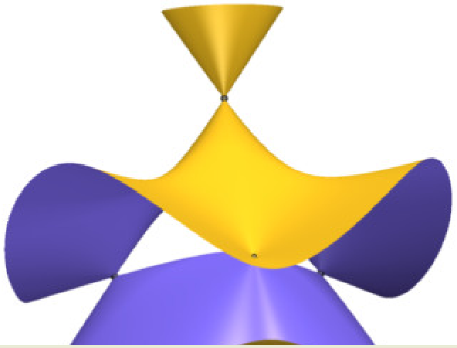
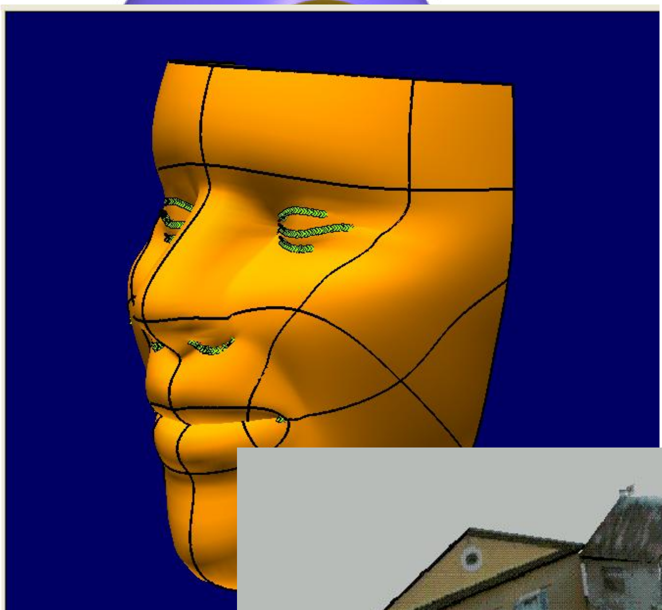
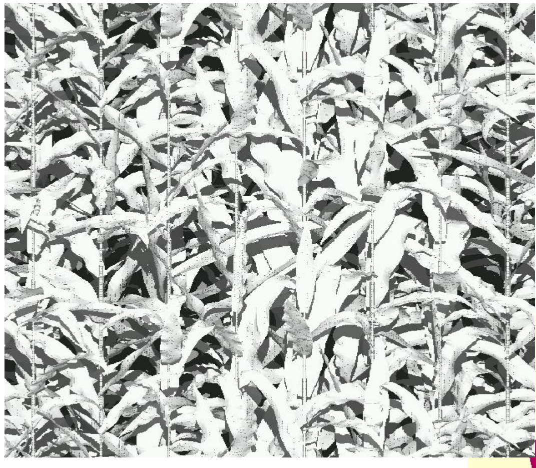
Solving Polynomial Equations in Geometric Problems

