Symbolic deformation techniques for polynomial system solving

Lecture 1

by Grégoire Lecerf

Université de Versailles & CNRS
France
http://www.math.uvsq.fr/~lecerf

Complexity of Numerical Computation, 2009

General Introduction to algebraic solvers

$$f_1(x_1, \ldots, x_n) = \cdots = f_s(x_1, \ldots, x_n) = 0, \qquad g(x_1, \ldots, x_n) \neq 0$$

Motivation

- Cornerstone for all operations arising in computational algebraic geometry and differential algebra.
- Several applications in the industry: signal theory, robotics (motion planning), cryptography,...
- At the present time no quasi-optimal algorithm is known for the general case.
- Still a very active research area: several methods, symbolic and numeric, offer different advantages and drawbacks according to the system to be solved.
- Deciding which is the best method on a given system is a difficult theoretical and practical problem.

Known families of algorithms

- First techniques and heurisistics go back to the very early ages of mathematics. The first general method seems to be due to KRONECKER (1882).
- General symbolic algorithmic descriptions really started in the sixties:
  - standard bases: constructive elimination was used by HIRONAKA in his seminal works in desingularization;
  - Gröbner bases: popularized via BUCHBERGER's algorithm.
- Latter, older symbolic techniques have been studied and improved for computions: resultants, triangular decompositions, MACAULAY's matrices, F4&5 (by FAUGÈRE), etc.
- Numerical techniques (subdivisions, Newton, homotopy continuation, etc) have been designed independently untill the 90's.
- Nowadays mixed numerical and symbolic techniques offer good performances. Complexity analysis has been carefuly done in several cases.

Sample of general references

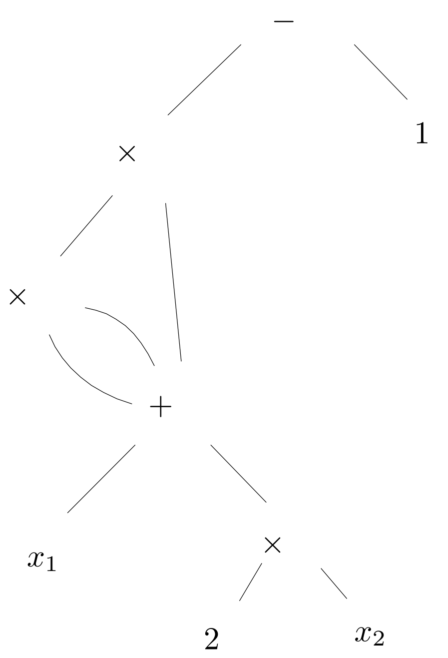
- BECKER and WEISPFENNING, Gröbner bases. A computational approach to commutative algebra, Springer, 1993.
- GREUEL, PFISTER, BACHMANN, and LOSSEN, A `Singular` Introduction to Commutative Algebra, Springer, 2007.
- COX, LITTLE, and O'SHEA, Ideals, varieties, and algorithms. An introduction to computational algebraic geometry and commutative algebra, Springer, 1997.
- COX, LITTLE, and O'SHEA, Using algebraic geometry, Springer, 2005.
- VON ZUR GATHEN, and GERHARD, Modern computer algebra, Cambridge University Press, 2003.
- SCHENK, Computational algebraic geometry, Cambridge University Press, 2003 – examples with Macaulay 2.

- Sommese, and Wampler, The Numerical Solution of Systems of Polynomials: Arising in Engineering And Science, Word Scientific Publishing, 2005.
- Blum, Cucker, Shub, and Smale, Complexity and real computation, Springer, 1997.
- Dedieu, Points fixes, zéros et la méthode de Newton, Springer, 2006.
- ...

