

An introduction to algebraic methods for solving polynomial equations

B. Mourrain,



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

8th October 2001

Arrangement of surfaces

Constructions

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

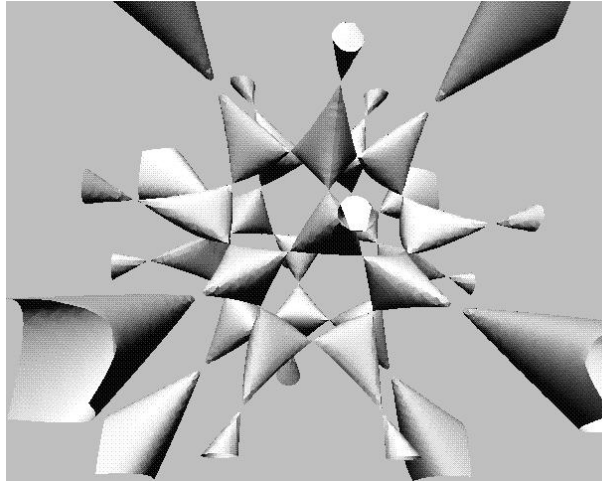
⇒ fast solveurs, control on the error, refinement procedures.

Predicats

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

⇒ fast tests (μs), filtering technics, polynomial formula/algebraic numbers. Algebraic manipulations, resultants.

Drawing implicit surfaces

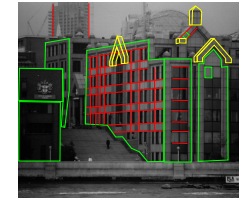


$$(8c + 4)x^2y^2z^2 - c^4(x^4y^2 + y^4z^2 + x^2z^4) + c^2(x^2y^4 + y^2z^4 + x^4z^2) - \frac{2c+1}{4}(x^2 + y^2 + z^2 - 1)^2 = 0, c = \frac{1+\sqrt{5}}{2}$$

Modelisation from images

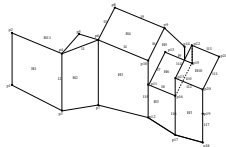


Extraction of points, lines, planes, . . .

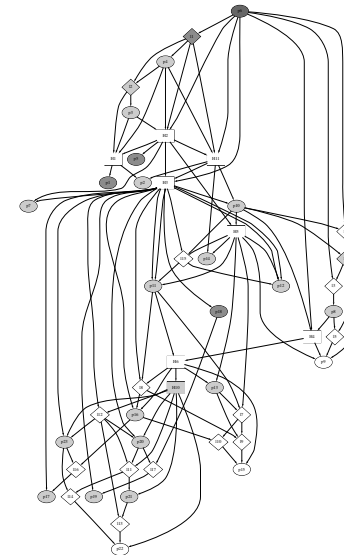


Symbolic treatment of the geometric constraints.

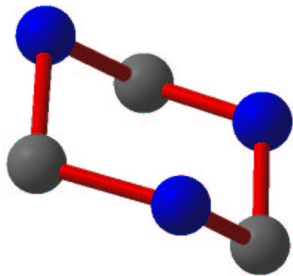
- **algebra**, rewriting, simplification.
- **proof, automatic** discovering of properties.



Numerical adjustment of the 3D model.



The cyclohexan

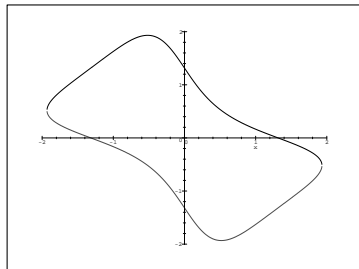


$$f_1 = -\frac{\sqrt{3}}{2} + \frac{1}{2}t_2^2 + \frac{1}{2}t_3^2 + 2t_2t_3 + \frac{\sqrt{3}}{2}t_2^2t_3^2 = 0$$

$$f_2 = -\frac{\sqrt{3}}{2} + \frac{1}{2}t_1^2 + \frac{1}{2}t_3^2 + 2t_1t_3 + \frac{\sqrt{3}}{2}t_1^2t_3^2 = 0$$

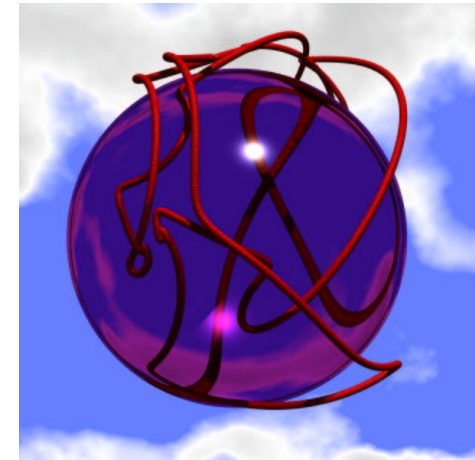
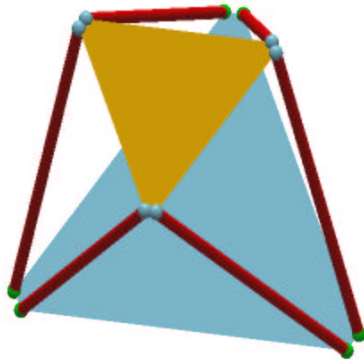
$$f_3 = -\frac{\sqrt{3}}{2} + \frac{1}{2}t_1^2 + \frac{1}{2}t_2^2 + 2t_1t_2 + \frac{\sqrt{3}}{2}t_1^2t_2^2 = 0$$

A curve of degree 4 + 2 isolated points + 6 embedded points



t_1	t_2	t_3
0.5176444559	0.5176444559	0.5176444563
-0.5176444567	-0.5176444567	-0.5176444555
0.5176444559	-1.931851652	0.5176444563
-0.5176444567	1.931851652	-0.5176444555
1.931851652	-0.5176444567	-0.5176444555
-1.931851652	0.5176444559	0.5176444563
0.5176444561	0.5176444561	-1.931851652
-0.5176444561	-0.5176444561	1.931851652

A robotic problem



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

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

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

Fast and accurate solveurs: $\simeq 1s$, error 10^{-6} .

Solving polynomial equations

Solving polynomial equations

The context:

- The equations $f_1 = \dots = f_m = 0$, $f_i \in R = \mathbb{C}[x_1, \dots, x_n]$.
- The solutions $\zeta \in \mathbb{C}^n$ st. $f_1(\zeta) = \dots = f_m(\zeta) = 0$.

The problems:

Count/compute an approximation of all the (real) solutions (in a box).

The objectives:

- Numerical stability.
- Certification et control.
- Efficiency via structure.

Solvers

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

Newton like methods, Minimisation methods, Weierstrass method.

- **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 unknowns.

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

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

Projective, toric, flat, deformation.

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

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

Analytic methods

Subdivision methods

A Univariate polynomial solver

- Usual representation of a $f(x)$ of degree d : $f(x) = \sum_{i=0}^d a_i x^i$.
 - Bernstein basis: $f(x) = \sum_{i=0}^d b_i B_d^i(x)$, where $B_d^i(x) = \binom{d}{i} x^i (1-x)^{d-i}$.
- $\mathbf{b} = [b_i]_{i=0,\dots,d}$ are called the *control coefficients*.

Properties:

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

Proposition: The number of sign changes $V(\mathbf{b})$ of $\mathbf{b} = [b_i]_{i=0,\dots,d}$ bounds the number of real roots of f on $[0, 1]$ and is equal to it modulo 2.

If $V(\mathbf{b}) = 0$, the number of real root on $[0, 1]$ is 0, and

If $V(\mathbf{b}) = 1$, the number of real root on $[0, 1]$ is 1.

De Casteljau subdivision algorithm

$$b_i^0 = b_i, \quad i = 0, \dots, d,$$

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

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

A polynomial = a sequence of coefficients + an interval $[a, b]$

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

Isolation algorithm

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

INPUT: A representation $(\mathbf{b}, [a, b])$ associate with f .

- *Compute the number of sign changes $V(\mathbf{b})$.*
- *If $V(\mathbf{b}) > 1$ and its size is greater than ϵ , subdivide the representation into two subrepresentations \mathbf{b}^- , \mathbf{b}^+ , corresponding to the two halves of the input interval and apply recursively the algorithm to them.*
- *If $V(\mathbf{b}) = 0$, remove the interval.*
- *If $V(\mathbf{b}) = 1$, the interval contains one root, that can be isolated within the precision ϵ .*

OUTPUT: list of subintervals of $[a, b]$ containing exactly one real root of f .

