Approximate Algebraic Geometry

for curves and surfaces

B. Mourrain

INRIA, BP 93, 06902 Sophia Antipolis

1st October 2003

Symbolic-numeric methods

for curves and surfaces

B. Mourrain

INRIA, BP 93, 06902 Sophia Antipolis

1st October 2003

Certified/controlled algorithms for curves and surfaces

B. Mourrain

INRIA, BP 93, 06902 Sophia Antipolis

1st October 2003

Long time ago

Long time ago

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

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).

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).

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* $\sqrt{2}$.

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*.

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.

Effective Geometric Objects

- **Effective real numbers** are such that we can ask:
  - algebraic::approximate(int n) a numerical approximation to an arbitrary precision 2^n,
  - compare(algebraic a, algebraic b) with a certified answer.

Effective Geometric Objects

- **Effective real numbers** are such that we can ask:
  - algebraic::approximate(int n) a numerical approximation to an arbitrary precision 2^n,
  - compare(algebraic a, algebraic b) with a certified answer.

- **Effective points, segments, ...**
  - defined with effective coordinates,
  - or provide directly approximation and queries at the geometric level.

Effective Geometric Objects

☐ **Effective real numbers** are such that we can ask:

- algebraic::approximate(int n) a numerical approximation to an arbitrary precision 2^n,
- compare(algebraic a, algebraic b) with a certified answer.

☐ **Effective points, segments, ...**

- defined with effective coordinates,
- or provide directly approximation and queries at the geometric level.

☐ **Effective curved objects**

- defined implicitly with algorithms to operate on them,
- combine it with numerical, geometric approximation.

Effective Geometric Objects

☐ **Effective real numbers** are such that we can ask:

- algebraic::approximate(int n) a numerical approximation to an arbitrary precision 2^n,
- compare(algebraic a, algebraic b) with a certified answer.

☐ **Effective points, segments, ...**

- defined with effective coordinates,
- or provide directly approximation and queries at the geometric level.

☐ **Effective curved objects**

- defined implicitly with algorithms to operate on them,
- combine it with numerical, geometric approximation.

Numbers

Algebraic numbers

Represented as

- ☐ 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.

Algebraic numbers

Represented as

- □ 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.

- Isolation via Descartes, Uspenksy, de Casteljau, Sturm algorithm.

Algebraic numbers

Represented as

□ 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.

• Isolation via Descartes, Uspenksy, de Casteljau, Sturm algorithm.

• Queries via Sturm(-Habicht) method, interval arithmetic, numerical approximation and separating bounds.

Algebraic numbers

Represented as

□ 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.

• Isolation via Descartes, Uspenksy, de Casteljau, Sturm algorithm.

• Queries via Sturm(-Habicht) method, interval arithmetic, numerical approximation and separating bounds.

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

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

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***.

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***.

- $f(0) = b_0, f(1) = b_d,$

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 $\Delta(\mathbf{b})_i = b_{i+1} - b_i$.

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:**

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

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 $[t, 1]$.

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

□ 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* $|b-a| \leq \epsilon$ *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.*

□ 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* $\lvert b-a\rvert > \epsilon$, *subdivide;*
• *If* $V(\mathbf{b}) = 0$, *remove the interval.*
• *If* $V(\mathbf{b}) = 1$, *output interval containing one and only one root.*
• *If* $\lvert b-a\rvert \leq \epsilon$ *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; $\epsilon = 0.000001$):

d	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

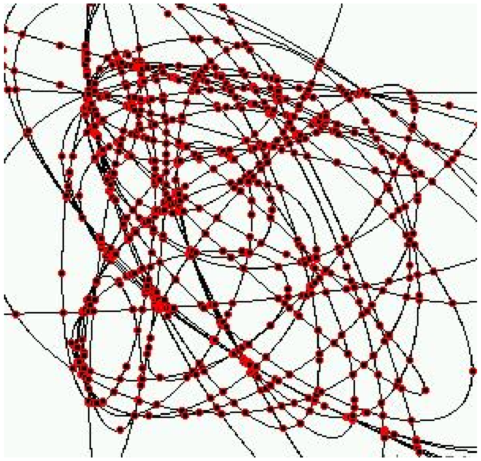
Equations per s. (precision bits vs. degree; $\epsilon = 0.000001$) using GMP library:

d	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`).

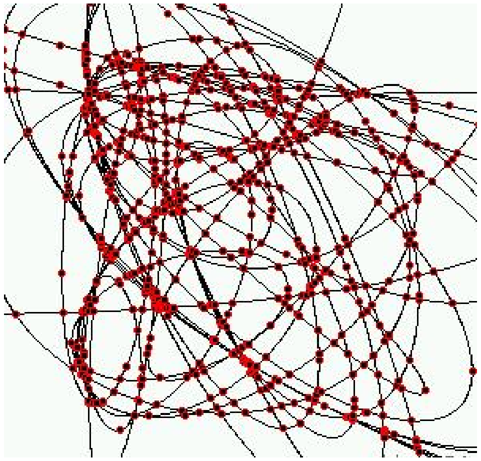
[Demo]

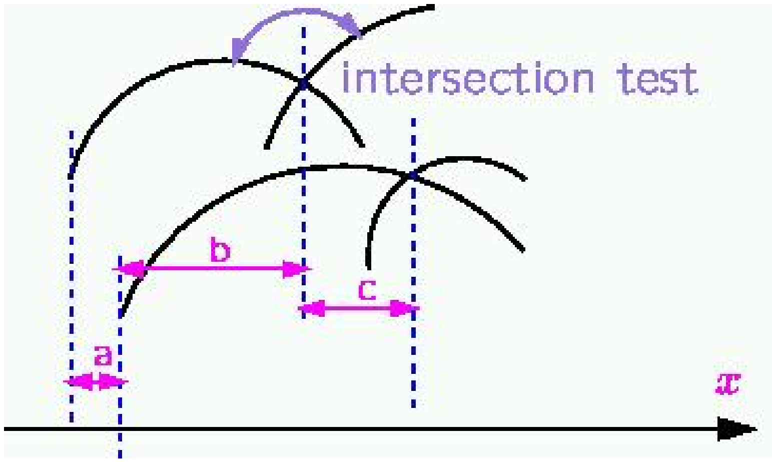
Sweeping a set of (conic) circular arcs



[DFMT00]

Sweeping a set of (conic) circular arcs





- Arcs represented by the equations of a circle and two lines.