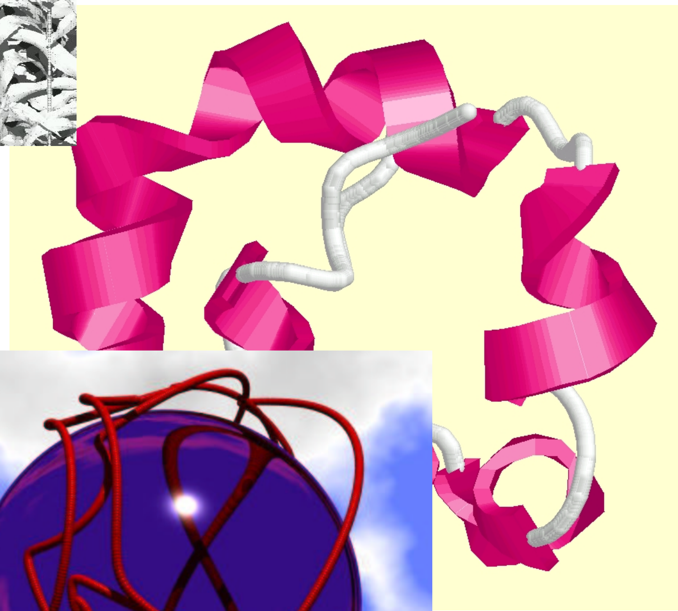
B. Mourrain

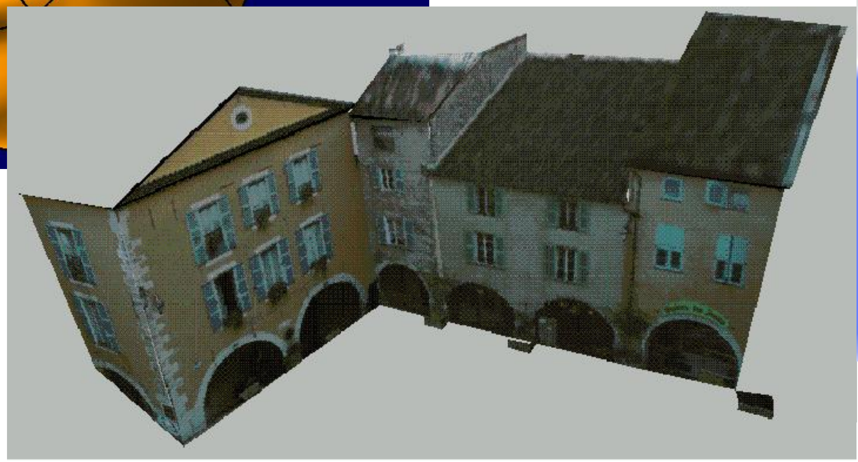
INRIA, BP 93, 06902 Sophia Antipolis
mourrain@sophia.inria.fr

December 9, 2004

SHAPES





- ☒ Semialgebraic models: Bezier parameterisation, NURBS, offset, draft, blending.
- ☒ *Initial* degree not high;
- ☒ Many algebraic patches;
- ☒ Coefficients known with incertainty: `double` type coefficients.
- ☒ Intensive use of algebraic tools;

Shape sampling

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 polynomial $f := (\mathbf{b},[a,b])$ with simple real roots 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], [MRR04]
- Natural extension to B-splines.

Ingredients

Theorem: $V(\mathbf{b}^-) + V(\mathbf{b}^+) \leq V(\mathbf{b})$.

Theorem: **(Vincent) If there is no complex root in the complex disc $D(\frac{1}{2}, \frac{1}{2})$ then**

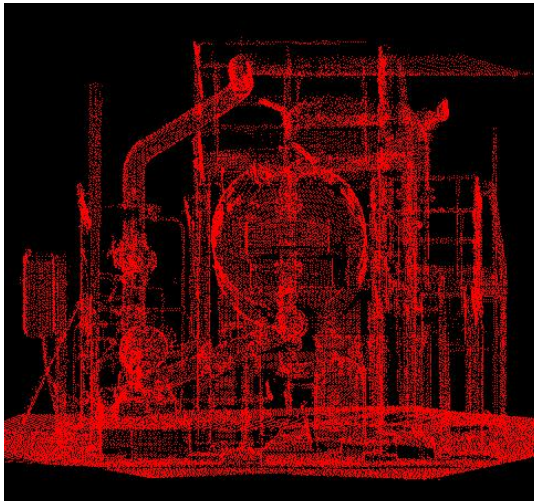
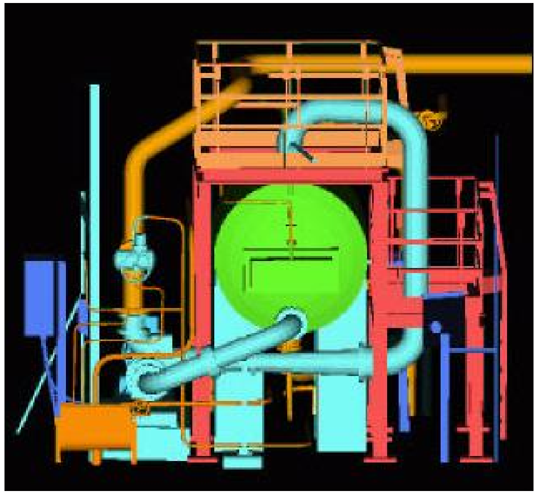
$$V(\mathbf{b}) = 0.$$