- Isolation and approximation of the roots on an interval.
- Multiple roots (and their multiplicity) computed within a precision ϵ
- All the real roots of $f(x)$, with $x := t/(1 - t)$, $t \in [0, 1[$ and $x = -x$.
- With this change of variable, also called Uspensky algorithm.
- Natural extension to B-splines.

Theorem: [MVY01] Assume that $f(x) = 0$ has simple roots.

- 1. An upper bound of the number of recursion steps of the isolation algorithm is**

$$l = \lceil \log_2 \left(\frac{5d}{2s} \right) \rceil.$$

where d is the degree of f and $s = \min_{\{x_i \neq x_j \text{ roots of } f\}} |x_i - x_j|$.

- 2. An upper bound of the number of arithmetic operations of the procedure isolation is**

$$v = \frac{1}{2}d(d+1)r \left(\lceil \log_2 \left(\frac{5d}{2s} \right) \rceil - \log_2(r) + 4 \right),$$

where r is the number of sign changes of the sequence $\mathbf{b} = (b_k)_{0,\dots,d}$.

Extension to higher dimension

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 $[0, 1] \times [0, 1]$.

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.

Properties of rectangular patches

- We associate to f the **matrix of control coefficients** $\mathbf{b} = (b_{i,j})_{0 \leq i \leq d_1, 0 \leq j \leq d_2}$ and **the box** $[0, 1] \times [0, 1]$.

- The subsequences

$$\left\{ \begin{array}{l} \mathbf{b}^S := (b_{0,0}, \dots, b_{0,d_1}) \\ \mathbf{b}^N := (b_{d_2,0}, \dots, b_{d_2,d_1}) \\ \mathbf{b}^E := (b_{0,0}, \dots, b_{d_2,0}) \\ \mathbf{b}^W := (b_{0,d_1}, \dots, b_{d_2,d_1}) \end{array} \right.$$

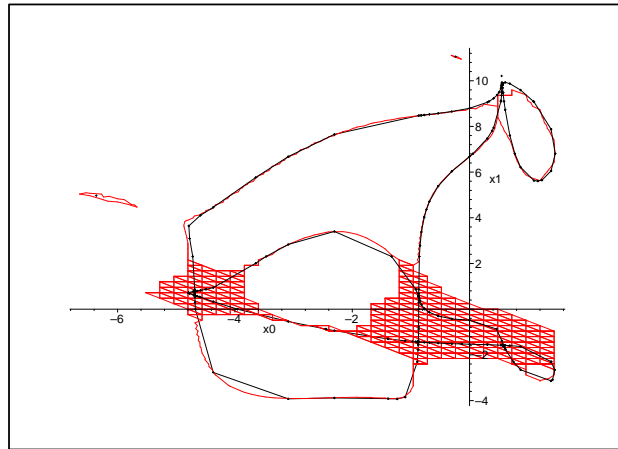
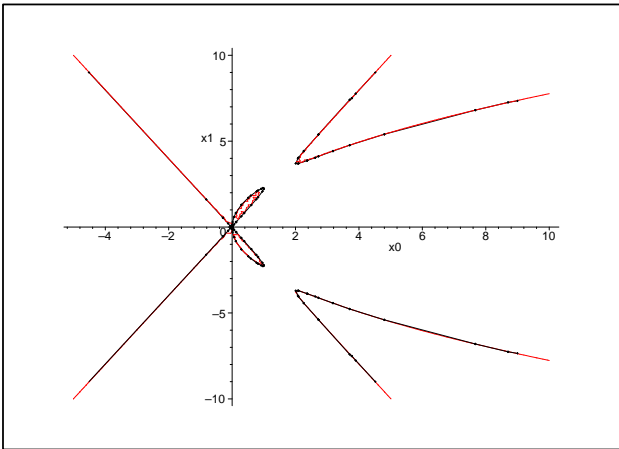
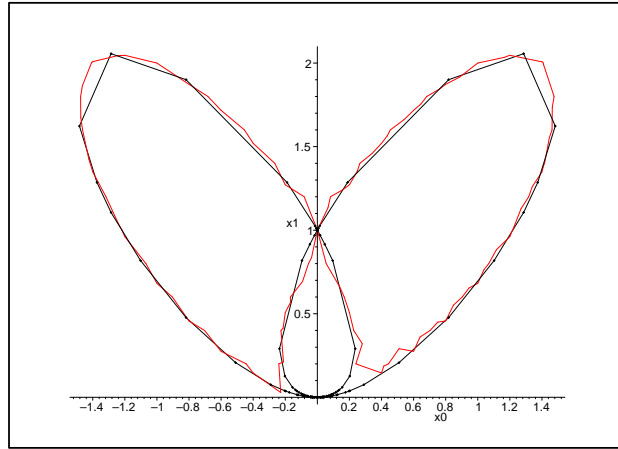
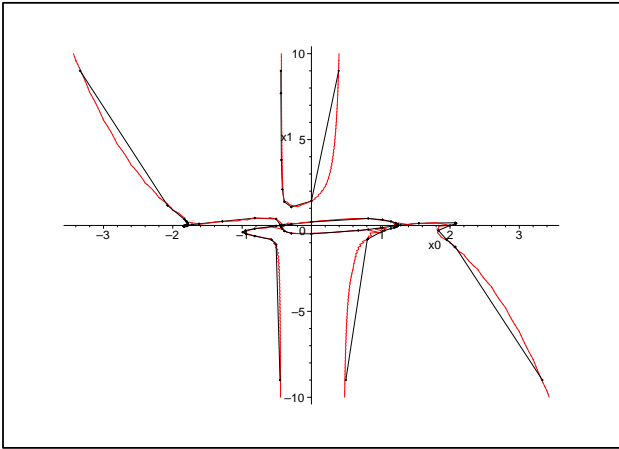
represents the function $f(x, 0)$ (resp. $f(x, 1)$, $f(0, y)$, $f(1, y)$) on $[0, 1]$.

- **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 $\mathcal{O}(d^2)$.

Applications (in progress)

- Compute an **approximation** of the curve (resp. surface) within the precision ϵ , with good **geometric/numerical properties**.
- Computation of the topological degree.
- Isolation of complex roots, resp. real roots of several functions.
- Intersection of curves, surfaces.
- Arrangement of curves, surface patches.

(still in progress)



Algebraic methods

Ingredients

□ Relations/equations/**constraints** satisfied by the **unknowns**.

□ Quotient algebra. Computation modulo the constraints.

Do as if the unknowns are known

□ Compute the roots, by reducing to **linear algebra problems**

Duality, eigenvalue computation.

The quotient algebra \mathcal{A}

- The polynomial ring $R = \mathbb{K}[x_1, \dots, x_n]$.
- The ideal $I = (f_1, \dots, f_m) = \{\sum_i h_i f_i; h_i \in R\}$.
- 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.

The quotient algebra

Let I be the ideal of $R = \mathbb{K}[x_1, x_2]$ generated by

> $f_1 := 13*x[1]^2 + 8*x[1]*x[2] + 4*x[2]^2 - 8*x[1] - 8*x[2] + 2$:

> $f_2 := x[1]^2 + x[1]*x[2] - x[1] - 1/6$:

The quotient ring $\mathcal{A} = \mathbb{K}[x_1, x_2]/I$ is a vector space of dimension 4. A basis of \mathcal{A} is $1, x_1, x_2, x_1x_2$. We have

> $\text{expand}(x[1]*x[1] - f_2)$;

$$x_1^2 \equiv -x_1x_2 + x_1 + \frac{1}{6}.$$

> $\text{expand}(x[1]*(x[1]*x[2]) + 1/9*x[1]*f_1 - (5/9 + 13/9*x[1] + 4/9*x[2])*f_2)$;

$$x_1^2x_2 \equiv -x_1x_2 + \frac{55}{54}x_1 + \frac{2}{27}x_2 + \frac{5}{54}.$$

More generally, any polynomial in $\mathbb{K}[x_1, x_2]$ can be reduced, modulo the polynomials f_1, f_2 , to a linear combination of the monomials $1, x_1, x_2, x_1x_2$.