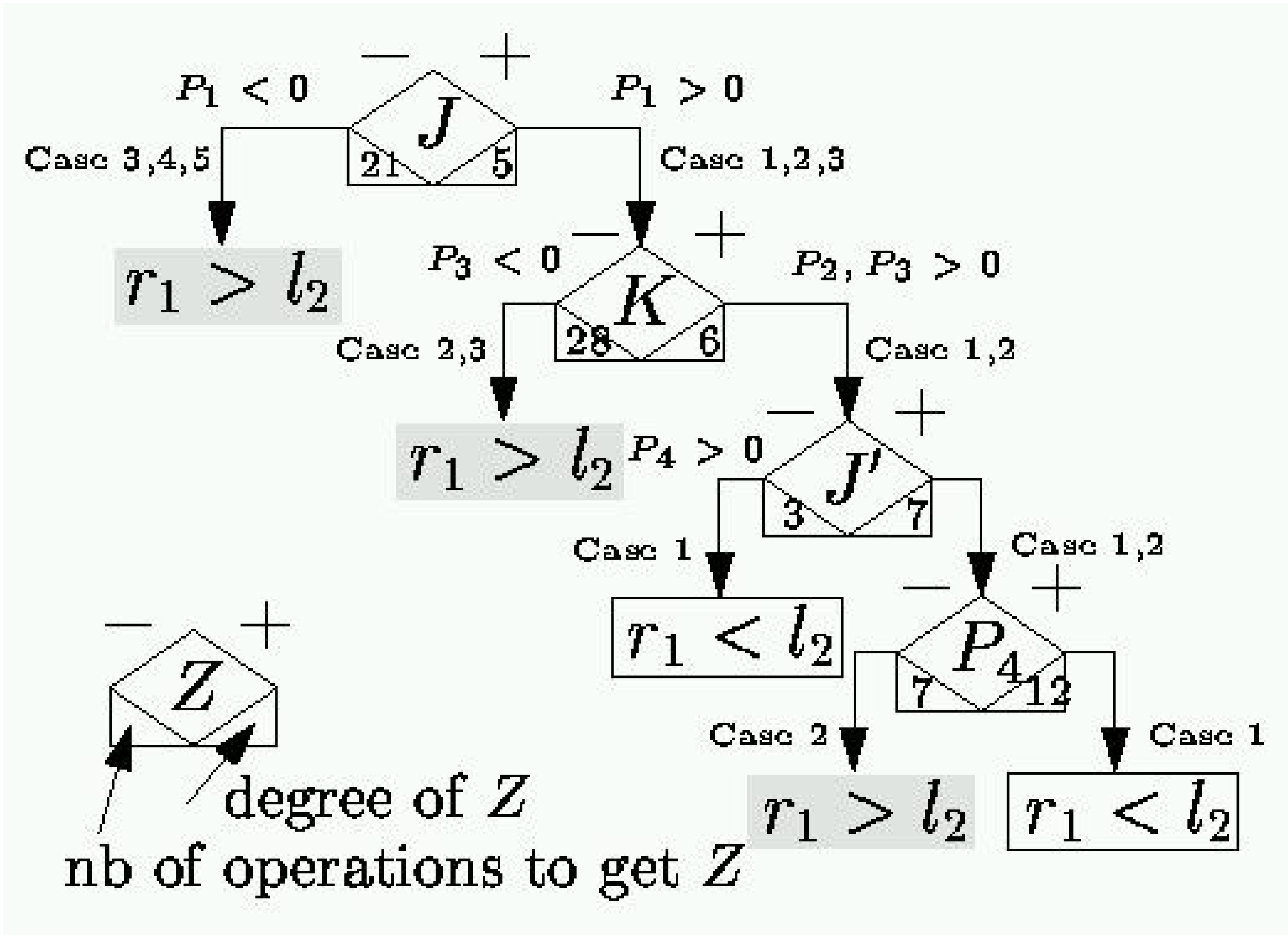
- By resultant computation, it reduces to sign evaluations of algebraic polynomials P_i (degree ≤ 12):

$$P_4 t^4 + P_3 t^3 + P_2 t^2 + P_1 t + P_0 = 0$$

- Inputs parameters in a bounded grid; static error bounds.

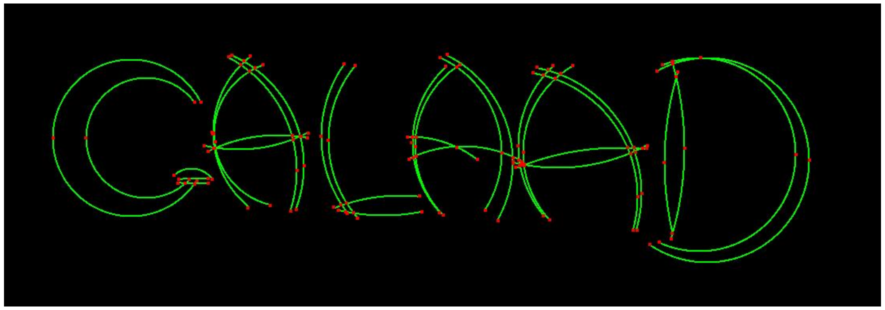
[DFMT00]

- Filtering technics can be used easily.
- Efficiency:



[DFMT00]

Preliminary benchmarks



A: 100, S:23123, V: 1916, E: 6878	Sweep (s)	Incr. (s)
SplCart, ExDFMT, double	0.6	0.7
RefCart, ExDFMT, double	0.8	0.2
SplCart, ExDFMT, ZZraw	9.3	11.0
SCN, ExDFMT, ZZraw	6.0	7.0
SCN, ExDFMT, Gmpz	9.0	6.0
RefCart, ExDFMT, ZZraw	8.2	9.3
SplCart, Naive, Core	23.2	34.7
RefCart, Naive, Core	24.9	36.3
SplCart, Naive, Leda	30.7	21.6
RefCart, Naive, Leda	30.8	21.9
SplCart, ExDFMT, Gmpq	51.7	60.5
RefCart, ExDFMT, Gmpq	57.5	61.2

A:Arcs, S:Seed, V:Vertices, E:Edges. Random data.