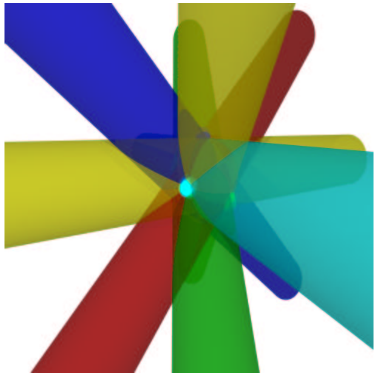
Theorem: **(Two circles) If there is no complex root in the union of the complex discs $D(\frac{1}{2} \pm \mathbf{i}\frac{1}{2\sqrt{3}}, \frac{1}{\sqrt{3}})$ except a simple real root, then**

$$V(\mathbf{b}) = 1.$$

Shape reconstruction

Reconstruction of cylinders



- 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: $18 = 3 \times 10 - 3 \times 4$.

Resultant-based method

☒ **Aim:** Project the problem onto a smaller (equivalent) one.

⇒ Algebraically speaking, deduce equations in the projection space

☒ **Means: resultant theory**.

⇒ Analysis of the geometry of the solution (**preprocessing**).

⇒ Use an adequate resultant formulation (**preprocessing**).

⇒ Construct a solveur implementing this formulation (**preprocessing**).

⇒ Instantiate the parameters and solve numerically (**at run-time**).

☒ **Projective resultant:** $\{\kappa_{i,j}(\mathbf{x})\} = \{\mathbf{x}^{\alpha_j};\ |\alpha_j| = d_i\}$. $X = \mathbb{P}^n$.

Sylvester-like matrix. Ratio of two Determinants. Determinant of the Koszul complex. [Mac1902], [J91].

☒ **Toric resultant:** $\{\kappa_{i,j}(\mathtt{t})\} = \{\mathtt{t}^{\alpha_j};\ \alpha_j \in A_i\}$, $\mathtt{t} \in (\mathbb{K} - \{0\})^n$, $X = \mathcal{T}_{A_0 \oplus \cdots \oplus A_n}$.

Polytope geomtry. Sylvester-like matrix. Maximal minors. Ratio of two Determinants [BKK75, GKZ91, PSCE93, DA01].

☒ **Resultant over a parameterised variety:** $\{\kappa_{i,j}(\mathtt{t})\}$ associated with the parametrisation of $X = \overline{\sigma(U)}$.

Bezoutian matrix. Maximal minors. A multiple of $\mathrm{Res}_X()$. [EM98, BEM00].

☒ **Residual resultant:** $\kappa_{i,j}(\mathbf{x}) \in (g_1(\mathbf{x}), \ldots, g_k(\mathbf{x}))$. X is the **blow-up** of $\mathbb{P}^n$ along $\mathcal{Z}(g_1, \ldots, g_k)$.

Explicit resolution of $(F : G)$. Gcd of the maximal minors. Degree formula. Ratio of determinants. [BKM75, BEM01, B01].