The univariate case

Consider $f = f_d x^d + \cdots + f_1 x + f_0$ ($f_d \neq 0$), with simple roots ζ_1, \dots, ζ_d .

- The **idempotents** are the **Lagrange interpolation polynomials** :

$$\mathbf{e}_i(x) = \prod_{j=1, j \neq i}^d \frac{x - \zeta_j}{\zeta_i - \zeta_j}.$$

$$\square \mathbf{e}_i^2 \equiv \mathbf{e}_i, \mathbf{e}_i \mathbf{e}_j \equiv 0, i \neq j, \mathbf{e}_1 + \cdots + \mathbf{e}_n \equiv 1.$$

$$\square \mathcal{A} = \mathbb{C}[x]/(f) = \mathbb{K} \mathbf{e}_1 \oplus \cdots \oplus \mathbb{K} \mathbf{e}_d; \quad \forall a \in \mathcal{A}, a \equiv a(\zeta_1) \mathbf{e}_1 + \cdots + a(\zeta_d) \mathbf{e}_d.$$

$$\square \text{The dual basis of } (\mathbf{e}_i) \text{ is } (\mathbf{1}_{\zeta_i}).$$

\square Another basis of \mathcal{A} is $(1, x, \dots, x^{d-1})$. The matrix of **multiplication** by x in this basis is

$$\mathbf{M}_x = \begin{pmatrix} 0 & \cdots & 0 & -\frac{f_0}{f_d} \\ 1 & \cdots & \vdots & \vdots \\ & \cdots & 0 & \vdots \\ 0 & & 1 & -\frac{f_{d-1}}{f_d} \end{pmatrix}.$$

The structure of $\mathcal{A} = R/I$

Theorem: \mathcal{A} finite dimensional vector space iff $\mathcal{Z}_{\mathbb{C}}(I) = \{\zeta_1, \dots, \zeta_d\}$ is finite.

$$\mathcal{A} = \mathbb{C}[x_1, \dots, x_n]/I = \mathcal{A}_1 \oplus \dots \oplus \mathcal{A}_d,$$

with

- $\mathcal{A}_i = e_i \mathcal{A}$,
- $e_i^2 = e_i$, $e_i e_j = 0$ if $i \neq j$,
- $e_1 + \dots + e_d = 1$.

- Primary decomposition of : $I = Q_1 \cap \dots \cap Q_d$ where Q_i is \mathfrak{m}_{ζ_i} -primary.
- $\mathcal{A}_i \sim R/Q_i$.

Duality

- **The dual de R** is $\widehat{R} = \{ \Lambda : R \rightarrow \mathbb{K}, \text{ linear} \}$.

- Let $(\mathbf{d}^\alpha)_{\alpha \in \mathbb{N}^n}$ be the dual basis of $(\mathbf{x}^\alpha)_{\alpha \in \mathbb{N}^n}$.

$$\Lambda = \sum_{\alpha \in \mathbb{N}^n} \Lambda(\mathbf{x}^\alpha) \mathbf{d}^\alpha \in \mathbb{K}[[\mathbf{d}_1, \dots, \mathbf{d}_n]].$$

Example: $\mathbf{1}_\zeta : p \mapsto p(\zeta)$ is of the form $\mathbf{1}_\alpha = \sum_{\alpha \in \mathbb{N}^n} \zeta^\alpha \mathbf{d}^\alpha = \frac{1}{\prod_{i=1}^n (1 - \zeta_i \mathbf{d}_i)}$.

- **The R -module structure:** $\forall a \in R, \forall \Lambda \in \widehat{R}, a \cdot \Lambda : b \mapsto a \cdot \Lambda(b) = \Lambda(ab)$.

Example: $x_1 \cdot \mathbf{d}_1^{\alpha_1} \mathbf{d}_1^{\alpha_2} \dots \mathbf{d}_1^{\alpha_n} = \mathbf{d}_1^{\alpha_1 - 1} \mathbf{d}_1^{\alpha_2} \dots \mathbf{d}_1^{\alpha_n}$ if $\alpha_1 > 0$ and 0 otherwise.

- **The dual of $\mathcal{A} = R/I$** is $\widehat{\mathcal{A}} = \{ \Lambda : R \rightarrow \mathbb{K} \text{ st. } \Lambda(I) = 0 \} = I^\perp$.

Example:

For $\zeta \in \mathcal{Z}(I)$, $\mathbf{1}_\zeta : a \mapsto a(\zeta)$.

$a \mapsto$ the coefficient of \mathbf{x}^α in the normal form of a .

The dual

The following computation gives the value of the linear form $1 + \mathbf{d}_1 + \mathbf{d}_1\mathbf{d}_2 + \mathbf{d}_3^2$ on the polynomial $1 + x_1 + x_1x_2$:

```
> apply((1+d[1]+d[1]*d[2]+d[3]^2), (1+x[1]+x[1]*x[2]));
```

3

Let us now illustrate the structure of *module*:

```
> (1+x[1]+x[1]*x[2]) &. (1+d[1]+d[1]*d[2]+d[3]^2);
```

$$3 + \mathbf{d}_1 + \mathbf{d}_1\mathbf{d}_2 + \mathbf{d}_3^2 + \mathbf{d}_2$$

We check that the constant term of this expansion is the value of the linear form $1 + \mathbf{d}_1 + \mathbf{d}_1\mathbf{d}_2 + \mathbf{d}_3^2$ at the polynomial $1 + x_1 + x_1x_2$.

Multiplication operators

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

$$\begin{array}{ccc} M_a : \mathcal{A} & \rightarrow & \mathcal{A} \\ u & \mapsto & a u \end{array} \quad \begin{array}{ccc} M_a^\dagger : \widehat{\mathcal{A}} & \rightarrow & \widehat{\mathcal{A}} \\ \Lambda & \mapsto & a \cdot \Lambda = \Lambda \circ M_a \end{array}$$

Theorem:

- **The eigenvalues of M_a are $\{a(\zeta_1), \dots, a(\zeta_d)\}$.**
- **The eigenvectors of all $(M_a^\dagger)_{a \in \mathcal{A}}$ 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

$$M_a = \begin{bmatrix} N_a^1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & N_a^d \end{bmatrix} \quad \text{with } N_a^i = \begin{bmatrix} a(\zeta_i) & & \star \\ & \ddots & \\ \mathbf{0} & & a(\zeta_i) \end{bmatrix}$$

Corollary: (Chow form)

$$\Delta(\mathbf{u}) = \det(u_0 + u_1 M_{x_1} + \dots + u_n M_{x_n}) = \prod_{\zeta \in \mathcal{Z}(I)} (u_0 + u_1 \zeta_1 + \dots + u_n \zeta_n)^{\mu_\zeta}.$$

Multiplication operators

Let us compute the matrix of multiplication by x_1 in the basis $(1, x_1, x_2, x_1x_2)$ of $\mathcal{A} = \mathbb{K}[x_1, x_2]/(f_1, f_2)$, where f_1, f_2 are the polynomials. We multiply these monomials by x_1 and reduce them to a normal form. According to the computations of the previous example, we have:

$$\begin{aligned} 1 \times x_1 &\equiv x_1, & x_1 \times x_1 &\equiv -x_1x_2 + x_1 + \frac{1}{6}, \\ x_2 \times x_1 &\equiv x_1x_2, & x_1x_2 \times x_1 &\equiv -x_1x_2 + \frac{55}{54}x_1 + \frac{2}{27}x_2 + \frac{5}{54}. \end{aligned}$$

```
> M1 := matrixof([x[1], x[1]*x[1] -f2, x[1]*x[2],  
> x[1]*(x[1]*x[2]) + 1/9*x[1]*f1 - (5/9 + 13/9*x[1] + 4/9*x[2])*f2],  
> [[1, x[1], x[2], x[1]*x[2]]]);
```

$$M_1 = \begin{bmatrix} 0 & \frac{1}{6} & 0 & \frac{5}{54} \\ 1 & 1 & 0 & \frac{55}{54} \\ 0 & 0 & 0 & \frac{2}{27} \\ 0 & -1 & 1 & -1 \end{bmatrix}.$$