A: 400, S: 43123, V: 33653, E: 131402	Sweep (sec)	Incr. (sec)
SplCart, ExDFMT, double	3.5	0.6
RefCart, ExDFMT, double	5.0	1.3
SplCart, ExDFMT, ZZraw	179.5	174.9
RefCart, ExDFMT, ZZraw	156.9	147.2
SCN, ExDFMT, ZZraw	100.0	95.0
SCN, ExDFMT, Gmpz	140.0	25.0
SplCart, Naive, Core	Abort	Abort
RefCart, Naive, Core	Abort	Abort
SplCart, Naive, Leda	520.6	337.7
RefCart, Naive, Leda	523.6	345.6
SplCart, ExDFMT, Gmpq,	989.9	960.3
RefCart, ExDFMT, Gmpq,	1104.0	1009.3

Benchmarks

Times							Exactness	
Data		Naive		Polynomial			Naive	Pol.
		`double`	Interval + real	`double`	GMP	static + semi-static + Naive Int. + GMP	`double`	`double`
		μs	μs	μs	μs	μs	%	%
$l_1 \leq r_2?$	rnd22	0.60	2.48	0.25	82	0.36	100	100
	rnd16	0.60	2.48	0.25	78	0.41	100	100
	almost	0.60	67.	0.25	115	6.80	99	78
	degenerate	0.60	2170.	0.25	115	128.	8	17
$l_1 \leq l_2?$	rnd22	0.60	2.45	0.28	107	0.44	100	100
	rnd16	0.60	2.45	0.28	102	0.56	100	100
	almost	0.60	38.1	0.28	130	5.64	99	99
	degenerate	0.60	2180.	0.28	130	144.	7	5

- PC-Linux Pentium-III 500MHz, `g++` 2.95.1-02 `-mcpu=pentiumpro -march=pentiumpro`
- **rnda:** pick at random in $[-M, M]^2$, $M = 2^a$, the centers of two cercles and a common point. Arc defined by radical axis and first cercle.
- **degenerate:** same construction with ending points of same abcissae.
- **almost:** same construction with one square radius slightly incremented.

Points

Solvers

Solvers

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

  Newton like methods, Minimisation methods, Weierstrass method.

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.

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.

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.

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.

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.

Multiplication operators

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

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

Multiplication operators

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

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

Theorem:

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

Multiplication operators

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

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

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**

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

Multiplication operators

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

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

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)

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

Resultant in one variable

Let $f_0 = c_{0,0} + \cdots + c_{0,d_0}\, x^{d_0}$, $\; f_1 = c_{1,0} + \ldots + c_{1,d_1}\, x^{d_1}$ (with $d_0 \leq d_1$).

Resultant in one variable

Let $f_0 = c_{0,0} + \cdots + c_{0,d_0}\,x^{d_0}$, $\;f_1 = c_{1,0} + \ldots + c_{1,d_1}\,x^{d_1}$ (with $d_0 \leq d_1$).

Sylvester (1840)

$$\begin{array}{c} \overbrace{\begin{array}{cccccc} f_0 & \cdots & x^{d_1-1}f_0 & f_1 & \cdots & x^{d_0-1}f_1 \end{array}}^{d_0+d_1} \\ \left[\begin{array}{ccc|ccc} c_{0,0} & & \mathbf{0} & c_{1,0} & & \mathbf{0} \\ \vdots & \ddots & & \vdots & \ddots & \\ \vdots & & c_{0,0} & \vdots & & c_{1,0} \\ c_{0,d_0} & & \vdots & c_{1,d_1} & & \vdots \\ & \ddots & \vdots & & \ddots & \vdots \\ \mathbf{0} & & c_{0,d_0} & \mathbf{0} & & c_{1,d_1} \end{array}\right] \end{array} \qquad \left.\begin{array}{l} 1 \\ x \\ \cdot \\ x^{d_1-1} \\ \cdot \\ x^{d_0+d_1-1} \end{array}\right\} d_0+d_1$$

Resultant in one variable

Let $f_0 = c_{0,0} + \cdots + c_{0,d_0}\, x^{d_0}$, $\;f_1 = c_{1,0} + \ldots + c_{1,d_1}\, x^{d_1}$ (with $d_0 \le d_1$).

Sylvester (1840)

$$
\overbrace{\begin{array}{cccc|cccc} f_0 & \cdots & & x^{d_1-1} f_0 & f_1 & \cdots & & x^{d_0-1} f_1 \end{array}}^{d_0+d_1}
$$

$$
\left[\begin{array}{cccc|cccc}
c_{0,0} & & & \mathbf{0} & c_{1,0} & & & \mathbf{0} \\
\vdots & \ddots & & & \vdots & \ddots & & \\
\vdots & & & c_{0,0} & \vdots & & & c_{1,0} \\
c_{0,d_0} & & & \vdots & c_{1,d_1} & & & \vdots \\
 & \ddots & & \vdots & & \ddots & & \vdots \\
\mathbf{0} & & & c_{0,d_0} & \mathbf{0} & & & c_{1,d_1}
\end{array}\right]
\qquad
\left.\begin{array}{c} 1 \\ x \\ \cdot \\ x^{d_1-1} \\ \cdot \\ x^{d_0+d_1-1} \end{array}\right\} d_0 + d_1
$$

Bézout (1779)

$$
\Theta_{f_0,f_1}(x,y) := \frac{f_1(x)\, f_0(y) - f_1(y)\, f_0(x)}{y - x} = \sum_{i=0}^{d_1-1} \theta_{f_0,f_1,i}(x)\, y^i = \sum_{i=0}^{d_1-1} \sum_{j=0}^{d_1-1} \theta_{i,j}\, x^i\, y^j .
$$

The Bézout matrix is $\mathsf{B}_{f_0,f_1} = (\theta_{i,j})_{0 \le i,i \le d_1}$.

Resultant in one variable

Let $f_0 = c_{0,0} + \cdots + c_{0,d_0}\, x^{d_0}$, $\; f_1 = c_{1,0} + \ldots + c_{1,d_1}\, x^{d_1}$ (with $d_0 \leq d_1$).

