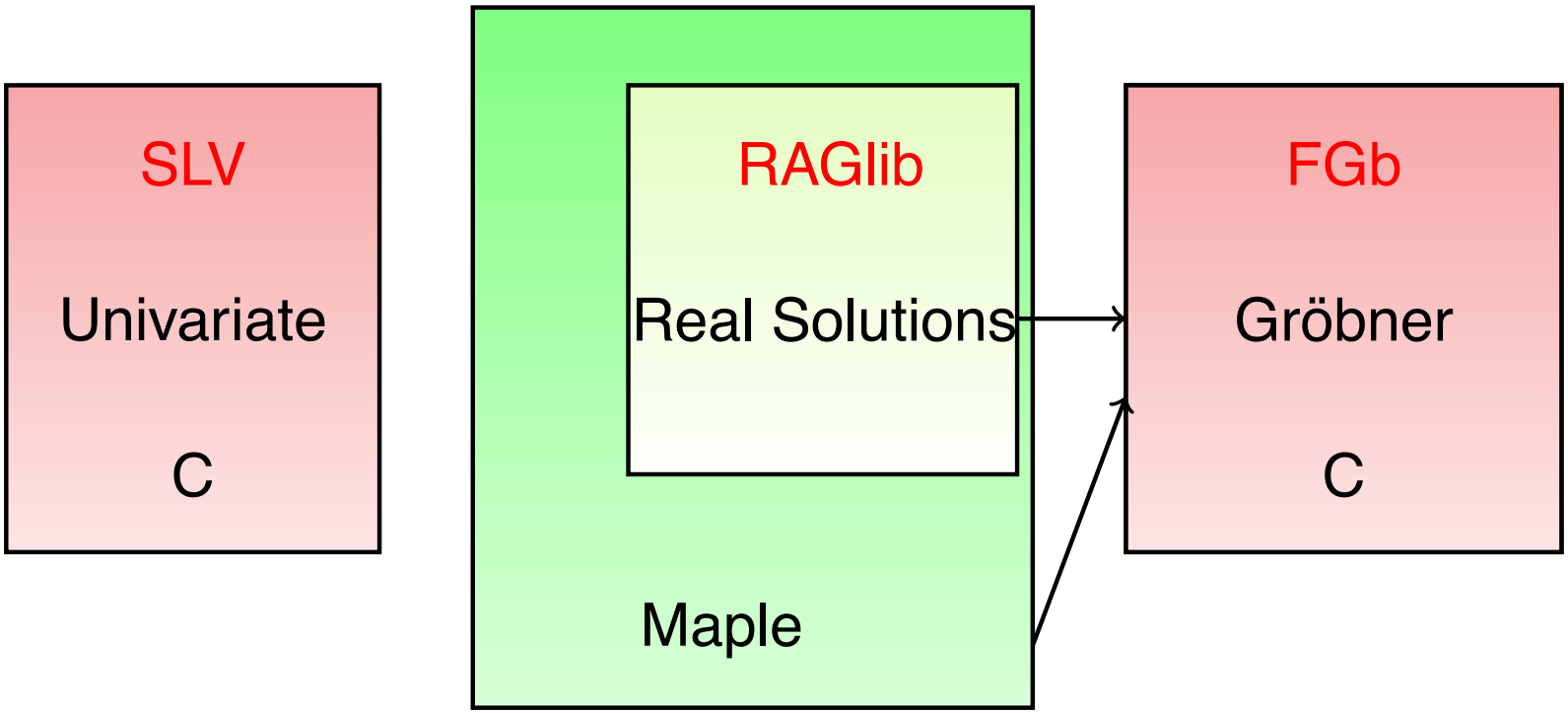
Software Development @ PolSys

Polynomial System Solving

Solving efficiently polynomial system of equations is a fundamental problem in Computer Algebra with many applications.

- Zero-Dimensional Systems / Gröbner Bases: FGb
- Positive Dimension: RAGLIB solutions over the reals
- SLV: new C library for isolating and refining the real roots of univariate polynomials.



Outline

Outline

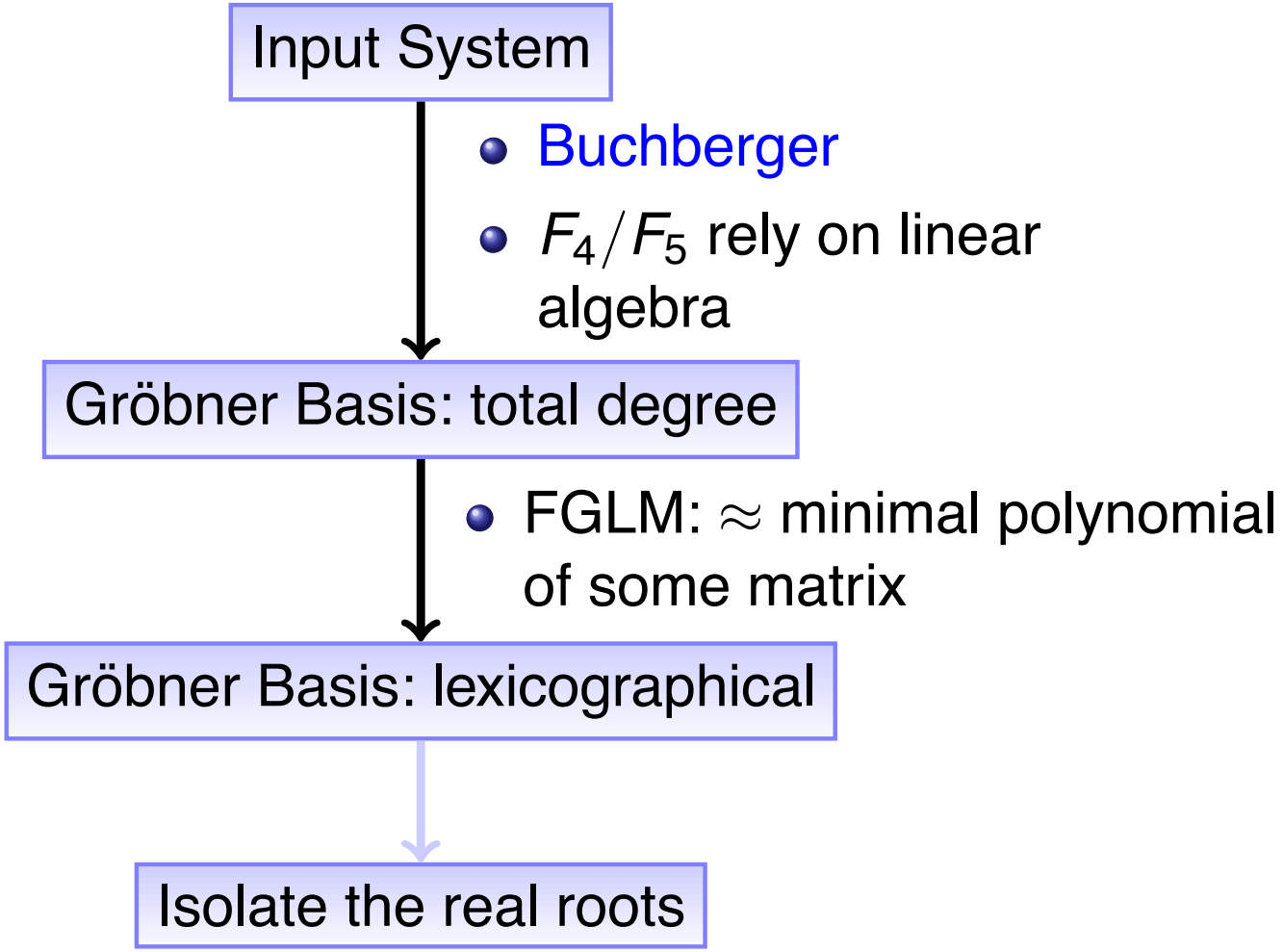
FGb: Goal and architecture of the library

Purpose of the `FGb` library :

- efficient implementations of state-of-the-art algorithms for computing Gröbner bases.
  Mandatory to demonstrate the *practical efficiency* of new algorithms.