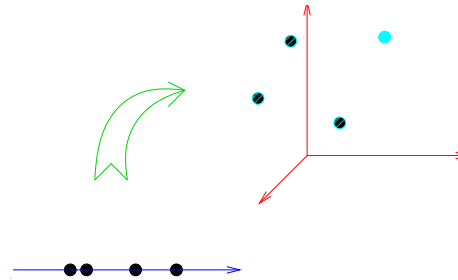
We compute the eigenvalues, their multiplicity, and the corresponding normalised eigenvector of the transposed of the matrix of multiplication by x_1 :

> `eigenvects(transpose(M1), 1);`

$$\left[-\frac{1}{3}, 2, \left\{ \left[1, -\frac{1}{3}, \frac{5}{6}, -\frac{5}{18}\right] \right\}, \left[\frac{1}{3}, 2, \left\{ \left[1, \frac{1}{3}, \frac{7}{6}, \frac{7}{18}\right] \right\}\right].$$

As the basis chosen for the computation is $(1, x_1, x_2, x_1x_2)$, the previous theorem tells us that the solutions of the system can be read off, from the 2^{nd} and the 3^{rd} coordinates of the normalised eigenvectors: $\zeta_1 = \left(-\frac{1}{3}, \frac{5}{6}\right)$ and $\zeta_2 = \left(\frac{1}{3}, \frac{7}{6}\right)$. Moreover, the 4^{th} coordinate of these vectors is the product of the 2^{nd} by the 3^{rd} coordinates.

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 $\zeta_1 = \frac{d_1(u_0)}{d'_0(u_0)}, \dots,$
 $\zeta_n = \frac{d_n(u_0)}{d'_0(u_0)}, d_0(u_0) = 0.$
4. Factorize $d_0(u_0)$ and keep the good factors for a minimal representation.

Remark: t is generic iff $\gcd(d_0(u_0), d'_0(u_0)) = 1$.

Chow form

We compute the Chow form of the variety $I = (f_1, f_2)$, using the matrices of multiplication by x_1 and x_2 , computed previously.

```
> factor(det(u[0]+ u[1]*M1+ u[2]*M2));
```

$$\left(u_0 + \frac{1}{3}u_1 + \frac{7}{6}u_2\right)^2 \left(u_0 - \frac{1}{3}u_1 + \frac{5}{6}u_2\right)^2$$

We check that it is a product of linear forms, whose coefficients yield the roots $\zeta_1 = (-\frac{1}{3}, \frac{5}{6})$ and $\zeta_2 = (\frac{1}{3}, \frac{7}{6})$. The exponents yield the multiplicity of the roots (here 2). As here the roots are rational, we can easily factorise this polynomial as a product of linear forms. But usually, this factorisation is possible only on an algebraic extension of the coefficient field.

From the previous Chow form, we deduce:

$$\xi_1 = -\frac{1}{6(1+u_0)}, \quad \xi_2 = \frac{11+12u_0}{12(1+u_0)}, \quad \left(u_0 + \frac{3}{2}\right) \left(u_0 + \frac{1}{2}\right) = 0.$$

which reduces to the constant representations

$$\begin{cases} u_0 = -3/2, x_1 = 1/3, x_2 = 5/6 \\ u_0 = -1/2, x_1 = -1/3, x_2 = 7/6 \end{cases}$$

Linear and quadratic forms

Let Λ be the linear form

$$\Lambda = 2 \times \mathbf{1}_{(-1/3, 5/6)} + 2 \times \mathbf{1}_{(1/3, 7/6)}.$$

We check that $(-1/3, 5/6)$ and $(1/3, 7/6)$ are in $\mathcal{Z}(f_1, f_2)$. The matrix of Q_Λ in the basis $\{1, x_1, x_2, x_1 x_2\}$ of \mathcal{A} is

$$[Q_\Lambda] = \begin{bmatrix} \Lambda(1) & \Lambda(x_1) & \Lambda(x_2) & \Lambda(x_1 x_2) \\ \Lambda(x_1) & \Lambda(x_1^2) & \Lambda(x_1 x_2) & \Lambda(x_1^2 x_2) \\ \Lambda(x_2) & \Lambda(x_1 x_2) & \Lambda(x_2^2) & \Lambda(x_1 x_2^2) \\ \Lambda(x_1 x_2) & \Lambda(x_1^2 x_2) & \Lambda(x_1 x_2^2) & \Lambda(x_1^2 x_2^2) \end{bmatrix} = \begin{bmatrix} 4 & 0 & 4 & \frac{2}{9} \\ 0 & \frac{4}{9} & \frac{2}{9} & \frac{4}{9} \\ 4 & \frac{2}{9} & \frac{37}{9} & \frac{4}{9} \\ \frac{2}{9} & \frac{4}{9} & \frac{4}{9} & \frac{37}{81} \end{bmatrix}$$

Trace

$$\text{Tr}(x_1) := \text{trace}(M_{x_1}) = \text{trace} \left(\begin{bmatrix} 0 & \frac{1}{6} & 0 & \frac{5}{54} \\ 1 & 1 & 0 & \frac{55}{54} \\ 0 & 0 & 0 & \frac{2}{27} \\ 0 & -1 & 1 & -1 \end{bmatrix} \right) = 0$$

By a direct computation, we get $\text{Tr}(1) = 4$, $\text{Tr}(x_1) = 0$, $\text{Tr}(x_2) = 4$, $\text{Tr}(x_1 x_2) = \frac{2}{9}$.

By using the transposed operators $M_{x_i}^t$ we get:

```
> T0 := evalm([4,0,4,2/9]);
> T1 := evalm(transpose(M1)&*T0): T2:= evalm(transpose(M2)&*T0):
> T11 := evalm(transpose(M1)&*T1): T12:= evalm(transpose(M2)&*T1):
> T112:= evalm(transpose(M2)&*T11):
> Q1 := matrix(4,4,[T0,T1,T2,T12]);
> Qx1 := matrix(4,4,[T1,T11,T12,T112]);
```

$$Q_1 = \begin{pmatrix} 4 & 0 & 4 & \frac{2}{9} \\ 0 & \frac{4}{9} & \frac{2}{9} & \frac{4}{9} \\ 4 & \frac{2}{9} & \frac{37}{9} & \frac{4}{9} \\ \frac{2}{9} & \frac{4}{9} & \frac{4}{9} & \frac{37}{81} \end{pmatrix}, \quad Q_{x_1} = \begin{pmatrix} 0 & \frac{4}{9} & \frac{2}{9} & \frac{4}{9} \\ \frac{4}{9} & 0 & \frac{4}{9} & \frac{2}{81} \\ \frac{2}{9} & \frac{4}{9} & \frac{4}{9} & \frac{37}{81} \\ \frac{4}{9} & \frac{2}{81} & \frac{37}{81} & \frac{4}{81} \end{pmatrix}$$

Real roots

Let $f_i \in \mathbb{R}[\mathbf{x}]$, $i = 1, \dots, m$ and $\mathcal{Z}_{\mathbb{C}} = \{\zeta \in \mathbb{C}^n, f_1(\zeta) = \dots = f_m(\zeta) = 0\}$.

Consider the **linear form**:

$$Tr : a \mapsto \text{trace}(M_a) = \sum_{\zeta \in \mathcal{Z}} \mu_{\zeta} a(\zeta)$$

and the associated **quadratic form**:

$$Q_h : (a, b) \mapsto Tr(h a b).$$

Its matrix in the basis $(\mathbf{x}^{\alpha})_{\alpha \in E}$ is $(Tr(h \mathbf{x}^{\alpha+\beta}))_{\alpha, \beta \in E}$.

Theorem: (Hermite) Let $h \in \mathbb{R}[\mathbf{x}]$.

- 1. The rank of Q_h is the number of (complex) roots $\zeta \in \mathcal{Z}_{\mathbb{C}}$ st. $h(\zeta) \neq 0$.**
- 2. The signature of Q_h is**
 $\#\{\zeta \text{ real root such that } h(\zeta) > 0\} - \#\{\zeta \text{ real root such that } h(\zeta) < 0\}$.