Sylvester (1840)

$$
\begin{array}{c}
\overbrace{\begin{array}{cccccc} f_0 & \cdots & x^{d_1-1}f_0 & f_1 & \cdots & x^{d_0-1}f_1 \end{array}}^{d_0+d_1} \\
\left[\begin{array}{ccc|ccc}
c_{0,0} & & \mathbf{0} & c_{1,0} & & \mathbf{0} \\
\vdots & \ddots & & \vdots & \ddots & \\
\vdots & & c_{0,0} & \vdots & & c_{1,0} \\
c_{0,d_0} & & \vdots & c_{1,d_1} & & \vdots \\
& \ddots & \vdots & & \ddots & \vdots \\
\mathbf{0} & & c_{0,d_0} & \mathbf{0} & & c_{1,d_1}
\end{array}\right]
\end{array}
\qquad
\left.\begin{array}{l} 1 \\ x \\ . \\ x^{d_1-1} \\ . \\ x^{d_0+d_1-1} \end{array}\right\} d_0+d_1
$$

Bézout (1779)

$$
\Theta_{f_0,f_1}(x,y) := \frac{f_1(x)\, f_0(y) - f_1(y)\, f_0(x)}{y - x} = \sum_{i=0}^{d_1-1} \theta_{f_0,f_1,i}(x)\, y^i = \sum_{i=0}^{d_1-1} \sum_{j=0}^{d_1-1} \theta_{i,j}\, x^i\, y^j.
$$

The Bézout matrix is $\mathsf{B}_{f_0,f_1} = (\theta_{i,j})_{0 \leq i,i \leq d_1}$.

Theorem : $R(c_{i,j}) := det(S)$ **vanishes iff** $f_0 = 0, f_1 = 0$ **has a common root.**

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

1. Construct the resultant matrix $\mathtt{S}(x_n)$ of $f_1, \ldots, f_n$ as polynomials in $x_1, \ldots, x_{n-1}$ with coefficients in $\mathbb{K}[x_n]$.

2. Solve $\mathtt{S}(x_n)^t \mathbf{w} = 0$.

   - Either by solving $\det(\mathtt{S}(x_n)) = 0$ and by deducing the corresponding $\mathbf{w}$.
   - or by reducing it to an eigenproblem:

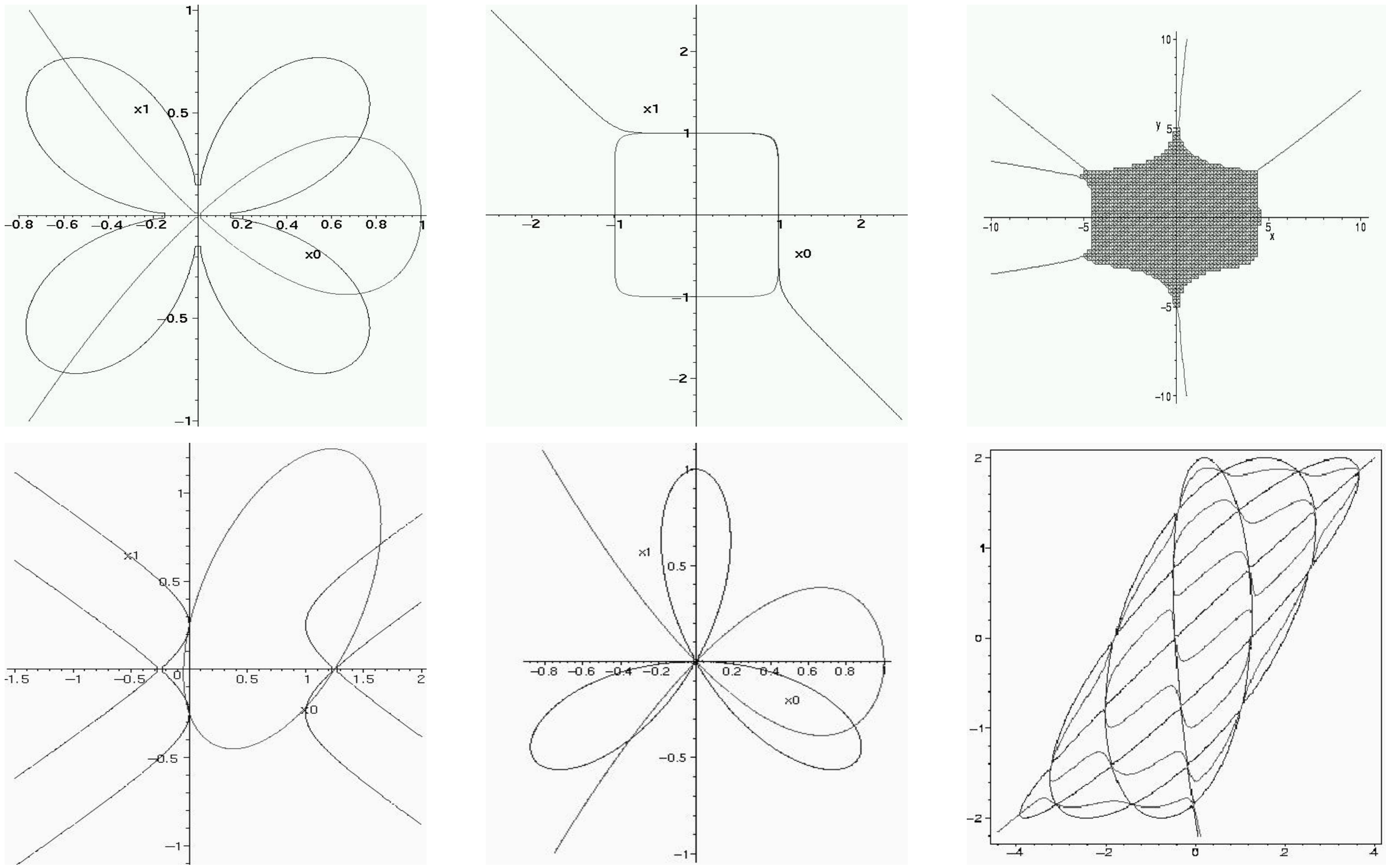
$$(\mathtt{S}_d^t\, x_n^d + \mathtt{S}_{d-1}^t x_n^{d-1} + \cdots + \mathtt{S}_0^t)\, \mathbf{w} = 0$$

   or

$$\left(\begin{bmatrix} 0 & \mathbb{I} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \mathbf{0} & \mathbb{I} \\ \mathtt{S}_0^t & \cdots & \mathtt{S}_{d-2}^t & \mathtt{S}_{d-1}^t \end{bmatrix} - x_n \begin{bmatrix} \mathtt{S}_d^t & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathtt{S}_d^t & 0 \\ 0 & \cdots & 0 & \mathtt{S}_d^t \end{bmatrix} \right) \underline{\mathbf{w}} = 0,$$