- In conjunction with other software, the FGb library has been used in various applications (Robotics, Signal Theory, Biology, Computational Geometry, . . .).
  Wide range of problems in Cryptology (`FGb` was explicitly used to break several cryptosystems or to find security parameters).

Algorithms . . . relying on linear algebra

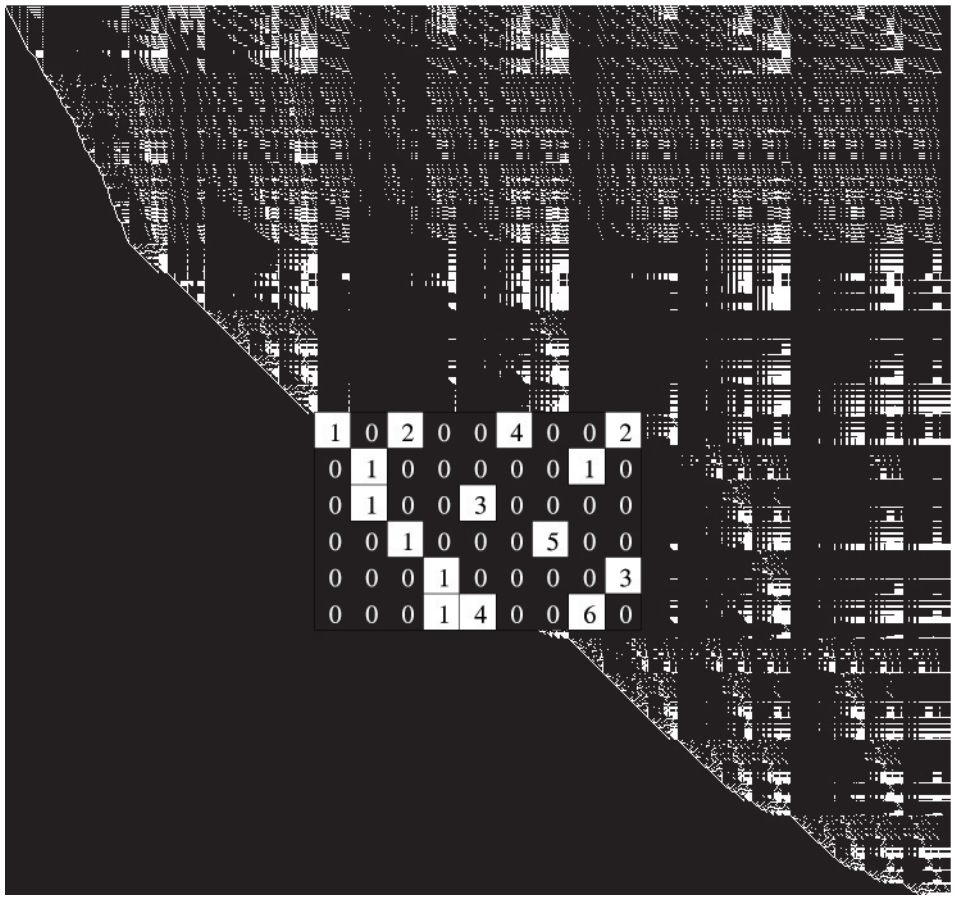


Main cost of the computation

Gröbner basis computation:

- sequences of Matrix Generation+Gaussian eliminations
- main cost of the whole global computation : Linear Algebra

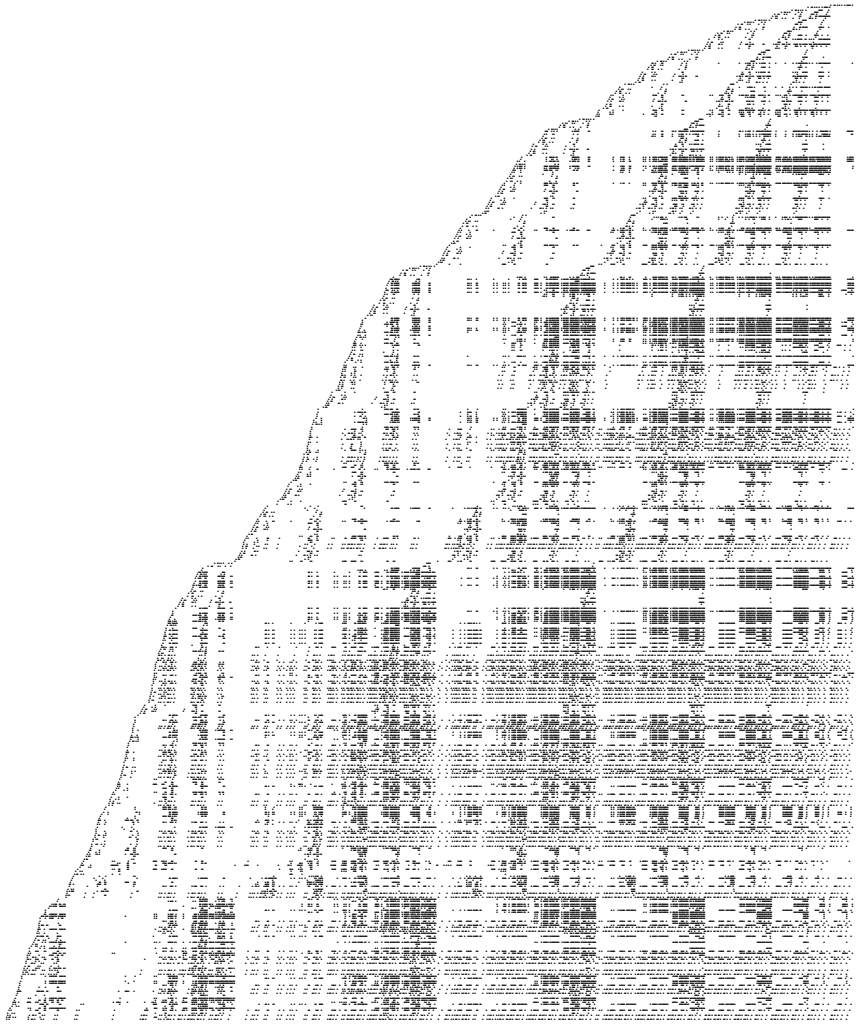
Katsura 7 in $\mathbb{F}_{65521}$: 694×738 matrix of density 8%



Main cost of the computation

Gröbner basis computation:

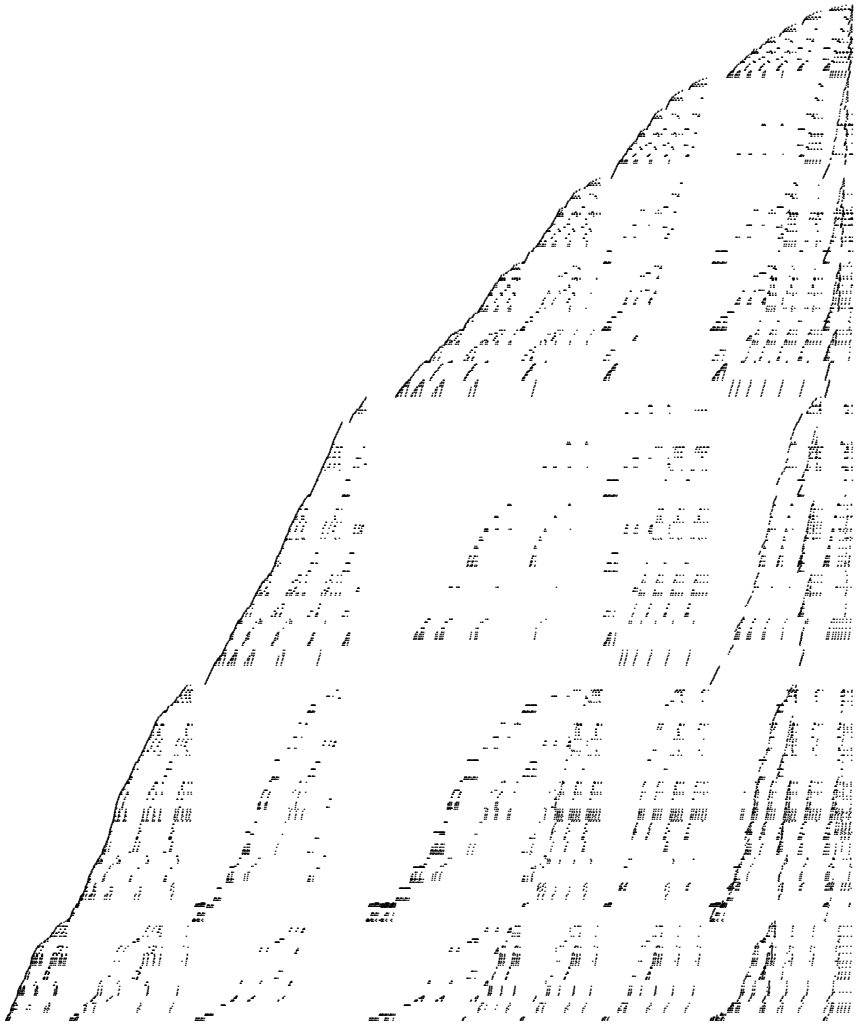
- sequences of Matrix Generation+Gaussian eliminations
- main cost of the whole global computation : Linear Algebra



Main cost of the computation

Gröbner basis computation:

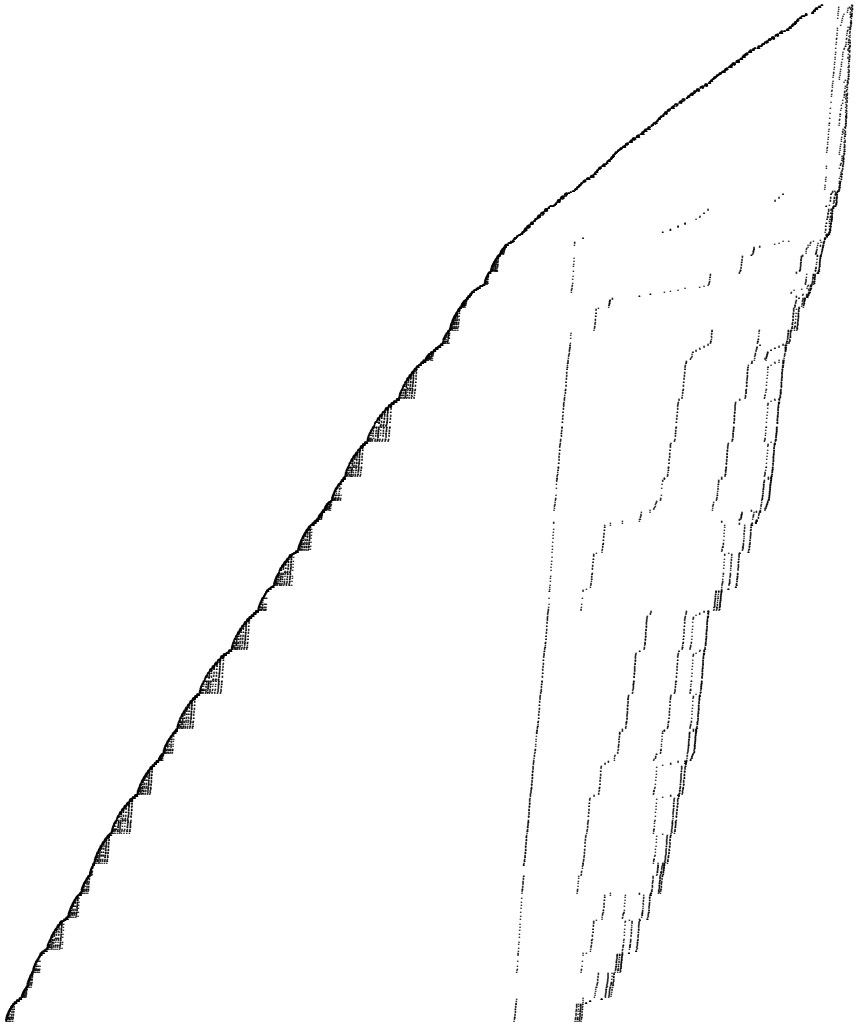
- sequences of Matrix Generation+Gaussian eliminations
- main cost of the whole global computation : Linear Algebra



Main cost of the computation

Gröbner basis computation:

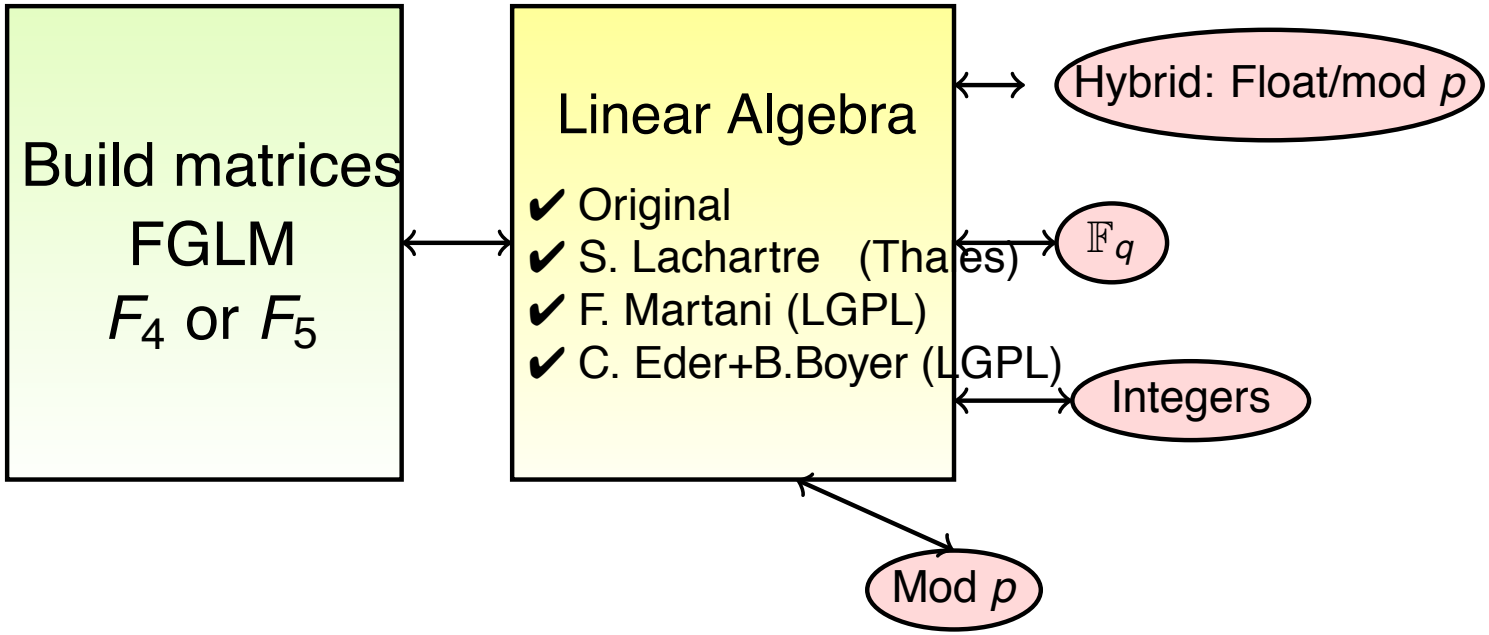
- sequences of Matrix Generation+Gaussian eliminations
- main cost of the whole global computation : Linear Algebra



FGb

library (216 000 lines of C code) (F_4, matrix-F_5, F_5,...).

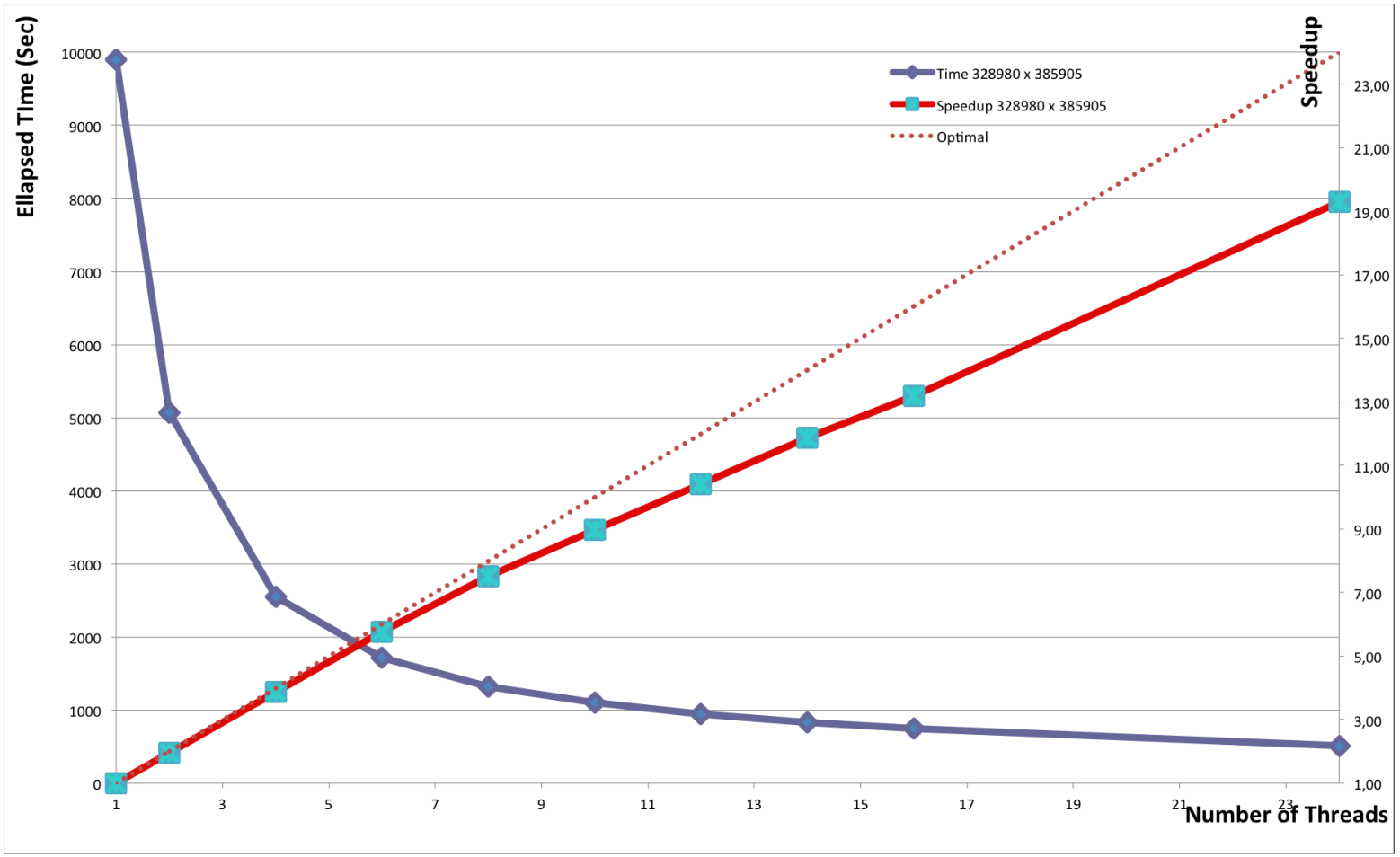
☞ Critical part in `FGb`: linear algebra package.



New Linear Algebra Library (with C. Eder and B. Boyer)

- LGPL
- Multi-core implementations $\approx$ 32/64 cores
- HPAC project
- cache memory : block operations
- data structure adapted for both matrix types: hybrid structures

New Linear Algebra Library



Katsura 16 - mat9 - Faugère/Lachartre/Martani

New radical function (zero-dimensional system)

`fgb_matrixn_radical`

or

`fgb_matrixn_radical2` for computing Critical Points [Safey El Din]

I_1 ideal in $\mathbb{K}[\mathbf{x}, \mathbf{y}]$

I_2 ideal in $\mathbb{K}[\mathbf{y}]$

Compute: $\sqrt{(I_1 \cap \mathbb{K}[\mathbf{y}]) \cap I_2}$

1. Compute G_1 a Gröbner basis for a elimination ordering (and eliminate some variables)
2. Compute G_2 a second Gröbner basis of $G_1 + F$
3. Apply the Sparse-FGLM to G_2 and take the radical
4. Output the radical of $I \longrightarrow$ Shape Position