Example (continued)

$$Q_1 = \begin{pmatrix} 4 & 0 & 4 & \frac{2}{9} \\ 0 & \frac{4}{9} & \frac{2}{9} & \frac{4}{9} \\ 4 & \frac{2}{9} & \frac{37}{9} & \frac{4}{9} \\ \frac{2}{9} & \frac{4}{9} & \frac{4}{9} & \frac{37}{81} \end{pmatrix}, \quad Q_{x_1} = \begin{pmatrix} 0 & \frac{4}{9} & \frac{2}{9} & \frac{4}{9} \\ \frac{4}{9} & 0 & \frac{4}{9} & \frac{2}{81} \\ \frac{2}{9} & \frac{4}{9} & \frac{4}{9} & \frac{37}{81} \\ \frac{4}{9} & \frac{2}{81} & \frac{37}{81} & \frac{81}{4} \end{pmatrix}$$

The rank and the signature of the quadratic forms Q_1, Q_{x_1} are

> $\text{rank}(Q_1), \text{signature}(Q_1), \text{rank}(Q_{x_1}), \text{signature}(Q_{x_1});$

$$2, [2, 0], 2, [1, 1],$$

which tell us (without computing these roots) that there are 2 real roots, one with $x_1 < 0$ and one with $x_1 > 0$.

Computing the quotient structure

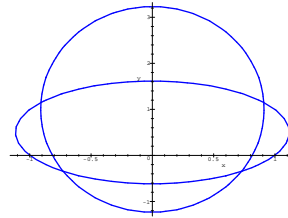
Gröbner basis methods

- A total **monomial ordering** \prec , compatible with the monomial multiplication.
- The generators $f_i = m - m_1 - m_2 - \dots$ of I are seen as rewriting rules
$$m \rightarrow m_1 + m_2 + \dots$$
- The Gröbner basis computation is a **completion** procedure st.
$$f \in I \text{ iff } f \xrightarrow{*} 0.$$
- A **basis** of \mathcal{A} is the set of monomials which are **not divisible by the leading monomials** of the rewriting rules.
- The **multiplication table** M_a :
 - multiplication of the monomial basis by a .
 - normalisation.

GB are going boink, don't they ?

A system:

$$\begin{cases} p_1 := a x_1^2 + b x_2^2 + l_1(x_1, x_2) \\ p_2 := c x_1^2 + d x_2^2 + l_2(x_1, x_2) \end{cases}$$

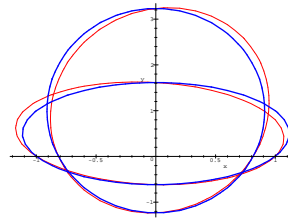


Basis of $\mathcal{A} = \mathbb{K}[x_1, x_2]/(p_1, p_2)$:

$$(1, x_1, x_2, x_1 x_2).$$

A small perturbation:

$$\begin{cases} \tilde{p}_1 := p_1 + \epsilon_1 x_1 x_2 \\ \tilde{p}_2 := p_2 + \epsilon_2 x_1 x_2 \end{cases}$$



Basis of $\tilde{\mathcal{A}} = \mathbb{K}[x_1, x_2]/(\tilde{p}_1, \tilde{p}_2)$:

$$(1, x_1, x_2, x_2^2).$$

Catastroph !

- A new set of monomials for the basis.
- Big coefficients (in $\frac{1}{\epsilon_i}$) appear.

Perturbation of Grobner basis

```
> f1 := x[1]^2+x[2]^2 -x[1]+x[2]-2;
> f2 := x[1]^2-x[2]^2 + 2*x[2]-3;
> gbasis([f1,f2],tdeg(x[1],x[2]));
```

$$[2x_2^2 - x_1 - x_2 + 1, 2x_1^2 - x_1 + 3x_2 - 5]$$

Leading monomials: x_1^2, x_2^2 . Basis of \mathcal{A} : $\{1, x_1, x_2, x_1 x_2\}$

A small perturbation:

```
> gbasis([f1,f2+1./10000000*x[1]*x[2]],tdeg(x[1],x[2]));
```

$$[0.0000001x_1x_2 - 2x_2^2 + x_1 + x_2 - 1, x_1^2 + x_2^2 - x_1 + x_2 - 2, \\ x_2^3 - 10000000.99999999999999950000000000000000125x_2^2 \\ + 5000000.2500000124999993749999687500015625000781250x_1 \\ + 5000000.7500000374999931249999062500171875002343750x_2 \\ - 5000000.2500000624999993749998437500015625003906250]$$

Leading monomials: $x_1 x_2, x_1^2, x_2^3$. Basis of \mathcal{A} is $\{1, x_1, x_2, x_2^2\}$.

What we would like to have:

- The basis $(1, x_1, x_2, x_1 x_2)$
- Rewriting rules of the form:

$$x_1^2 \equiv -0.00000005 x_1 x_2 + 1/2 x_1 - 3/2 x_2 + 5/2$$

$$x_2^2 \equiv +0.00000005 x_1 x_2 + 1/2 x_1 + 1/2 x_2 - 1/2$$

$$x_2 x_1^2 \equiv 0.49999999 x_1 x_2 - 0.74999998 x_1 + 1.75000003 x_2 + 0.74999994,$$

$$x_1 x_2^2 \equiv 0.49999999 x_1 x_2 - 0.25000004 x_1 - 0.74999991 x_2 + 1.25000004]$$

This set of relations yields directly the matrices of multiplication by x_1, x_2 in \mathcal{A} .

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.

A new normal form criterion

Proposition: If B is a basis of \mathcal{A} for some regular value of the parameters, it will be the case in a neighborhood.

- We fix the vector space $\langle B \rangle$ on which, we want to normalize. polynomials
- $B^+ = B \cup x_1 B \cup \dots \cup x_n B$

Hypothesis:

- B is a set of monomials of $R = \mathbb{K}[\mathbf{x}]$, connected to 1: $\forall m = x_{i_1} \cdots x_{i_k} \in B$, $\forall l = 0, \dots, k$, $x_{i_1} \cdots x_{i_l} \in B$.
- A projection $N : \langle B^+ \rangle \rightarrow \langle B \rangle$, with $N^2 = N$.

Question: IS N A NORMAL FORM MODULO $I = (\text{Ker}(N))$?

Theorem: Let $M_i : \langle B \rangle \rightarrow \langle B \rangle$ such that $M_i(b) = N(x_i b)$.
 $M_i \circ M_j = M_j \circ M_i, i, j = 1, \dots, n \Leftrightarrow B$ basis of $\mathcal{A} = R/I$.

Algorithm of normal form

Algorithm: Normal form for \mathcal{A} .

INPUT: Let $f_1, \dots, f_m \in R$, $I = (f_1, \dots, f_m)$ and assume that $\mathcal{A} = R/I$ is zero-dimensional. Let L be a finite vector space of R containing f_1, \dots, f_m and connected to 1.

(1) $K_0 = \langle f_1, \dots, f_m \rangle$;

(2) While $K_n \neq K_{n-1}$, compute $K_{n+1} = K_n^+ \cap L$, replace n by $n + 1$;

(3) Compute a supplementary vector space B of $K_*(= K_{n_0})$, which is connected to 1;

(4) If $B^+ \not\subset L$, replace L by L^+ and go to step (2). Otherwise stop.

OUTPUT: A basis of K_* .

- The projection for B^+ along K^* on B is a normal form.
- Linear algebra on vector spaces of polynomials (sparse matrices, superlu).
- Applications: Gröbner basis as a special case, Laurent polynomials, local rings
...

Timing for square systems (normal form + eigenvectors):

Degree	Nb Var	Nb Sol	Machine	Coefficients	Time	$Max(f_i)$
2	3	8	ultra1	[-1,1]	0.05(0.02)s	1e-11
2	3	8	ultra1	[-1000,1000]	0.05(0.02)s	1e-10
2	3	8	PII933	[-1,1]	"0.01"s	1e-11
2	3	8	PII933	[-1000,1000]	"0.01"s	1e-10
3	3	27	PIII933	[-1,1]	0.02(0.01)s	1e-11
3	3	27	PIII933	[-1000,1000]	0.02s	1e-7
3	3	27	ultra1	[-1,1]	0.22(0.04)s	1e-11
3	3	27	ultra1	[-1000,1000]	0.22(0.04)s	1e-7
4	4	256	ultra1	[-1,1]	186.36(1.31)s	1e-3
4	5	1024	PIII933	[-1,1]	-(0.15)s	-

Time: mean over 10 runs. """: most of the time spent for disk access etc...