Shape structuring

Arrangement of surfaces

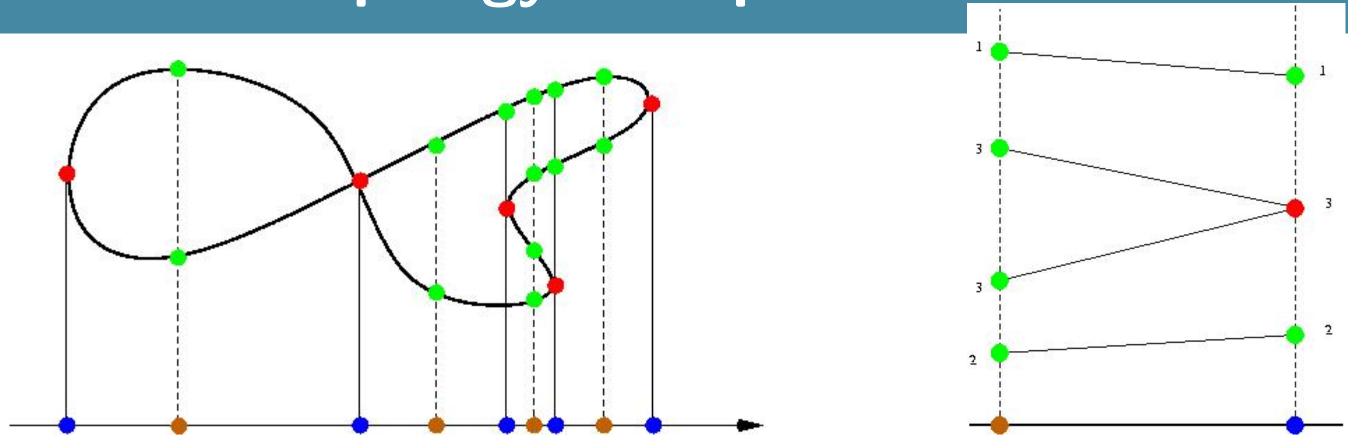
- ☒ **Constructions**
- ☒ Intersection points of curves, surfaces.
- ☒ Approximation of curves of intersection.
- ☒ Offsets, Median of curves, surfaces.

  $\Rightarrow$ **fast solveurs, control on the error, refinement procedures.**

- ☒ **Predicats**
- ☒ Sorting points on a curve.
- ☒ Connectivity. Topological coherence.
- ☒ Geometric predicats on the constructed points, curves, . . .

$\Rightarrow$ **fast tests (μs), filtering technics, polynomial formula/algebraic numbers. Algebraic manipulations, resultants.**

Topology of implicit curves



Algorithm: Topology of an implicit curve

1. *Compute the critical value for the projection along the y-abcisses.*
2. *Above each point, compute the y-value, with their multiplicity.*
3. *Between two critical points, compute the number of branches.*
4. *Connect the points between two slices according to their y-order.*

$\Rightarrow$ **Generic position:** atmost one critical point per vertical.

$\Rightarrow$ Sturm-Habicht sequence to express y in terms of the x.

$\Rightarrow$ Descartes rule to separate the multiple point from the regular ones.

$\Rightarrow$ Specialisation for union of simple primitives (critical and intersection points).

Topology of 3D curves

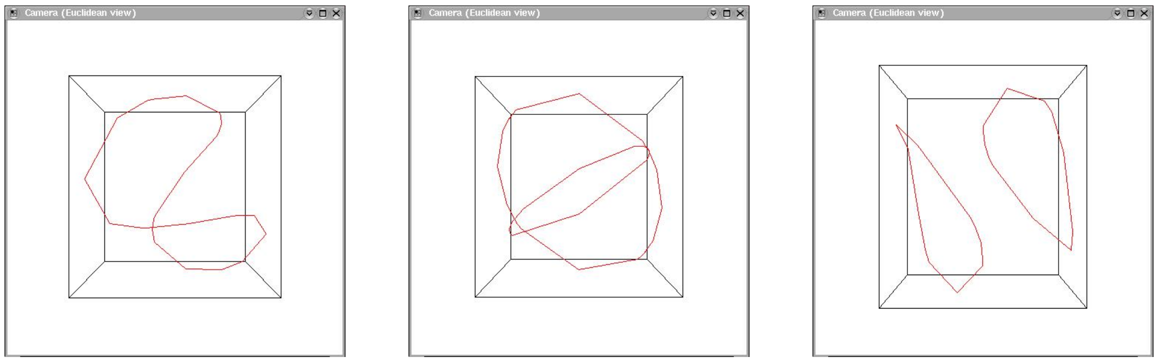
A curve $\mathcal{C} \subset \mathbb{R}^3$ defined by $P(x,y,z) = 0, Q(x,y,z) = 0$.