Benchmark [Bannwarth3]

degree $356 \longrightarrow 80$

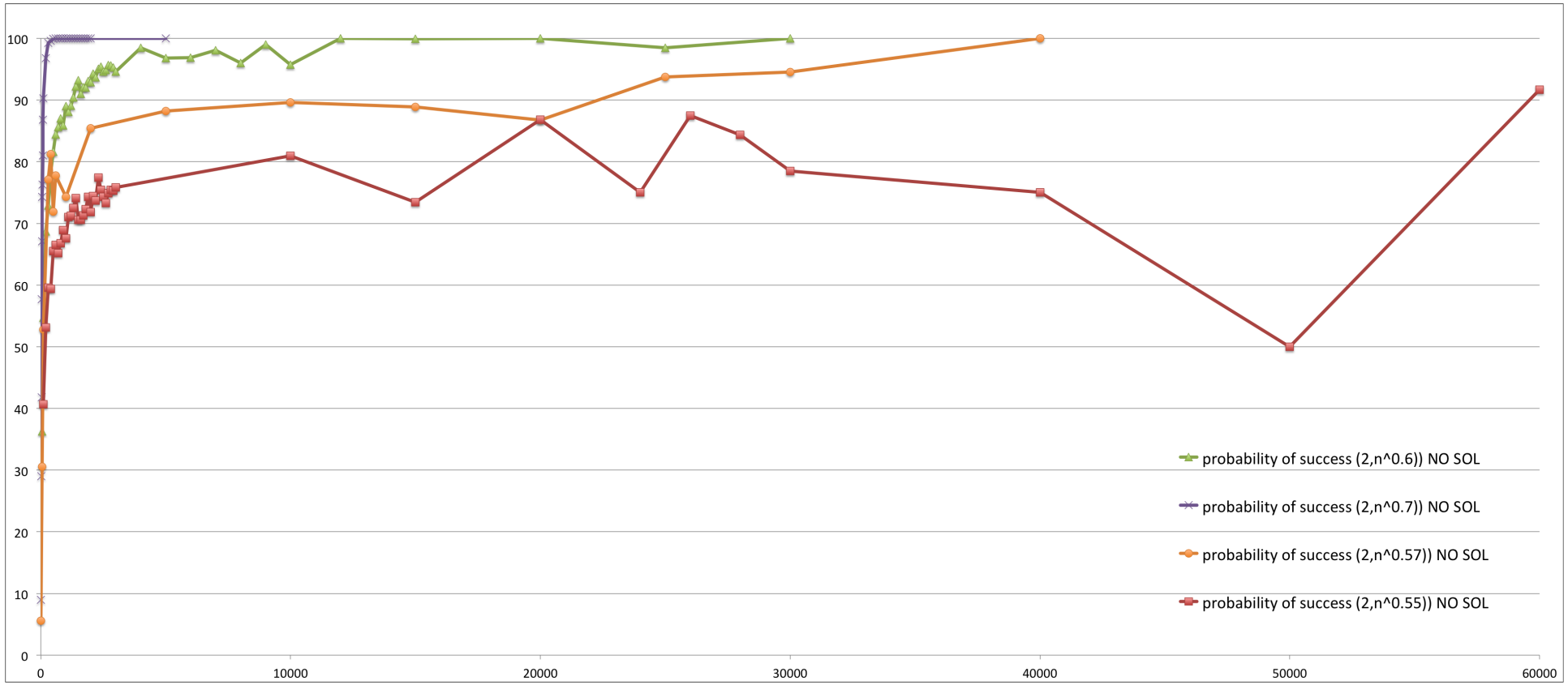
size of the integers: $2^{10093} \longrightarrow 2^{2277}$

Fewnomials

On the Complexity of Quadratic Fewnomial Systems . . .
with PJ Spaenlehauer and J Svartz

$$f_i(x_1,\ldots,x_n) = \sum_{\alpha\in S} c_\alpha \mathbf{x}^\alpha \text{ where } S \text{ of size } n+k$$

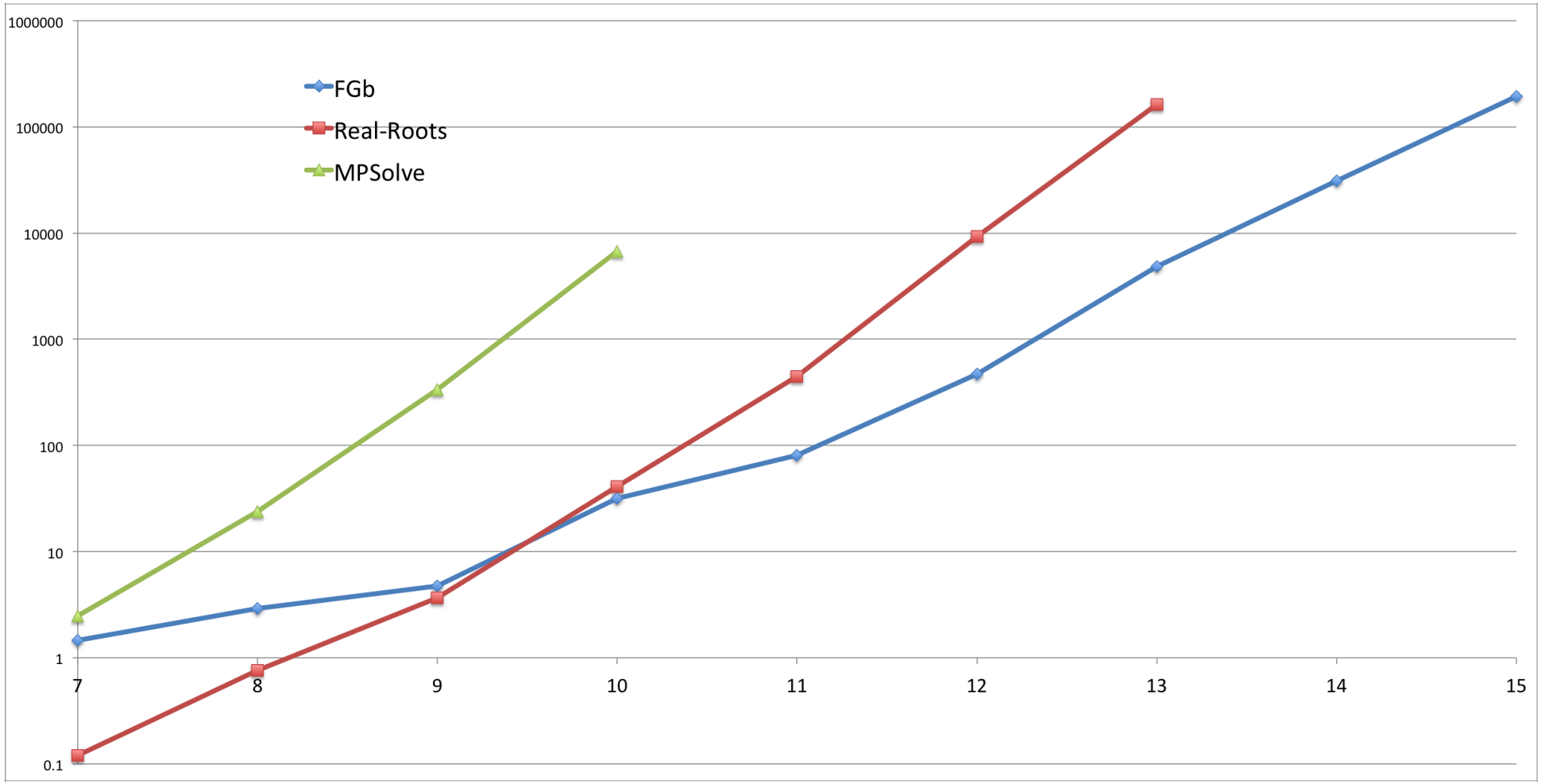
Probabilistic polynomial time algorithm - Link with graph theory - Very efficient ($n = 60000$).