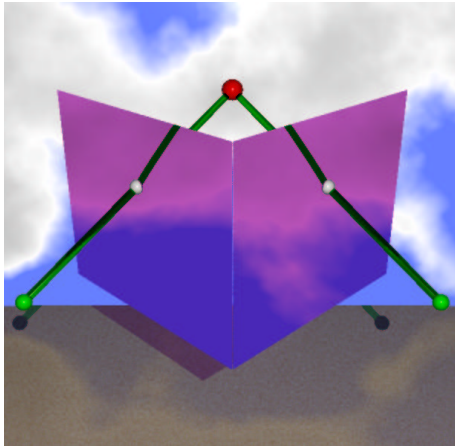
Numerical degradation: eigenvectors computation.

Macaulay revisited

Size of the linear system(s) to invert:

n	5	6	7	8	9	10	11
	5	6	7	8	9	10	11
	20	30	42	56	72	90	110
	30	60	105	168	252	360	495
	20	60	140	280	504	840	1320
	5	30	105	280	630	1260	2310
		6	42	168	504	1260	2772
			7	56	252	840	2310
				8	72	360	1320
					9	90	495
						10	110
							11
Σ	80	192	448	1024	2304	5120	11264
M	430	1 652	6 307	24 054	91 866	351 692	1 350 030
D	32	64	128	256	512	1024	2048

Autocalibration Problem, in computer vision



- Camera, pinhole model
- if \mathbf{m}, \mathbf{m}' two images of $M \in \mathbb{R}^3$ in two photos $\mathbf{m} F \mathbf{m}' = 0$ where F is the Fundamental matrix.

Problem: Compute the intrinsic parameter matrix, using Kruppa equations:

$$F X F^t = \lambda T_e X T_e'$$

□ 6 quadratic homogeneous equations in 6 variables (0.38s, Alpha 500Mhz).

Exact root	Computed root
1.049401330318981	1.049378730793354
4.884653820635368	4.884757558650871
6.011985256613766	6.011985146332036
.1726009605860270	.1725610425715577
1.727887086410446	1.727898150468536

Blind identification in Signal Processing

□ Transmission of an input signal $\mathbf{x}(n)$ of information of size p depending on the discrete time n into a channel of length L . The output is $\mathbf{y}(n)$.

□ We want to compute the impulse response matrix $H(n)$ s.t.

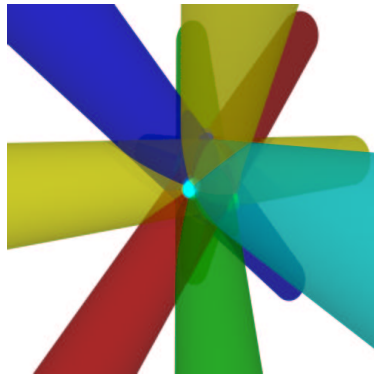
$$\mathbf{y}(n) = \sum_{m=0}^{L-1} H(m)\mathbf{x}(n-m) + \mathbf{b}(n), \mathbf{b}(n) \text{ is the noise.}$$

If $\mathbf{b}(n)$ is Gaussian centred, a statistic analysis yields

$$\sum_{m=0}^{L-1} \sum_{i=1}^p h_{\alpha,i}(m)h_{\beta,i}(m)(-1)^{n-m} = E(y_{\alpha}(n)y_{\beta}(n-l)).$$

	A real root
x0	-1.803468527372455
x1	-5.162835380624794
x2	-7.568759900599482
x3	-6.893354578266418
x4	-3.998807562745594
x5	-1.164422870375179
Error = 10^{-8} , Time = 0.76s	

Cylinders through 4 and 5 points



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

<i>Problem</i>	<i>time</i>	<i>max(f_i)</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

Controlled iterative methods

We consider the elements in \mathcal{A} as multiplicative operators. Induction in the quotient \mathcal{A} .

- $u_{n+1} = f_0 u_n$ (Bernoulli).
- $u_{n+1} \equiv u_n^2$ (Sebastio e Sylva).
- $u_{n+1} \equiv \frac{1}{2}(u_n \pm u_n^{-1})$ (Joukovski).

- Convergence to one (or a subsum) of the "eigenvectors" or **idempotents** associated with the roots.
- Convergence from the beginning and quadratic for Sebastio and Joukovski.

Applications:

- Compute the root(s) which maximize(s), minimize(s) $|h|$.
- Count/compute the roots which are almost real.
- Count/compute the roots in a box.

Iterative methods in one variable

- **Allows to select the root of $f(x) = 0$ st. $|u_0(\zeta)|$ maximum, minimum, $Re(u_0(\zeta)) > 0$, almost real, in a box,**
- **(Fast and controlled)** convergence toward a combination of idempotents of \mathcal{A} .

Exemple: Sebastiao e Sylva, by H. Prieto in

<http://www.inria.fr/galaad/logiciels/ALP/>

	d	x0	k	T
double	10^4	0.0	8	22.03s
double	10^5	0.0	8	346.52s
complex	10^4	(0.4,0.6)	6	16.86s
complex	10^5	(0.4,0.6)	6	292.02s
complex	10^6	(0.4,0.6)	7	2520s

Let us consider a system with two real and two complex roots:

```
> f1:= x[1]^2+2*x[1]*x[2]-x[1]-1;
> f2:= x[1]^2+x[2]^2-8*x[1];
```

Approximation of the roots are

ζ_1	ζ_2	ζ_3	ζ_4
6.8200982	$-0.19395427 + 0.20520688 i$	$-0.19395427 - 0.20520688 i$	0.36781361
-2.8367388	$-0.61937124 - 1.3895199 i$	$-0.61937124 + 1.3895199 i$	1.6754769

We illustrate the Sebastiao e Sylva method by computing first, the root for which $|x_1|$ is maximal. We start with $u_0 = x_1$. After 4 iterations, we obtain

$$u_4 = 7.6055995 + 7.7975926x_1 - 0.46159096x_2 - 15.740471x_1x_2.$$

By multiplying it by x_1 and x_2 in \mathcal{A} , we obtain $\zeta_1 = (6.820095, -2.836734)$. If we start with

$$u_0 \equiv \left(x_1 - \frac{1}{2}\right)^{-1} \equiv -\frac{78}{35} - \frac{228}{35}x_1 - \frac{32}{35}x_2 - \frac{16}{7}x_1x_2,$$

the algorithm should converge to the root for which x_1 is the closest to $\frac{1}{2}$. Indeed, after 4 iterations, we obtain

$$u_4 = 0.15292071 + 0.89409187x_1 + 0.16270766x_2 + 0.29923055x_1x_2,$$

which yields the root $\zeta_4 = (0.3678148, 1.675476)$.

Homotopic methods

Weierstrass in one variable

$$P(x) = \prod_{i=1}^d (x - \zeta_i) = x^d + \sum_{i=0}^{d-1} \sigma_{d-i}(\zeta) x^i = x^d + a_1 x^{d-1} + \dots + a_d \quad (1)$$
$$(\Sigma) : \begin{cases} \sigma_i(\zeta) - a_i \\ i \in \{1, \dots, d\} \end{cases}$$

Apply Newton's method to the system $(\Sigma) \Rightarrow$ Weierstrass, Durand-Kerner's method.

Advantages:

- No inversion of the jacobian matrix. An explicit formula for the iteration

$$I\left(\mathbf{z}^{(n+1)}\right)_i = \mathbf{z}_i^{(n)} - \frac{P\left(\mathbf{z}_i^{(n)}\right)}{\prod_{j \neq i} \mathbf{z}_i^{(n)} - \mathbf{z}_j^{(n)}} \quad (2)$$

- Unproved global convergence.

Relations between roots and coefficients

We consider $\{f_1, \dots, f_n\} \subset \mathbb{K}[\mathbf{x}]$, and $\mathcal{A} = \mathbb{K}[x_1, \dots, x_n] / (f_1, \dots, f_n)$.

□ $\mathcal{Z}(f_1, \dots, f_n) = \left\{ \mathbf{z} \in \overline{\mathbb{K}}^n \mid f_1(\mathbf{z}) = \dots = f_n(\mathbf{z}) = 0 \right\} = \{\mathbf{z}_1, \dots, \mathbf{z}_D\}$.

□ A basis $\mathbf{x}^E = (\mathbf{x}^\alpha)_{\alpha \in E}$ of \mathcal{A} as \mathbb{K} -vector space ($E = \{\alpha_1, \dots, \alpha_D\} \subset \mathbb{N}^n$).