3. Deduce the other coordinates of the roots from $\mathbf{w}$.

Comparison



Alg. mth.	# r-roots	deg.	# x-roots	Time
1	2	4	2	0.02 s
2	6	3	6	0.02 s
3	6	3	6	0.02 s
4	4	2	4	0.02 s
5	16	4	8	0.02 s
6	4	2	4	0.02 s
7	8	4	8	0.02 s
8	6	3	0	–
9	8	4	8	0.03 s
10	12	6	12	0.03 s
11	10	10	10	0.12 s
12	16	6	0	–
13	6	5	6	0.02 s
14	42	6	42	0.12 s

Sylv. SVD	# r-roots	deg.	# x-roots	Time
1	2	4	2	0.00 s
2	6	3	4	0.00 s
3	6	3	1	0.00 s
4	4	2	2	0.00 s
5	16	4	0	–
6	4	2	4	0.00 s
7	8	4	3	0.00 s
8	6	3	2	0.00 s
9	8	4	Np	–
10	12	6	Np	–
11	10	10	Np	–
12	16	6	10	0.07 s
13	6	5	2	0.00 s
14	42	6		

Sylv. Eig.	# r-roots	deg.	# x-roots	Time
1	2	4	Np	–
2	6	3	6	0.00 s
3	6	3	4	0.00 s
4	4	2	4	0.00 s
5	16	4	16	0.01 s
6	4	2	Np	–
7	8	4	2	0.00 s
8	6	3	0	0.00 s
9	8	4	Np	–
10	12	6	Np	–
11	10	10	3	0.82 s
12	16	6	Np	–
13	6	5	1	0.00 s
14	42	6	Np	–

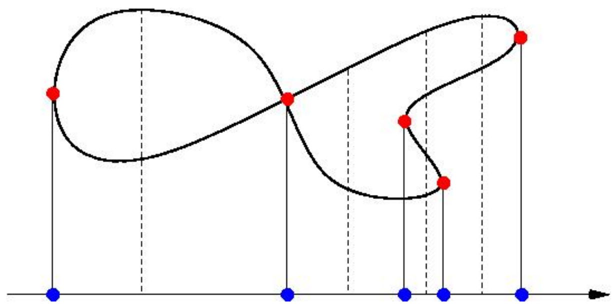
Bez. Eig.	# r-roots	deg.	# x-roots	Time
1	2	4	Np	–
2	6	3	Uc	–
3	6	3	3	0.00 s
4	4	2	Np	–
5	16	4	Np	–
6	4	2	Np	–
7	8	4	1	0.01 s
8	6	3	2	0.01 s
9	8	4	6	0.00 s
10	12	6	11	0.00 s
11	10	10	6	0.19 s
12	16	6	Np	–
13	6	5	2	0.00 s
14	42	6	Np	–

- **Subdvision**

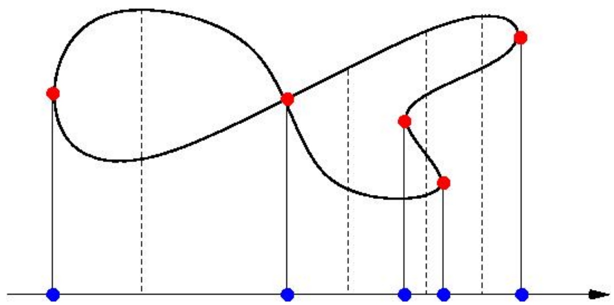
Example	ε	Evaluation	Number of intervals	Time	Number of real roots
10	10^{-5}	10^{-5}	5	0.030	5
11	10^{-5}	10^{-4}	770	79.188	2
15	10^{-5}	10^{-4}	4	0.016	4