Algorithm: Topology of a 3D implicit curve

1. *Compute the x-critical points of $\mathcal{C}$.*
2. *Compute the singular points of $\pi_{x,y}(\mathcal{C})$ and $\pi_{x,z}(\mathcal{C})$.*
3. *Lift these points onto $\mathcal{C}$.*
4. *Inbetween two critical values, compute a regular section of $\mathcal{C}$.*
5. *Connect the points between two slices according to their (y,z)-order.*

$\Rightarrow$ **Generic position:**
$\forall \alpha \in \mathbb{R}, \quad \#\{\ (\alpha,\beta,\gamma)\ x\text{-critical}\ \} \leq 1$; no (x,y)-asymptotic direction.

$\Rightarrow$ **Ingredients:** resultants, univariate gcd, multivariate solver.



Meshing singular implicit surfaces

Input: $S = V(f(x,y,z) = 0)$ in a Box.
Output: A triangulation of S isotopic to S.

Algorithm: Triangulation of algebraic surfaces

1. *Compute a Whitney stratification $\mathcal{S}$ for S.*
2. *Deduce the sections where the topology changes so that between two sections, the surface is "topologically trivial".*
3. *Compute the topology of the sections.*
4. *Compute the topology of the apparent countour.*
5. *Use it to connect the sections together.*

Ingredients

- **Polar variety:** $\mathrm{VP}_z(S) = \{\mathbf{x} \in \mathbb{R}^3;\ f(\mathbf{x}) = 0; \partial_z(f)(\mathbf{x}) = 0\}$.
- The squarefree part $R(x, y)$ of $Resultant_z(f(x, y, z), \partial_z f(x, y, z))$.
- **A Whitney stratification of S:**

  $S_0 =$ points of S which projects to a x-critical of $V(R(x, y) = 0)$.

  $S_1 = \mathrm{VP}_z(S) - S_0$.

  $S_2 = S - S_1$.
- **Thom's lemma**:

Theorem: Let Z be a Whitney stratified subset of $\mathbb{R}^3$ and $f : Z \to \mathbb{R}^n$ be a proper stratified submersion. Then there is a stratum preserving homeomorphism

$$h : Z \to \mathbb{R}^n \times (f^{-1}(0) \cap Z)$$

which is smooth on each stratum and commutes with the projection to $\mathbb{R}^n$.

Algebraic numbers

☒ **Representation:**

☒ an arithmetic tree ($\sqrt{x+y+2\sqrt{x\,y}} - \sqrt{x} - \sqrt{y}$), and/or

☒ a (irreducible) **polynomial** $p(x) = 0$ **and an isolating interval**.

☒ **Construction:**

⇒ Isolation via Descartes, Uspenksy, de Casteljau, Sturm(-Habicht) algorithm.

☒ **Predicates:**

⇒ Comparison of two numbers by refinement until a separating bound:

$$\alpha \neq 0 \Rightarrow |\alpha| > B(Symbolic\ Expression\ of\ \alpha).$$

⇒ Queries such as comparision, sign determination via Sturm(-Habicht) method.

Sturm method

- Univariate polynomials $A(x), B(x)$ of degree d_1, d_2
- Sturm sequence $R_0 := A, R_1 := B$, $R_{i+1} = -rem(R_{i-1}, R_i) \ldots R_N$.
- $V_{A,B}(a) :=$ number of sign variation of $[R_0(a), R_1(a), \ldots R_N(a)]$.