□ Decomposition of $\mathcal{A} = \mathbb{K}\mathbf{e}_{\zeta_1} \oplus \dots \oplus \mathbb{K}\mathbf{e}_{\zeta_D}$.

□ **idempotent:**

$$\mathbf{e}_{z_i}(\mathbf{z}, \mathbf{x}) = (-1)^i \begin{vmatrix} \mathbf{x}^{\alpha_1} & \dots & \mathbf{x}^{\alpha_D} \\ \mathbf{z}_1^{\alpha_1} & \dots & \mathbf{z}_1^{\alpha_D} \\ \vdots & & \vdots \\ \mathbf{z}_{i-1}^{\alpha_1} & \dots & \mathbf{z}_{i-1}^{\alpha_D} \\ \mathbf{z}_{i+1}^{\alpha_1} & \dots & \mathbf{z}_{i+1}^{\alpha_D} \\ \vdots & & \vdots \\ \mathbf{z}_D^{\alpha_1} & \dots & \mathbf{z}_D^{\alpha_D} \end{vmatrix} / V_E(\mathbf{z}) \text{ with } V_E(\mathbf{z}) = \begin{vmatrix} \mathbf{z}_1^{\alpha_1} & \dots & \mathbf{z}_1^{\alpha_D} \\ \vdots & \ddots & \vdots \\ \mathbf{z}_D^{\alpha_1} & \dots & \mathbf{z}_D^{\alpha_D} \end{vmatrix}.$$

$$\bullet R_Q(\mathbf{z}, \mathbf{x}) = \begin{vmatrix} Q(\mathbf{x}) & \mathbf{x}^{\alpha_1} & \dots & \mathbf{x}^{\alpha_D} \\ Q(\mathbf{z}_1) & \mathbf{z}_1^{\alpha_1} & \dots & \mathbf{z}_1^{\alpha_D} \\ \vdots & \dots & \vdots & \vdots \\ Q(\mathbf{z}_D) & \mathbf{z}_D^{\alpha_1} & \dots & \mathbf{z}_D^{\alpha_D} \end{vmatrix}$$

$$\bullet F_Q(\zeta, \mathbf{x}) = Q(\mathbf{x}) - \frac{R_Q(\mathbf{z}, \mathbf{x})}{V_E(\zeta)}$$

Proposition:

- $R_Q(\mathbf{z}, \mathbf{z}_i) = 0, \forall i \in \{1, \dots, D\},$
- $F_Q(\mathbf{x})$ is the normal form of Q in the basis \mathbf{x}^E of \mathcal{A} .

$$\text{Proposition: } \frac{\partial}{\partial z_{i,j}} F_Q(\mathbf{z}, \mathbf{x}) = \frac{\frac{\partial R_Q(\mathbf{z}, \mathbf{z}_i)}{\partial x_j}}{V_E(\mathbf{z})} \mathbf{e}_{\mathbf{z}_i}(\mathbf{z}, \mathbf{x})$$

Weierstrass's method for multivariate algebraic systems

$$\bullet \mathcal{F} : \begin{cases} (K^n)^D & \rightarrow & (\mathbb{K}[x_1, \dots, x_n]_E)^n \\ \mathbf{z} & \mapsto & \mathcal{F}(\mathbf{z}, \mathbf{x}) = \begin{pmatrix} F_{f_1} \\ \vdots \\ F_{f_n} \end{pmatrix} \end{cases}$$

$$\bullet \Delta_{z_i} \mathcal{F}(\mathbf{z}) = \begin{pmatrix} \frac{\frac{\partial R_{f_1}(\mathbf{z}^{(k)}, \mathbf{z}_i^{(k)})}{\partial x_1}}{V_E(\mathbf{z})} & \cdots & \frac{\frac{\partial R_{f_1}(\mathbf{z}^{(k)}, \mathbf{z}_i^{(k)})}{\partial x_n}}{V_E(\mathbf{z})} \\ \vdots & \ddots & \vdots \\ \frac{\frac{\partial R_{f_n}(\mathbf{z}^{(k)}, \mathbf{z}_i^{(k)})}{\partial x_1}}{V_E(\mathbf{z})} & \cdots & \frac{\frac{\partial R_{f_n}(\mathbf{z}^{(k)}, \mathbf{z}_i^{(k)})}{\partial x_n}}{V_E(\mathbf{z})} \end{pmatrix}.$$

Theorem: Iteration formula points of Newton's method's applied to \mathcal{F} :

$$\mathbf{z}_i^{(k+1)} = \mathbf{z}_i^{(k)} - \Delta_{z_i} \mathcal{F}(\mathbf{z})^{-1} \begin{pmatrix} f_1(\mathbf{z}_i^{(k)}) \\ \vdots \\ f_n(\mathbf{z}_i^{(k)}) \end{pmatrix} \quad (3)$$

Local algorithm

Algorithm: Weierstrass's iteration function

• Input : \mathbf{z} , E and (f_1, \dots, f_n) .

• Step 1 : Compute the following objects (specialized at \mathbf{z}) :

$$MV_E(\mathbf{z}) = \begin{pmatrix} \mathbf{z}_1^{\alpha_1} & \dots & \mathbf{z}_1^{\alpha_D} \\ \vdots & \ddots & \vdots \\ \mathbf{z}_D^{\alpha_1} & \dots & \mathbf{z}_D^{\alpha_D} \end{pmatrix}, \quad MF(\mathbf{z}) = \begin{pmatrix} f_1(\mathbf{z}_1) & \dots & f_n(\mathbf{z}_1) \\ \vdots & \ddots & \vdots \\ f_1(\mathbf{z}_D) & \dots & f_n(\mathbf{z}_D) \end{pmatrix}$$

and $dv_{f_k}^{(i)}(\mathbf{x}) = \left(\frac{\partial}{\partial x_i} \mathbf{x}^{\alpha_1}, \dots, \frac{\partial}{\partial x_i} \mathbf{x}^{\alpha_D}, \frac{\partial}{\partial x_i} f_k(\mathbf{x}) \right)$.

• Step 2 : Solve $MV_E(\mathbf{z}) * Y = MF(\mathbf{z})$, the column of index k in Y is denoted by S_{f_k} .

• Step 3 : $dv_{f_j}^{(i)}$ (specialized at \mathbf{z}_i) inner product with $S_{f_k}(\mathbf{z})$ (normalized by its last coefficient) $\rightarrow_{j,k} \Delta_{\mathbf{z}_i} \mathcal{F}(\mathbf{z})$

• Step 4 : For each $i \in \{1, \dots, D\}$ we solve the system $(f_1(\mathbf{z}_i), \dots, f_i(\mathbf{z}_i))^t = \Delta_{\mathbf{z}_i} \mathcal{F} * \mathbf{z}'_i$ where \mathbf{z}'_i is the column of index i of the output matrix.

• Output : \mathbf{z}' .

Proposition: The arithmetic complexity of a Weierstrass's iteration is bounded by $\mathcal{O}(D^3 + n^2D^2 + Dn^3)$ arithmetic operations.

Weierstrass's method with continuation

□ (g_1, \dots, g_n) a system with **known solutions** $\mathbf{z}^{(0)}$ with \mathbf{x}^E as basis of $\mathcal{A}_{\mathbf{z}^{(0)}}$.

$$\square \begin{cases} [0, 1] & \longrightarrow & \mathbb{K}[x_1, \dots, x_n]^n \\ t & \longmapsto & \mathbf{H}(t) = \begin{pmatrix} tf_1 + (1-t)g_1 \\ \vdots \\ tf_n + (1-t)g_n \end{pmatrix} \end{cases}$$

Algorithm: Weierstrass method with continuation

- Input : $\mathbf{z}^{(0)}$, (f_1, \dots, f_n) , (g_1, \dots, g_n) , the set E , and $M \in \mathbb{N} - \{0\}$.
- For i from 1 to M do
$$\mathbf{z}^{(i)} = \text{Weierstrass}(\mathbf{H}(i/M), \mathbf{z}^{(i-1)})$$
- Output : Return $\mathbf{z}^{(M)}$.

Geometric methods

Aim: Project the problem onto a smaller (equivalent) one. Algebraically, we eliminate some of the variables.

- 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**).