Curves

Curves



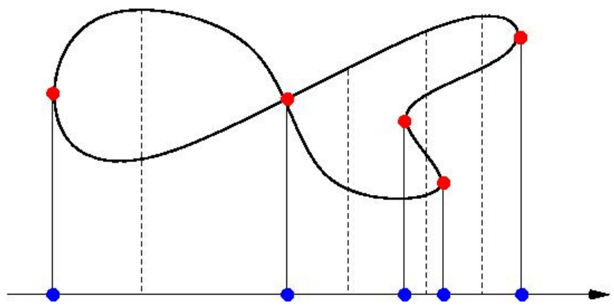
Curves



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.*

Curves



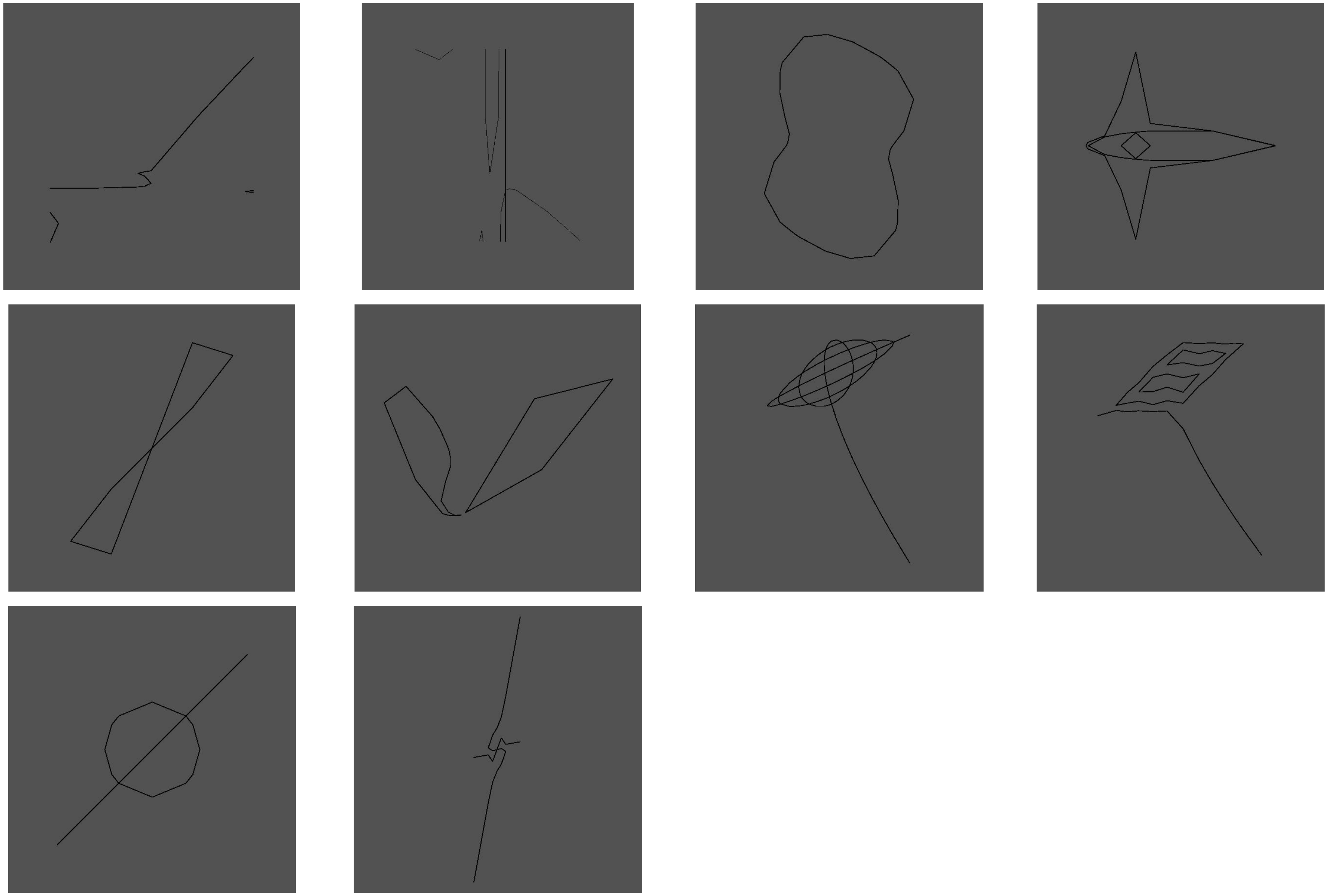
Algorithm: Topology of an implicit curve

- *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.

Examples



topology	time execution running
1	0.03 s
2	0.04 s
3	0.04 s
4	0.10 s
5	0.05 s
6	0.05 s
7	0.62 s
8	0.17 s
9	0.02 s
10	0.09 s

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

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

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**.

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*.

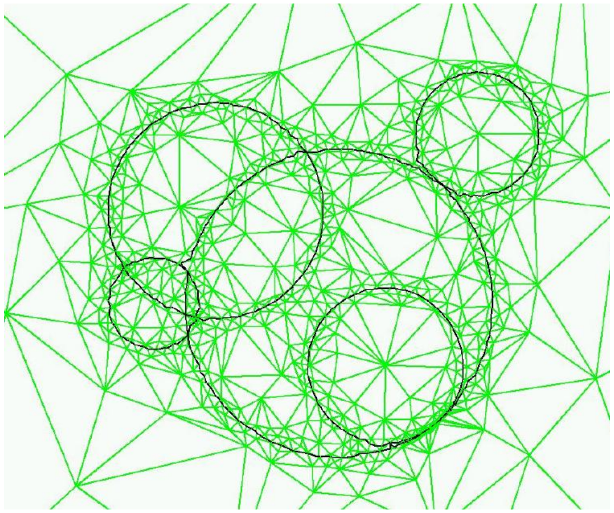
- No specific directions/axes used.

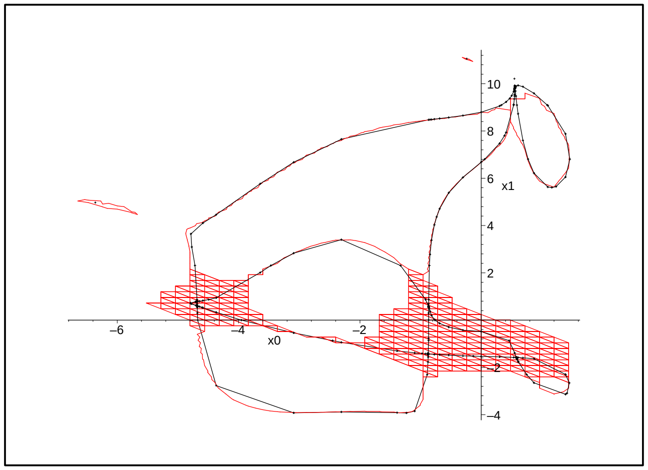
- New edges are constructed, no tangency problem.

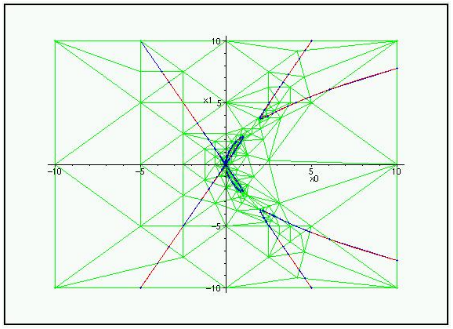
- Number of triangles related to the complexe local feature size.

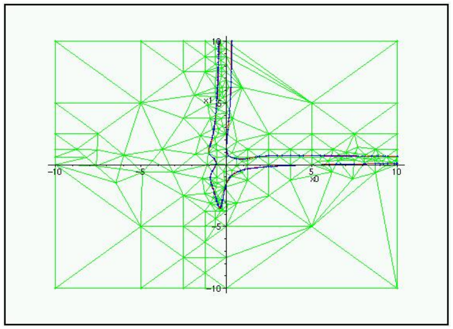
- Application to the intersection of curves, surfaces.