Theorem: $V_{A,A'B}(a) - V_{A,A'B}(b)$ **is the number of real roots of** A **such that** $B > 0$ **- the number of real roots of** A **such that** $B < 0$ **on the interval** $]a, b[$**.**

- Application to sign determination of polynomials at the root of A on an isolating interval.
- Precomputation for fixed degree.
- Habicht variant based on sign of minors of the Sylvester matrix. Control of the coefficient size.

Algebraic solvers

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

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

☒ **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**

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

Algorithm: Solving a zero-dimensionnal multivariate system.

1. *Compute the table of multiplication by $x_i, i = 1, \ldots, n$.*
2. *Compute the eigenvectors of the tranposed matrices $M_{x_i}^t$.*
3. *Deduce the coordinates of the roots from the eigenvectors.*

Normal form computation

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

$\Rightarrow$ Grobner basis [CLO92, F99].

Compatibility with a monomial ordering but numerical instability.

$\Rightarrow$ 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.

- Examples with kastura(n), modular arithmetic:

n	mac	random	dlex
6	0.17s	0.28s	0.58s
7	0.95s	5.07s	4.66s
10	256.81s	7590.85s	635s
11	1412s	∞	4591.43s

- Katsura(6), and floating point arithmetic :

choice function	number of bits	time	$max(\|\|f_i\|\|_\infty)$
dlex	128	1.48s	10^{-28}
dinvlex	128	4.35s	10^{-24}
mac	128	1s	10^{-30}
dinvlex	80	3.98s	10^{-15}
mac	80	0.95s	10^{-19}
dlex	80	1.35s	10^{-20}
dlex	64		$-$
dinvlex	64		$-$
mac	64	0.9s	10^{-11}

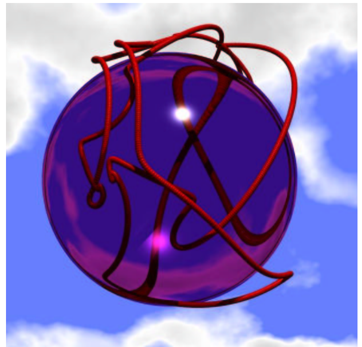
- Parallel robot, approximate coefficients.

choice function	number of bits	time	$max(\|\|f_i\|\|_\infty)$
dlex	250	11.16s	$0.42 * 10^{-63}$
mac	250	11.62s	$0.46 * 10^{-63}$
dinvlex	250	13.8s	$0.135 * 10^{-60}$
dlex	128	9.13s	$0.3 * 10^{-24}$
dinvlex	128	11.1s	$0.3 * 10^{-23}$
mac	128	9.80s	$0.1 * 10^{-24}$
dlex	80	-	-
dinvlex	80	-	-
mac	80	6.80s	10^{-12}

- Parallel robot, rational coefficients.

	mac	minsz	dlex	mix
size	18M	30M	50M	45M

The 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: [RV92], [L93], [M93], [M94], [FL95], . . .

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

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

Direct modelisation		Quaternions		Redundant	
250 b. $3.21s$	128 b. -	250 b. $8.46s$	128 b. $6,25s$	250 b. $1.5s$	128 b. $1.2s$.

experimentation by Ph. trebuchet

Shape interrogation

Multivariate Bernstein representation

☒ **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.

Multivariate subdivision solver

$$\begin{cases} f_1(\mathbf{u}) = \sum_{i_1,\ldots,i_n} b^1_{i_1,\ldots,i_n} B^{d_1,\ldots,d_n}_{i_1,\ldots,i_n}(u_1,\ldots,u_n), \\ \vdots \\ f_s(\mathbf{u}) = \sum_{i_1,\ldots,i_n} b^s_{i_1,\ldots,i_n} B^{d_1,\ldots,d_n}_{i_1,\ldots,i_n}(u_1,\ldots,u_n), \end{cases}$$

☒ **Algorithm**

1. preconditioning on the equations;
2. reduction of the domain;
3. if the reduction ratio is too small, subdivision of the domain.