Resultants

Condition on $\mathbf{c} = (c_{i,j})$ such that the system has a solution in the projective variety X of dimension n :

$$\begin{cases} f_0(\mathbf{x}) &= \sum_{j=0}^{k_0} c_{0,j} \kappa_{0,j}(\mathbf{x}) \\ &\vdots \\ f_n(\mathbf{x}) &= \sum_{j=0}^{k_n} c_{n,j} \kappa_{n,j}(\mathbf{x}) \end{cases}$$

- Projection on the space of coefficients: hypersurface defined by the resultant

$$\text{Res}_X(\mathbf{c}) = 0$$

- Explicit formula for the degree in the coefficients of each f_i .
- Explicit construction as maximal minor of the matrix of a map such as

$$\begin{aligned} \mathcal{S} : \langle \mathbf{x}^{E_0} \rangle \times \cdots \times \langle \mathbf{x}^{E_n} \rangle &\rightarrow \langle \mathbf{x}^F \rangle \\ (q_0, \dots, q_n) &\mapsto \sum_{i=0}^n q_i f_i \end{aligned}$$

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. Macaulay (1902), Jouanolou (1990).

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

Polytope geomtry. Sylvester-like matrix. Maximal minors. Ratio of two Determinants (BBK75, GKZ91, PSCE93, DA01).

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

Bezoutian matrix. Maximal minors. A multiple of $\text{Res}_X(\mathbf{c})$. (EM99, BEM00).

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

Explicit resolution of $(F : G)$. Gcd of the maximal minors. Degree formula. Ratio of determinants. (BKM75, BEM00, B01).

Solving $f_1 = \dots = f_n = 0$ by hiding a variable

1. Construct the resultant matrix $S(x_n)$ of f_1, \dots, f_n as polynomials in x_1, \dots, x_{n-1} with coefficients in $\mathbb{K}[x_n]$.
2. Solve $v S(x_n) = 0$.
 - Either by solving $\det(S(x_n)) = 0$ and by deducing the corresponding v .
 - or by reducing it to an eigenproblem:

$$v \left(\begin{array}{c} v (S_d x_n^d + S_{d-1} x_n^{d-1} + \dots + S_0) = 0 \Rightarrow \\ \left[\begin{array}{cccc} 0 & \dots & 0 & -S_0 S_d^{-1} \\ \mathbb{I} & \ddots & \vdots & \vdots \\ \vdots & \ddots & 0 & -S_{d-2} S_d^{-1} \\ 0 & \dots & \mathbb{I} & -S_{d-1} S_d^{-1} \end{array} \right] - x_n \mathbb{I}_{ds} \end{array} \right) = 0,$$

3. Deduce the other coordinates of the roots from v .

We illustrate this algorithm on the system

$$\begin{cases} f_1 = x_1 x_2 + x_3 - 2, \\ f_2 = x_1^2 x_3 + 2 x_2 x_3 - 3, \\ f_3 = x_1 x_2 + x_2^2 + x_2 x_3 - x_1 x_3 - 2 \end{cases}$$

We hide x_3 and use the projective resultant formulation. We obtain a 15×15 matrix $S(x_3)$, and compute its determinant:

```
> S := mresultant([f1,f2,f3],[x1,x2]): det(S);
```

$$x_3^4 (x_3 - 1) (2 x_3^5 - 11 x_3^4 + 20 x_3^3 - 10 x_3^2 + 10 x_3 - 27).$$

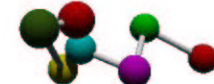
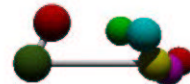
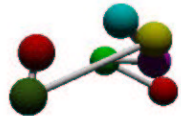
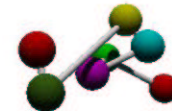
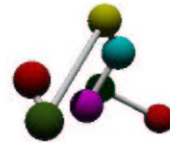
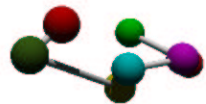
The root $x_3 = 0$ does not yield an affine root of the system $f_1 = f_2 = f_3 = 0$ (the corresponding point is at infinity). Substituting $x_3 = 1$ in $S(x_3)$, we get a matrix of rank 14. The kernel of $S(1)^t$ is generated by

$$[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$$

which implies that the corresponding root is $(1, 1, 1)$. For the other eigenvalues (which are the roots of the last factor), we proceed similarly in order to obtain the 5 other (simple) roots of $f_1 = f_2 = f_3 = 0$.

Conformation of molecules

Problem: Compute the possible conformations of a molecule when the position and orientation of the extremity is fixed.

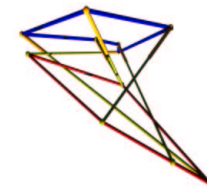
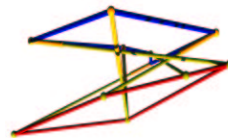
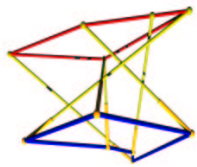
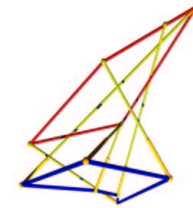
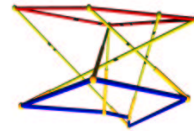
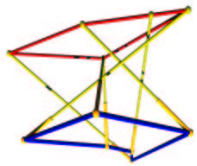


Error: $\|A_1 \circ \dots \circ A_6 - H\| < 10^{-6}$

Time: 0.090s

Direct kinematic problem of a parallel robot:

Problem: Compute the position of the platform for fixed lengths of the arms.



Error: $|\|R Y_i + T - X_i\|^2 - d_i^2| < 10^{-6}$

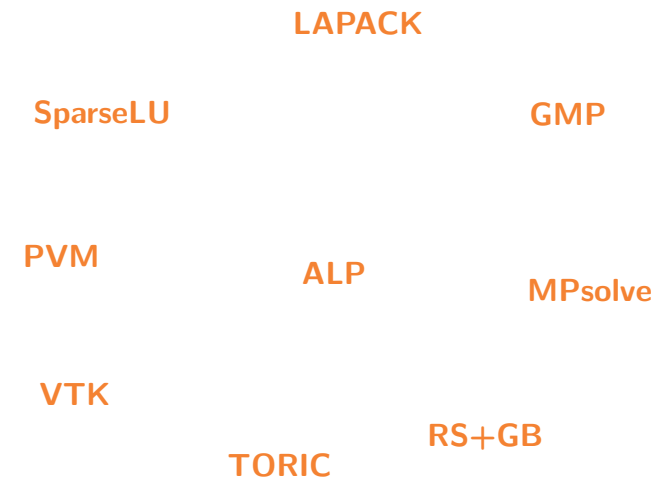
Time: 132s + 0.5s

Other applications

- Solving overdetermined systems.
- Elimination of variables, projection on subspaces.
- From parameterized to implicit equations.
- Intersection of parameterized curves, surfaces.
- Offset of curves, surfaces.
- Univariate representation of isolated points.
- Geometrical decomposition of varieties.
- Certification of geometric predicates.
-

ALP: an environment for symbolic and numeric computations

- Basic data structures : **vectors**, **matrices (dense, Toeplitz, Hankel, sparse, . . .)**, **univariate polynomials**, **multivariate polynomials**.
- **Genericity of the coefficient type, the internal representation and specialisation.**
- Parametrised type, template expression, instantiation at compile time.
- C++, under LGPL,
- <http://www.inria.fr/galaad/logiciels/ALP/>,



Three level of objects:

- **Container** : internal representation, associated with iterators, and access/modification methods.

```
typedef array2d<double> rep1; typedef lapack<double> rep2;
```

- **View** : How we see the container, eg. as a `Vector<R>` or as a univariate polynomial `UPolyDense<R>`.

```
typedef MatDense<lapack<double> > Mat;  
typedef Monom<double, dynamicexp<'x'> > Mon;  
typedef MPoly<list<Mon>, Dlex<Mon> > Pol;
```

Local views sharing datas: `MatRef S = M(Range(3,10),Range(2,9)); Mat N=S*S;`

- **Module** : set of (generic) functions which apply to a category of objects (eg. `VECTOR::Print`, `MATRIX::mult`). Allows specialisation for some data types.

```
template<>  
void MATRIX::mult(lapack<double> & r, lapack<double> & a, lapack<double> & b) {
```