Reduction to univariate polynomials

Zero-dim system: very efficient multi-modular + hensel lifting

Find real roots: compute a (lexico) GB and the isolate the real roots.



Outline

Outline

What can we compute over the reals?

$$F_1 = \cdots = F_p = 0, \quad G_1 > 0, \ldots, G_s > 0$$

Infinite number of complex solutions

- Decide the existence of real solutions of the input system
- Compute real solutions with geometric informations (at least one in each path connected information)

RAGlib (release 3.23, May 2014)

What is it?

- a small library written in Maple. $\sim$ 14 000 lines and 3 functions exported (for basic users) only.
- first release in 2004, initially: compilation of maple programs.

To do what?

- solving positive dimensional systems over the reals $\rightarrow$ applications.
- infinity of solutions $\rightarrow$ no possible enumeration of solutions
  $\rightsquigarrow$ what is the meaning of solving?

What are the features/goals?

- proof-of-concept of one of the most basic idea in algorithmics:
  improve practical efficiency by implementing theoretically fast algorithms
- research tool
- tackle applications unreachable by traditional methods/software.

Basic objects

$$F_1 = \cdots = F_p = 0, G_1 > 0, \ldots, G_s > 0$$

in $\mathbb{Q}[X_1, \ldots, X_n]$.

3 **functions:**

- `HasRealRoots`(sys);
  - returns [] iff Solutions(sys) $= \emptyset$ else returns a point in Solutions(sys)
- `PointsPerComponents`(sys);
  - returns a list of points meeting each connected component of Solutions(sys)

Basic objects

$$F_1 = \cdots = F_p = 0, G_1 > 0, \ldots, G_s > 0$$

in $\mathbb{Q}[X_1, \ldots, X_n]$.

3 **functions:**

> `HasRealRoots`(sys);

- returns [] iff Solutions(sys) $= \emptyset$ else returns a point in Solutions(sys)

> `PointsPerComponents`(sys);

- returns a list of points meeting each connected component of Solutions(sys)

> New! `UnivariateSumOfSquaresDec`(f);

♯ Warning: here f is assumed to be univariate

- decides if there exists $f_1, \ldots, f_s \in \mathbb{R}[X]$ such that $f = f_1^2 + \cdots + f_s^2$; computes such a decomposition when it exists (Schweighofer's algo.)

Software components on which RAGLIB is built on

MAPLE: arithmetic operations/manipulations on multivariate polynomials, and basic data-structures (lists, sets, etc.), factorization, linear algebra (matrices with polynomial entries).

Solving Zero-Dimensional Systems:

$$\boldsymbol{F} = 0 \rightarrow q(T) = 0, (X_i = q_i(T)/q_0(T))_{1 \leq i \leq n} \rightarrow \text{ Real root isolation}$$

Computing Rational Parametrizations:

$$\boldsymbol{F} = 0 \rightarrow \text{GROBNERBASIS}(\boldsymbol{F}) \rightarrow (X_i = q_i(T)/q_0(T))_{1 \leq i \leq n}$$

- FGB (Faugère, implemented in C) $\rightarrow$ Gröbner basis
  row echelon forms of matrices, multi-modular computations

On the difficulty of solving

$$F_1 = \cdots = F_p = 0, G_1 > 0, \ldots, G_s > 0$$

These problems are exponential in the number of variables !

Some additional remarks:

- Complexity $= s^n D^{O(n)}$
- Quadratic case $\rightsquigarrow s^n \mathrm{poly}(n)$.
- Number of inequalities plays an important role, increasing the number of inequalities does not help.

On the difficulty of solving

$$F_1 = \cdots = F_p = 0, G_1 > 0, \ldots, G_s > 0$$

These problems are exponential in the number of variables !

Some additional remarks:

- Complexity $= s^n D^{O(n)}$
- Quadratic case $\rightsquigarrow s^n \text{poly}(n)$.
- Number of inequalities plays an important role, increasing the number of inequalities does not help.

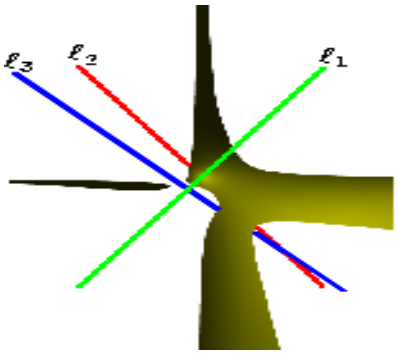
Some History

- Tarski, $\simeq 1930's \rightarrow$ complexity not elementary recursive
- Collins, $\simeq 1970's \rightarrow$ complexity doubly exponential in # variables
- Grigoriev/Vorobjov $\simeq 1988 \rightarrow$ complexity singly exponential in the # variables, but not practically efficient

  $\rightarrow$ Tricky use of Gröbner bases to yield practical efficiency

An application in Computational Geometry

Topological invariance of Voronoi diagrams of 3 lines in $\mathbb{R}^3$?



- Use of geometric properties to model the problem
  Points are equidistance to the 3 lines
- reduced to solve a single inequality (existence of real solutions) obtained as the discriminant of a characteristic polynomial
  (4 variables, degree 18, 253 terms)
- Solved within 5 minutes.

Conclusion: there is no change of topology of the Voronoi diagram when the lines move