Preconditioning (for square systems)

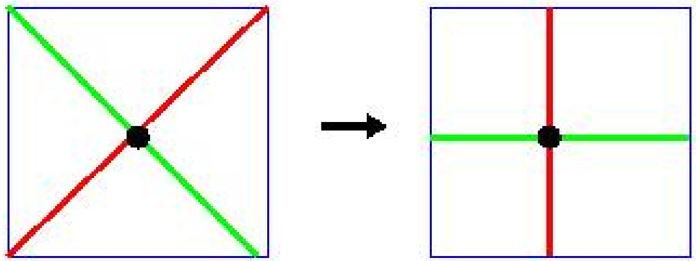
Transform $\mathbf{f}$ into $\tilde{\mathbf{f}} = M\,\mathbf{f}$

a) Optimize the distance between the equations:

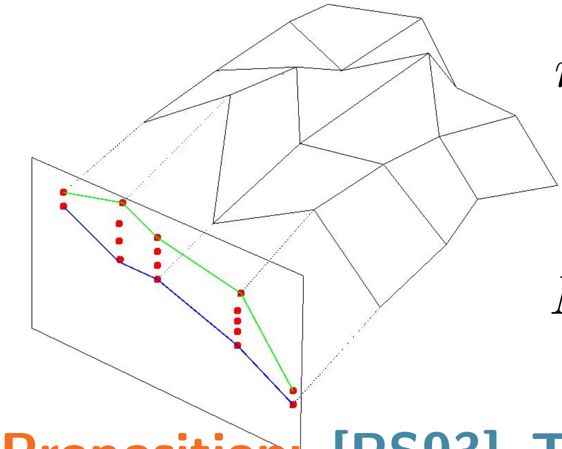
$$||f||^2 = \sum_{0\le i_1\le d_1,\ldots,0\le i_n\le d_n} |\mathbf{b}(f)_{i_1,\ldots,i_n}|^2,$$

by taking for M, the matrix of eigenvectors of $Q = (\langle f_i|f_j\rangle)_{1\le i,j\le s}$.

b) $M = J_{\mathbf{f}}^{-1}(\mathbf{u}_0)$ for $\mathbf{u}_0 \in \mathcal{D}$.



Reduction



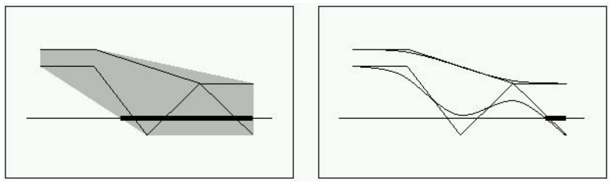
$$m_j(f;x_j) = \sum_{i_j=0}^{d_j} \min_{\{0\le i_k\le d_k, k\neq j\}} b_{i_1,\ldots,i_n} B_{d_j}^{i_j}(x_j;a_j,b_j)$$

$$M_j(f;x_j) = \sum_{i_j=0}^{d_j} \max_{\{0\le i_k\le d_k, k\neq j\}} b_{i_1,\ldots,i_n} B_{d_j}^{i_j}(x_j;a_j,b_j).$$

Proposition: [PS93] The intersection of the convex hull of the control polygon with the axis contains the projection of the zeroes of $\mathbf{f}(\mathbf{u}) = 0$.

Proposition: For any $\mathbf{u} = (u_1,\ldots,u_n) \in \mathcal{D}$, and any $j = 1,\ldots,n$, we have

$$m_j(f;u_j) \le f(\mathbf{u}) \le M_j(f;u_j).$$



Use the roots of $m_j(f,u_j) = 0$, $M_j(f,u_j) = 0$ to reduce the domain of search.

Theorem: (Multivariate Vincent theorem) If $f(\mathbf{x})$ has no root in the complex polydisc $D(1/2, 1/2)^n$, then the coefficients of f in the Bernstein basis of $[0, 1]^n$ are of the same sign.