Examples



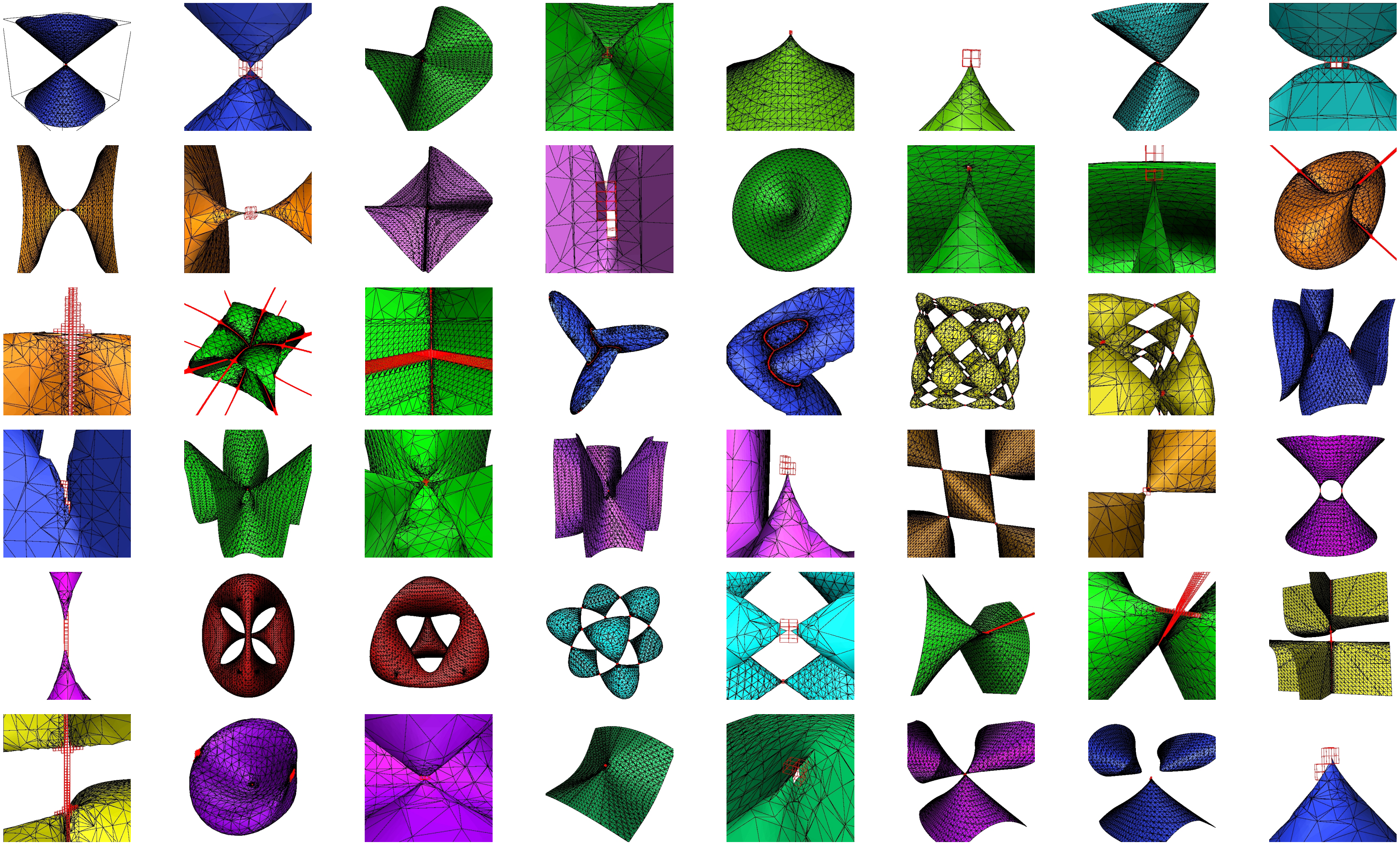


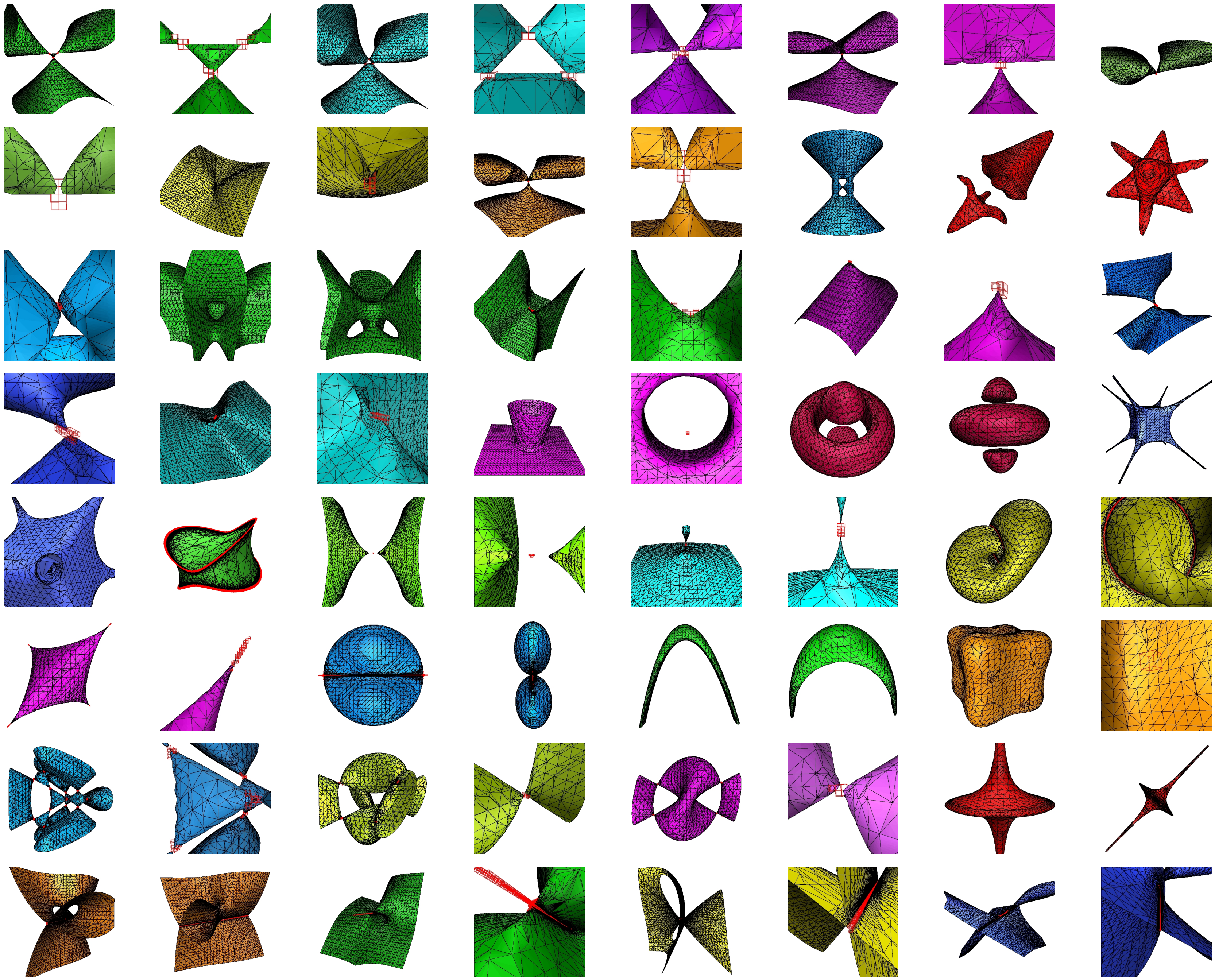


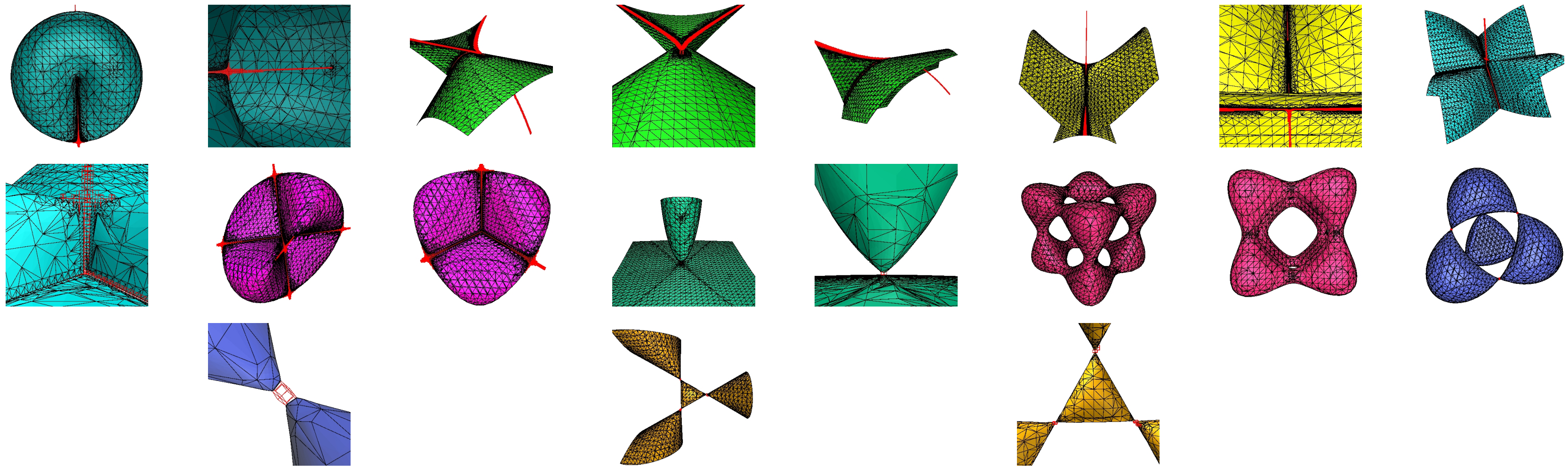


Surfaces

Examples







	nb triangles	nb cellules	temps d'execution
A1 - -	8093	3962 + 6 nt	3.37818 s
A2 + -	6395	3134 + 6 nt	2.90356 s
A2 - -	3598	1768+ 7 nt	1.9934 s
A3 + -	6179	3009 + 10 nt	3.01165 s
A3 - -	6290	6290 + 6 nt	3.14654 s
A4 + -	5859	2825 + 13 nt	3.14721 s
D4 + -	4035	1975 + 6 nt	1.84261 s
D4 - -	8937	4340 + 12 nt	4.29918 s
D4 - - unfold 1	7291	3577 + 10 nt	3.53391 s
D4 - - unfold 2	8269	4019 + 31 nt	3.86696 s
D4 – unfold 3	9633	4615 + 14 nt	4.35244 s

	nb triangles	nb cellules	temps d'execution
D5 + -	7407	3596 + 6 nt	3.61756 s
D5 - -	4254	2014 + 9 nt	2.23405 s
D6 + -	3925	1882 +12 nt	2.1853 s
D6 - -	6983	3364 + 12 nt	3.83701 s
E6 + -	6017	2920 + 18 nt	3.53177 s
E6 - -	3638	1751 + 10 nt	2.11355 s
E7	6019	2927 + 24 nt	3.20513 s