$\rightsquigarrow$ potential use of this property to accelerate computations

$$-4\,a^8\alpha^3 y\beta\,x^3 - 56\,a^6\alpha^3 y^3\beta\,x + 28\,x^2\beta^4 y a^5\alpha + 28\,x^2 y\beta^2 a^5\alpha - 16\,a^{10} y\beta\,x\alpha - 32\,a^7 x^2\beta^2 y\alpha^3 + 24\,x\alpha\,a^6 y\beta - 28\,a^8\alpha^3 xy\beta + 16\,a^3 x^2\beta^4 y\alpha + 28\,a^3 x^2 y\beta^2\alpha - 32\,a^3 xy^2\alpha^2\beta - 32\,x^2 y\beta^2 a^9\alpha - 40\,a^7 x^2\beta^2 y\alpha - 48\,a^8\alpha\,\beta^3 y x^3 + 16\,y^2\alpha^4 x a^9\beta + 28\,xy^2 a^9\alpha^2\beta - 20\,a^3 xy^2\beta^3\alpha^2 - 32\,xy^2\beta^3 a^5\alpha^2 - 24\,\beta^3 yx\alpha\,a^2 - 16\,y\beta\,x\alpha\,a^2 - 20\,a^7 x^2\beta^2\alpha\,y^3 - 20\,x^2 y\beta^2\alpha^3 a^9 + 16\,a^7\alpha^2 y^4\beta\,x - 32\,a^7\alpha^2 y^2\beta\,x^3 - 28\,a^8\alpha^3 y^3\beta\,x - 24\,a^{10}\alpha^3 y\beta\,x + 28\,y^4 a^5 x\alpha^2\beta + 54\,\beta^2 x^2 a^4 y^2\alpha^2 + 80\,y^2\alpha^2 a^6 x^2\beta^2 - 4\,\beta^3 a^6\alpha^3 xy - 2\,\beta^5 yx\alpha\,a^2 - 28\,y^3\beta\,a^6 x\alpha - 12\,\beta^3 a^6 x\alpha\,y - 28\,a^4\beta^3 x\alpha\,y - 12\,a^6\alpha^3 y\beta\,x - 2\,a^{10}\alpha^5 xy\beta - 2\,y^5\beta\,x\alpha\,a^6 + 54\,a^8\alpha^2 x^2\beta^2 y^2 + 28\,y^2\alpha^4 x a^7\beta - 40\,a^5 y^2 x\alpha^2\beta + 28\,xy^2 a^7\alpha^2\beta - 12\,x^3 a^8\alpha\,y\beta - 24\,y^3\beta\,a^8 x\alpha - 28\,a^4\alpha\,\beta^3 x^3 y - 24\,x^3 a^4\alpha\,y\beta - 48\,a^4\alpha^3 y^3\beta\,x - 4\,y^3\beta\,a^6 x^3\alpha - 56\,a^6 x^3\beta^3\alpha\,y + 28\,a^7\beta^2 x^4\alpha\,y - 12\,y^3\beta\,a^4 x\alpha - 28\,a^6 x^3\alpha\,y\beta - 2\,y\beta\,a^6 x^5\alpha - 32\,a^5\beta^2 x^2\alpha\,y^3 - 20\,a^5\alpha^2 y^2\beta\,x^3 + 16\,a^5\beta^2 x^4\alpha\,y - 4\,a^4\alpha\,\beta^3 y^3 x + 32\,\beta^2 x^2\alpha^2 a^{10} + 2\,a^9\alpha^2 x^3\beta + 2\,xy^2\beta^3 a^5 + 32\,y^2\alpha^2 a^4 x^2 + 44\,y^2\alpha^2 a^6 x^2 + 44\,a^4\beta^2 y^2\alpha^2 + 32\,\beta^2 y^2\alpha^2 a^2 + 22\,a^6\alpha^2 x^2\beta^2 + x^4 a^6\alpha^2 y^2 + 22\,y^2\alpha^2 a^6\beta^2 + 10\,a^3 xy^2\beta^3 + 8\,\alpha^2\beta\,xa^{11} + 10\,x^3 a^7\alpha^2\beta + 8\,xy^2 a^9\beta + 8\,x^2 ya^9\alpha + 12\,\beta^2 y^3\alpha^3 a^5 + 12\,\beta^3 x^3\alpha^2 a^7 + 10\,x^2 ya^9\alpha^3 + 2\,x^2 ya^7\alpha^3 + 10\,y^3 a^5\alpha\,\beta^2 + 2\,y^3 a^3\alpha\,\beta^2 + 14\,\alpha^4\beta\,a^9 x + 10\,\alpha^4\beta\,a^{11} x + 2\,a^8\alpha^2 x^4\beta^2 + 8\,\beta^2 y^3\alpha^3 a^3 + 12\,y^3\alpha^3 a^7 x^2 + y^4 a^6 x^2\beta^2 + 8\,a^3 xy^2\beta + 8\,a^3 x^2 y\alpha - 4\,a^5 y^2 x\beta + 22\,\beta^2 x^2 a^4 y^2 + 44\,y^2 a^6 x^2\beta^2 - 4\,x^2 ya^7\alpha + 2\,a^6\alpha^4\beta^2 y^2 + a^2\alpha^2\beta^4 y^2 + 2\,a^6\alpha^2\beta^4 x^2 + 44\,a^8\alpha^2 x^2\beta^2 + 10\,\alpha^2\beta^3 a^5 x + 2\,\alpha^3\beta^2 a^5 y + 2\,\beta^4 y^2 a^4 x^2 - 4\,\alpha\,\beta^2 a^5 y + 8\,\alpha^2\beta\,a^5 x + 14\,\alpha\,\beta^4 a^3 y + 2\,\alpha^2\beta^3 a^7 x + 10\,\alpha^3\beta^2 a^7 y + 8\,\alpha\,\beta^2 a^7 y - 4\,\alpha^2\beta\,a^7 x + 10\,a\alpha\,\beta^4 y + 8\,a\alpha\,\beta^2 y + 32\,a^8 x^2 y^2\beta^2 + \beta^2 a^{10}\alpha^4 x^2 + 2\,y^4\beta^2\alpha^2 a^4 + 22\,a^8 x^2 y^2\alpha^2 + 2\,a^8 x^2\alpha^4 y^2 + 8\,a^9\alpha^2 x^3\beta^3 + 10\,x^4 a^5\alpha\,y + 10\,y^3 x^2 a^7\alpha + 12\,a^5 x^3\beta^3 y^2 + 10\,y^4\beta\,a^7 x + 2\,y^2\beta\,a^7 x^3 + 8\,\beta^3 x^3 a^7 y^2 + 14\,x^4 a^7\alpha\,y + 2\,y^4\alpha^2 a^6 x^2 + 8\,a^5\alpha^3 x^2 y^3 + 10\,y^2\beta\,a^5 x^3 + 2\,a^5 x^2\alpha\,y^3 + 14\,y^4 a^5 x\beta + 2\,x^4\beta^2 a^6 y^2 + 20\,a^8 x^2 y^2 + 8\,a^8 x^4\beta^2 + 7\,a^8 y^4\alpha^2 + 6\,a^8 y^4\alpha^4 + 16\,a^8 x^4\beta^4 + 32\,a^9 x^3\beta^3 - 8\,y^5 a^5\alpha + 32\,y^2\alpha^2 a^4 + 16\,y^2\alpha^2 a^6 + 38\,a^4\beta^2 y^2 + 20\,a^6\alpha^2 x^2 + 16\,x^2 a^{10}\beta^2 + 6\,y^2\alpha^2 a^{10} + 6\,\beta^2 x^2 a^2 + 16\,x^4\beta^4 a^6 + 6\,x^4\beta^4 a^4 + 16\,y^4\alpha^4 a^4 + 8\,y^4 a^4\alpha^2 - 4\,a^3 x^3\beta + 8\,a^5 x^3\beta + 48\,x^3 a^7\beta + 20\,x^2\alpha^2 a^{10} - 4\,ya^{11}\alpha^3 + 8\,ya^9\alpha^3 + 16\,ya^9\alpha + 48\,\beta^3 x^3 a^7 + 48\,y^3\alpha^3 a^5 + 48\,y^3 a^5\alpha + 8\,y^3 a^7\alpha + 8\,a^3 x\beta^3 + 48\,a^5 x\beta^3 + 3\,y^4\beta^2 a^4 - 4\,y^5\alpha\,a^7 + 32\,y^3 a^3\alpha + 3\,a^{10}\alpha^4 x^2 + 8\,a^8\alpha^4 y^2 + 7\,a^{10}\alpha^4 y^2 + 3\,a^8\alpha^2 x^4 - 8\,ya^5 a^9 - 4\,ya^5 a^{11} - 4\,\beta^5 x^3 a^3 + 3\,y^4 a^6 x^2 - 4\,y^5\alpha^3 a^7 + 32\,a^9 x^3\beta + 38\,y^2 a^6 x^2 + 20\,y^2 a^4 x^2 + 7\,x^4\beta^2 a^4 + 8\,x^4\beta^2 a^6 + 8\,y^4\alpha^2 a^6 + 16\,y^4\alpha^4 a^6 + y^6 a^6\alpha^2 + 16\,y^2\alpha^2 a^2 + 20\,a^4\alpha^2\beta^2 + 20\,\beta^2 y^2 a^2 + 8\,a^4\beta^4 x^2 + 7\,x^2\beta^4 a^2 + 3\,a^4\beta^4\alpha^2 + 3\,\beta^4 y^2 a^2 + 16\,a^6 x^2\beta^2 + 38\,a^6\alpha^2\beta^2 + 20\,y^2 a^6\beta^2 + 8\,a^6\alpha^4 y^2 + 8\,a^6 x^2\beta^4 + 38\,a^8\alpha^2 x^2 + 3\,a^8\alpha^4\beta^2 + \beta^6 x^2 a^2 + 32\,ya^5\alpha^3 + 32\,a^7 x\beta^3 - 4\,ax\beta^5 - 8\,x\beta^5 a^3 + 16\,\beta\,xa^3 + 48\,\beta\,xa^5 + 32\,\beta\,a^7 x + 48\,a^7 y\alpha + 48\,a^7 y\alpha^3 + 32\,y\alpha\,a^5 - 4\,ax\beta^3 - 8\,a^3 x^3\beta^3 + 20\,\alpha^2\beta^2 a^8 - 8\,\beta^5 x^3 a^5 - 8\,y^3\alpha^5 a^7 + 16\,a^8 + 32\,a^6 + \beta^4 + 16\,a^4 + \beta^6 - 8\,a^{10}\alpha^2 + 16\,a^8\beta^2 + 8\,a^6 x^2 + 32\,a^6\beta^2 - 8\,a^2\beta^2 + 16\,y^2 a^2 - 8\,a^4 x^2 - 8\,a^4\beta^4 - 8\,a^2\beta^4 + 8\,y^2 a^6 + 32\,y^2 a^4 - 8\,y^4 a^6 + x^4 a^4 - 8\,a^8\alpha^4 + a^{12}\alpha^6 + 8\,a^8\alpha^2 - 8\,y^4 a^4 + y^6 a^6 + 16\,a^4\alpha^2 + 32\,a^6\alpha^2 - 8\,x^4 a^6 + 8\,a^4\beta^2 - 8\,y^2 a^8 + 16\,x^2 a^{10} - 8\,a^{10}\alpha^4 + a^{12}\alpha^4 + x^6 a^6 + 32\,a^8 x^2 - 8\,a^8 x^4 + a^8 y^4 + x^6 a^6\beta^2 - 8\,a^7 x^5\beta^3 - 4\,y^3 a^9\alpha - 8\,y^3\alpha^3 a^9 - 4\,y^3 a^9\alpha^5 - 4\,a^5 x^5\beta^3 - 8\,y^5\alpha^3 a^5 - 8\,a^7 x^5\beta + 32\,y^3\alpha^3 a^3 + 32\,a^8 x^2\beta^2 + 3\,x^4 a^6 y^2 + y^2 a^{10}\alpha^6 - 4\,a^5 x^5\beta < 0$$

