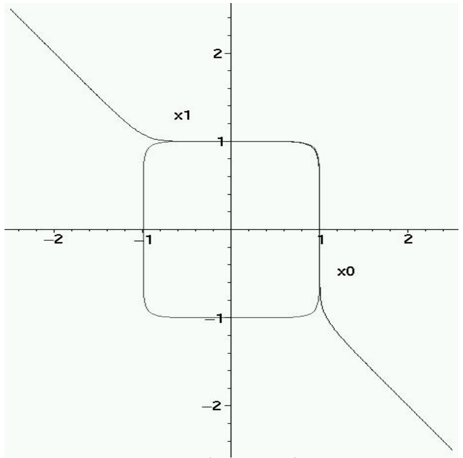
$Problem$	$time$	$max(\|f_i\|)$
Cylinders through 5 points	$0.03s$	$5 \cdot 10^{-9}$
Parallel cylinders through 2×4 points	$0.03s$	$5 \cdot 10^{-9}$
Cylinders through 4 points, extremal radius	$2.9s$	10^{-6}

Computations performed on an Intel PII 400 128 Mo of Ram

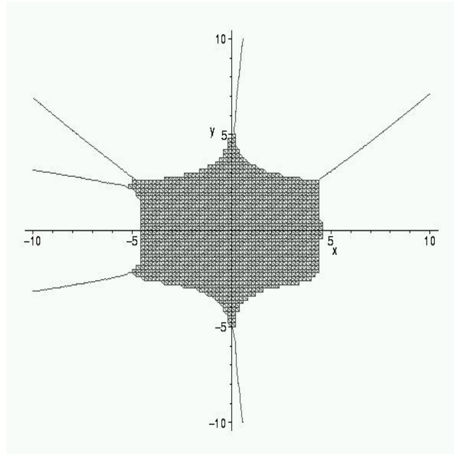
Comparison



$$f_1 = x^6 + 3x^4y^2 + 3x^2y^4 + y^6 - 4x^2y^2$$
$$f_2 = y^2 - x^2 + x^3$$



$$f_1 = x^9 + y^9 - 1$$
$$f_2 = x^{10} + y^{10} - 1$$



$$f = y^2 - 2y(x^{10} + 0.5x^9y^2 - \frac{1}{8}x^8y^4 + \frac{1}{16}x^7y^6 - \frac{5}{128}x^6y^8 + \frac{7}{256}x^5y^{10} - \frac{21}{1024}x^4y^{1}2 + \frac{23}{2048}x^3y^{1}4 - \frac{429}{32768}x^2y^{16} + \frac{715}{65536}xy^{18} - \frac{2431}{262144}y^{20}) + x^{20} + x^{19}y^2$$

- **Resultant in x_1**

Example	Degree of the variables			Evaluation	Time	Number of real roots
	Function	x_0	x_1			
10	f_1, f_2	6,3	6,2	10^{-9}	0.07	5
11	f_1, f_2	9,10	9,10	10^{-2}	0.63	2
15	f_1, f_2	6,4	6,4			4

- **Resultant + Univariate solving**

Example	Evaluation	Time	Number of real roots
10	10^{-16}	0.084	5
11	10^{-15}	3.489	2
15	10^{-9}	0.151	4

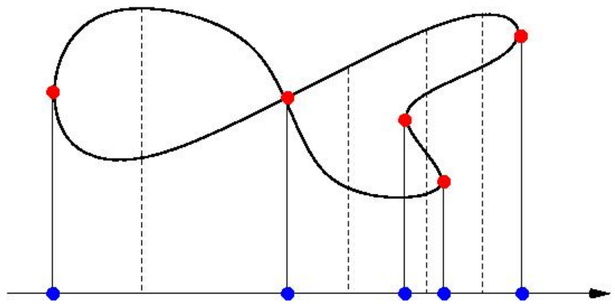
- **Subdvision**

Example	ε	Evaluation	Number of intervals	Time	Number of real roots
10	10^{-5}	10^{-5}	5	0.030	5
11	10^{-5}	10^{-4}	770	79.188	2
15	10^{-5}	10^{-4}	4	0.016	4

- **Normal form**

Example	Time / γ		Evaluation	Number of real roots
	$dinvlex$	mac		
10	0.01	0.01	10^{-6}	5
11	0.03	0.05	10^{-2}	2
15	0.02	0.01	10^{-6}	4