- **Quadratic convergence for the control polygon:**

Theorem: There exists $\kappa_2(f)$ such that for $\mathcal{D}$ of size ϵ small enought,

$$\forall \mathbf{x} \in \mathcal{D};\ |f(\mathbf{x}) - \mathbf{b}(f; \mathbf{x})| \leq \kappa_2(f)\, \epsilon^2.$$

- **Quadratic convergence for the reduction:** preconditioner (b).

Proposition: Let $\mathcal{D}$ a domain of size ϵ containning a simple root of f . There exists $\kappa_{\mathrm{f}} > 0$, such that for ϵ small enought

$$|\tilde{M}_j(\tilde{\mathbf{f}}; u_j) - \tilde{m}_j(\tilde{\mathbf{f}}; u_j)| \leq \kappa_{\mathbf{f}} \epsilon^2.$$

- **Guarantee:** adapt the arithmetic rounding mode during the reduction.

Experiments

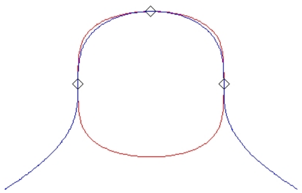
sbd subdivision.

rd reduction, based on a univariate root-solver using the Descarte's rule.

sbds subdivision using the preconditioner (a).

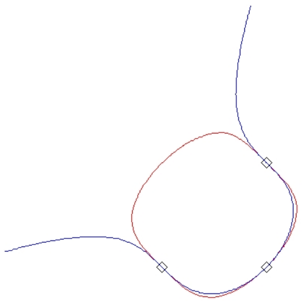
rds reduction using the global preconditioner (a).

rdl reduction using the jacobian preconditioner (b).



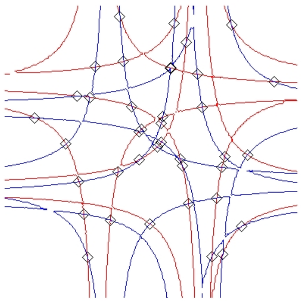
method	iterations	subdivisions	output	time (ms)
sbd	161447	161447	61678	1493
rd	731	383	36	18
sbds	137445	137445	53686	1888
rds	389	202	18	21
rdl	75	34	8	7

bidegrees (2,3), (3,4); 3 singular solutions.



method	iterations	subdivisions	output	time (ms)
sbd	235077	235077	98250	4349
rd	275988	166139	89990	8596
sbds	1524	1524	114	36
rds	590	367	20	29
rdl	307	94	14	18

bidegrees (3,4), (3,4); 3 singular solutions.



method	iterations	subdivisions	resultat	time (ms)
sbd	4826	4826	220	217
rd	2071	1437	128	114
sbds	3286	3286	152	180
rds	1113	748	88	117
rdl	389	116	78	44

bidegree (12,12), (12,12)

Tools

☒ **Synaps**:

- A library for symbolic and numeric computations.
- Data structures: **vectors, matrices (dense, Toeplitz, Hankel, sparse, . . .), univariate polynomials, multivariate polynomials.**
- Algorithm: different types of solvers, resultants. . .
- GPL+runtime exception, `cvs@cvs-sop.inria.fr`.
- `http://www-sop.inria.fr/galaad/logiciels/synaps/`

☒ **Axel**

- Algebraic Software-Components for gEometric modeLing;
- C++; gcc 3.*; configure; autoconf; cvs server; doxygen
- Data structures: points, point graph, parameterised and implicit curves and surfaces, quadrics, bezier, bspline . . .
- Algorithms: intersection, topology, meshing . . .
- `http://www-sophia.inria.fr/logiciels/axel/`

☒ Mathemagix

- Typed computer algebra interpreter.
- Hight level programming langage.
- Automatic tools for building external dynamic modules (play-plug-play).
- `ftp://ftp.mathemagix.org/pub/mathemagix/targz/`

☒ Texmacs

- High quality mathematical editor
- Import/export latex, html, xml
- Interface to computer algebra systems.