Future plans for RAGlib

- Dedicated algorithms for structured systems (e.g. determinantal problems, **S. Naldi**)
- Implementation of an algorithm for computing the real dimension of semi-algebraic sets (**I. Bannwarth**)

Outline

Univariate Real Root Isolation

Problem

Given $A \in \mathbb{Z}[X]$ such that

$$A = a_d X^d + \cdots + a_1 X + a_0 \quad \in \mathbb{Z}[X]$$

where

$$\mathbf{d} = \deg(A) \quad \text{and} \quad \mathcal{L}(A) = \max_{0 \le i \le d} \{\lg |a_i|\} = \tau$$

1. Count the real roots of A
2. Compute the real roots of A (isolating intervals)
3. Refine the intervals up to a desired precision

EXACT ALGORITHMS: Use only integer/rational arithmetic!

Example

Let

$$f = x^5 - 7x^4 + 22x^3 - 4x^2 - 48x + 36 = (x-1)\cdot(x^2 - 6x + 18)\cdot(x^2 - 2)$$

real roots	$-\sqrt{2}$	1	$+\sqrt{2}$
output	$(-49, 0)$	$(\frac{49}{64}, \frac{147}{128})$	$(\frac{147}{128}, 49)$

SoLVe (SLV)

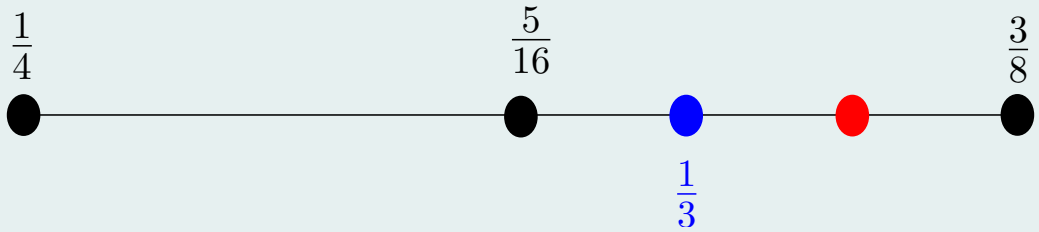
- version 0.2
- C library for real root isolation (currently $\sim 5\,000$ LoC)
- Algorithms
  - DESCARTES
  - CF
  - ...

CF vs Descartes

Their difference by an "example"

If the input is $[\frac{1}{4}, \frac{3}{8}]$ then

- Subdivision chooses $\frac{5}{16}$, thus splits in $[\frac{1}{4}, \frac{5}{16}]$ and $[\frac{5}{16}, \frac{3}{8}]$.
- CF chooses $\frac{1}{3}$ thus splits in $[\frac{1}{4}, \frac{1}{3}]$ and $[\frac{1}{3}, \frac{3}{8}]$



SoLVe (SLV)

- version 0.2
- C library for real root isolation (currently $\sim 5\,000$ LoC)

SoLVe (SLV)

- version 0.2
- C library for real root isolation (currently $\sim 5\,000$ LoC)
- Various algorithms

	Orig	MEM	FLINT
DESCARTES	✓	✓	✓
CF	✓		
HYBRID	✓		

SoLVe (SLV)

- version 0.2
- C library for real root isolation (currently $\sim 5\,000$ LoC)
- Various algorithms

	Orig	MEM	FLINT
DESCARTES	✓	✓	✓
CF	✓		
HYBRID	✓		

- Highly efficient for degree $\leq 1\,000$ and bitsize $\sim 6\,000$ bits

  e.g. Katsura-10 (216 real roots) in 32 *s*

SoLVe (SLV)

- version 0.2
- C library for real root isolation (currently $\sim 5\,000$ LoC)
- Various algorithms

	Orig	MEM	FLINT
DESCARTES	✓	✓	✓
CF	✓		
HYBRID	✓		

- Highly efficient for degree $\leq 1\,000$ and bitsize $\sim 6\,000$ bits
  e.g. Katsura-10 (216 real roots) in 32 *s*
- Expected release date: *end of 2014*

SLV future plans

- Expected release date: *end of 2014*
- Full integration with `FGb` (*in the very near future*)
- Parallel (multi-core) version

Polynomial System Solving

Solving efficiently polynomial system of equations is a fundamental problem in Computer Algebra with many applications.

- Zero-Dimensional Systems / Gröbner Bases: FGb
- Positive Dimension: RAGLIB solutions over the reals
- SLV: new C library for isolating and refining the real roots of univariate polynomials.