Curves



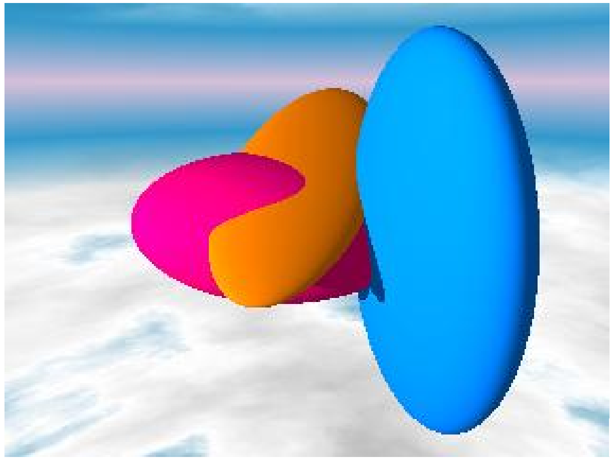
Algorithm: Topology of an implicit curve

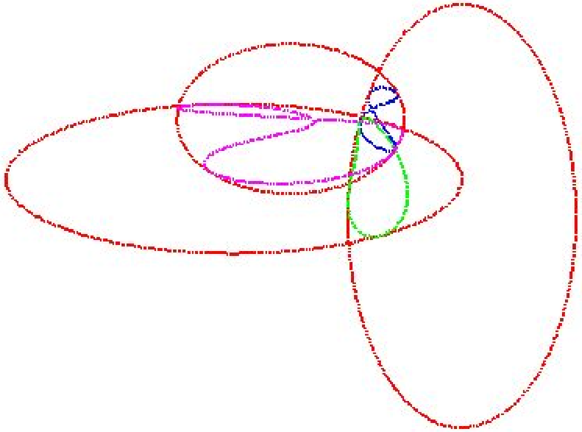
- *Compute the critical value for the projection along the y-abcisses.*
- *Above each point, compute the y-value, with their multiplicity.*
- *Between two critical points, compute the number of branches.*
- *Connect the points between two consecutive levels by y-order, the multi-branches beeing at* **the** *multiple point.*

⇨ Rationnal representation of the singular y in terms of the x.

⇨ Descartes rule to detect the multiple point among the regular ones.

Surfaces





Algorithm: Topology of an implicit surface

- *Project onto the plane.*
- *Compute the arrangement of the contour, singularity curves in the plane.*
- *Take a point inside each cell and compute the number of sheets above.*
- *Connect the regular sheets along the border of the contour, singular curves.*

⇨ Tangent curves in the projection.

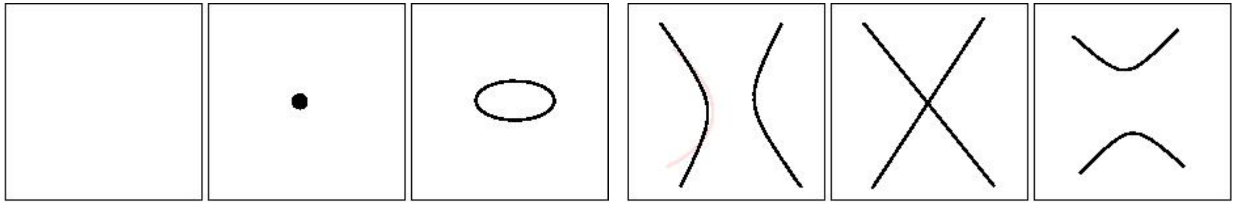
⇨ Degree, numeric problems inflated by projection.

See [Col75], [GHS01]

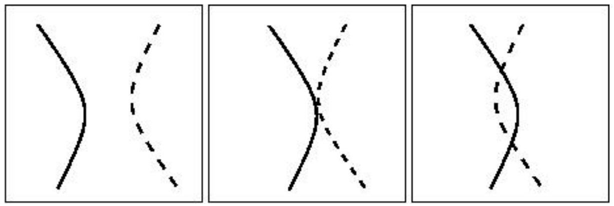
Arrangement of quadrics $Q_1, \ldots, Q_n$

Analyse the changes of topology of a section moving in the z-direction.

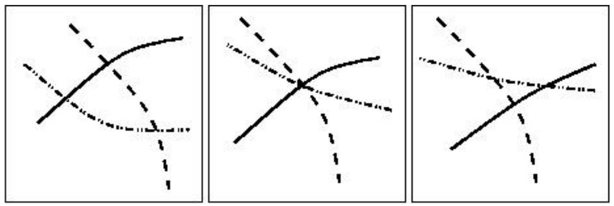
(a) $Q_{i_1} = 0, \partial_x(Q_{i_1}) = 0, \partial_y(Q_{i_1}) = 0$.