E8	4524	2182 + 14 nt	2.78838 s
Q1	11634	4760 + 185 nt	9.04296 s
Q2	8833	3821 + 84 nt	7.30251 s
Q3	6661	3074 + 24 nt	6.55459 s
S20	8016	3498 + 258 nt	2.9955 s
S21	13542	4848 + 1962 nt	4.16841 s
S22	16722	6234 + 792 nt	5.3028 s
S23	17757	6551 + 1263 nt	5.28855 s
S26	8699	3310 + 1178 nt	4.02468 s

	nb triangles	nb cellules	temps d'execution
D5 + -	7407	3596 + 6 nt	3.61756 s
D5 - -	4254	2014 + 9 nt	2.23405 s
D6 + -	3925	1882 +12 nt	2.1853 s
D6 - -	6983	3364 + 12 nt	3.83701 s
E6 + -	6017	2920 + 18 nt	3.53177 s
E6 - -	3638	1751 + 10 nt	2.11355 s
E7	6019	2927 + 24 nt	3.20513 s
E8	4524	2182 + 14 nt	2.78838 s
Q1	11634	4760 + 185 nt	9.04296 s
Q2	8833	3821 + 84 nt	7.30251 s
Q3	6661	3074 + 24 nt	6.55459 s
S20	8016	3498 + 258 nt	2.9955 s
S21	13542	4848 + 1962 nt	4.16841 s
S22	16722	6234 + 792 nt	5.3028 s
S23	17757	6551 + 1263 nt	5.28855 s
S26	8699	3310 + 1178 nt	4.02468 s

Thanks for your attention.