(b) $Q_{i_1} = 0, Q_{i_2} = 0, (\nabla Q_{i_1} \bigwedge \nabla Q_{i_2})_z = 0$.



(c) $Q_{i_1} = 0, Q_{i_2} = 0, Q_{i_3} = 0$.



Arrangement $=$ collection of cells of dimension 0,1,2,3 determined by sign conditions and adjacency relations.

Example: For a circle of equation $p(x, y) = x^2 + y^2 - 1$**,**

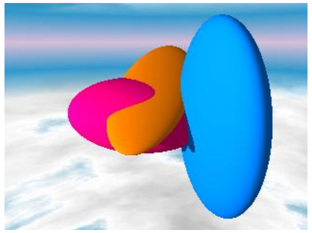
$$(\mathfrak{E}, p \geq 0), (C, p = 0), (\mathfrak{I}, p \leq 0) \;\; \& \;\; C \prec \mathfrak{I}, C \prec \mathfrak{E}.$$

Algorithm: Arrangement of quadrics

- *Compute the intersection points corresponding to the events (a), (b), (c).*
- *Sort them according to the z-abcissae, by increasing order.*
- *Compute the lowest arrangement of the conics.*
- *For each event, determine the cell to which the new critical point belongs and modify the arrangement of the neighbour cells accordingly.*
- *Connect in the different levels, the cells with the same sign conditions.*

⇨ Evaluation of sign conditions of rationnal quantities of z.

Example



$$272x^2 + 96xy + 192xz + 32y^2 + 64yz + 64z^2 - 571.2x - 142.4y - 252.8z + 323.64 = 0$$

$$128x^2 + 1152y^2 - 1024yz + 256z^2 - 144x - 886.4y + 358.4z + 220.12 = 0$$

$$64x^2 + 256y^2 + 128z^2 - 64x - 288y - 160z + 143 = 0$$

(a) 3×2 real solutions (0.01s):
(b) $3 \times 8 = 24$ complex solutions; 3×2 real (0.06s):
(c) 8 complex solutions; 2 real (0.02s):

```
(a) [0.825000,0.700000,0.287500]  C1
(a) [0.562500,0.544649,0.359835]  C1, C2
(a) [0.500000,0.562500,0.448223]  C1, C2, C3
(b) [0.498552,0.561349,0.448234]  C1, C2, C3, C23
(b) [0.687835,0.570199,0.508852]  C1, C2, C3, C23, C13
(b) [0.677133,0.617014,0.519616]  C1, C2, C3, C23, C13, C12
(c) [0.676862,0.612181,0.521687]  C1, C2, C3, C23, C13, C12, C123
(c) [0.638126,0.657542,0.685372]  C1, C2, C3, C23, C13, C12
(b) [0.534420,0.666721,0.719519]  C1, C2, C3,      C13, C12
(b) [0.662072,0.686211,0.723158]  C1, C2, C3,      C13
(b) [0.627783,0.558545,0.776837]  C1, C2, C3
(a) [0.500000,0.562500,0.801777]  C1, C2
(a) [0.562500,0.780351,0.890165]  C1
(a) [0.675000,0.300000,0.912500]
```

Dealing with algebraic numbers

represented

• by an **equation** and an **isolation interval**,

• reccursively, by equations with algebraic coefficients, and signs of polynomials at roots,

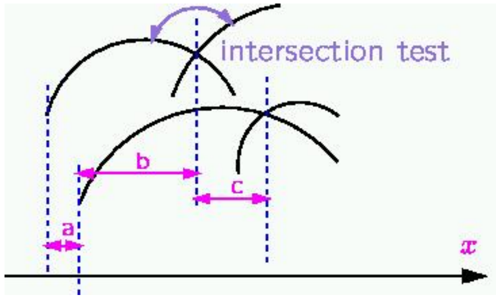
• by a **numerical approximation** and a way to **raffine** it

(evaluation of arithmetic DAG [LEDA, CORE], numerical procedure, . . .).

⇨ inequality test is certified.

⇨ equality requires **separation bound**.

Predicates on geometric objects



☐ Resultant formulations, in terms of a translation parameter.

☐ Sign of polynomials, of degree atmost 12.

☐ Filtering technics.

circle arcs

μs		first	polynomial	polynomial
		Interval + L_real	static +semi-static +Interval +GMP	static +semi-static +Interv. first +GMP
left right		2.48	0.36	0.36
		67	24.3	6.8
		2170	129	128