Introduction to the Kronecker solver

Representation of multivariate polynomials

- Dense representation: store all the monomials up to a certain degree.
  Used in Gröbner bases, triangular decompositions.
  $f(x_1, x_2) = 8\,x_2^3 + 12\,x_1\,x_2^2 + 6\,x_1^2\,x_2 + x_1^3 + 0\,x_2^2 + 0\,x_1\,x_2 + 0\,x_1^2 + 0\,x_2 + 0\,x_1 - 1$
- Sparse representation: store only the non-zero monomials.
  $f(x_1, x_2) = 8\,x_2^3 + 12\,x_1\,x_2^2 + 6\,x_1^2\,x_2 + x_1^3 - 1$
- Functional representation: store a function for evaluating the polynomial at any given point.
  Used in most of the numerical solvers and the Kronecker solver.
  $f(x_1, x_2) = (x_1 + 2\,x_2)^3 - 1 =$



Directed acyclic graph (DAG)

Impact of the representation on the complexity

Example 1. Solve a linear system made of symbolic coefficients:

$$\begin{pmatrix} a_{1,1} & \dots & a_{n,1} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \dots & a_{n,n} \end{pmatrix}\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}=\begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}.$$

Dense and sparse representation: $\det(a_{i,j})$ has $n!$ monomials of degree n.
Functional representation: $\det(a_{i,j})$ can be evaluated with $\mathcal{O}(n^4)$ operations by Berkowitz' algorithm.

Example 2. Decide if two univariate polynomials with symbolic coefficients have a common root:

$$f_1(x)=a_0+a_1\,x+\cdots+a_n\,x^n, \quad f_2(x)=b_0+b_1\,x+\cdots+b_n\,x^n.$$

Dense and sparse representation: Resultant (f_1, f_2) is a polynomial in the $2\,(n+1)$ coefficients of size that grows exponentially with n.
Functional representation: Resultant (f_1, f_2) can be evaluated in $\mathcal{O}(n^4)$ as the determinant of the Sylvester matrix.

Example 3. Polynomial system in $2\,n$ variables with n random equations of degree d.

Dense and sparse representation: eliminant polynomials in n variables and degree d^n, hence a size that grows with d^{n^2} for fixed d and n goes to infinity.
Functional representation: such eliminant polynomials can be evaluated with $d^{\mathcal{O}(n)}$ operations (e.g. via the Kronecker algorithm).

Paradigm. Polynomials produced by an elimination procedure have nive evaluation properties.

Prerequisite

$\mathbb{K}$: any field with algebraic closure $\bar{\mathbb{K}}$.
$\mathbb{K}[x_1, \dots, x_n]$: polynomial ring over $\mathbb{K}$ with n variables.
$\mathcal{I}\subseteq\mathbb{K}[x_1, \dots, x_n]$: ideal.

Zariski topology

Definition 4. *The affine variety $\mathcal{V}(\mathcal{I})$ defined by $\mathcal{I}$:*

$$\mathcal{V}(\mathcal{I})=\{(a_1, \dots, a_n)\in\bar{\mathbb{K}}^n\,|\,\forall f\in\mathcal{I}, f(a_1, \dots, a_n)=0\}.$$

Proposition 5. $\mathcal{V}(\mathcal{I}_1)\cap\mathcal{V}(\mathcal{I}_2)=\mathcal{V}(\mathcal{I}_1+\mathcal{I}_2)$, $\mathcal{V}(\mathcal{I}_1)\cup\mathcal{V}(\mathcal{I}_2)=\mathcal{V}(\mathcal{I}_1\cap\mathcal{I}_2)$.

Definition 6. *Zariski topology of $\bar{\mathbb{K}}^n$: affine varieties are the closed set.*

Definition 7. *The vanishing ideal $\mathcal{I}(\mathcal{E})$ of a subset $\mathcal{E}\subseteq\bar{\mathbb{K}}^n$:*

$$\mathcal{I}(\mathcal{E})=\{f\in\mathbb{K}[x_1, \dots, x_n]\,|\,\forall(a_1, \dots, a_n)\in\mathcal{E}, f(a_1, \dots, a_n)=0\}.$$

Proposition 8. *(Nullstellensatz)* $\mathcal{I}(\mathcal{V}(\mathcal{J}))=\sqrt{\mathcal{J}}:=\{f\,|\,\exists n, f^n\in\mathcal{J}\}$.

Saturation

Definition 9. *Saturation of $\mathcal{I}$ with respect to $g \in \mathbb{K}[x_1, \ldots, x_n]$:*

$$\mathcal{I}: g^\infty = \{ f \in \mathbb{K}[x_1, \ldots, x_n] \,|\, \exists n, g^n f \in \mathcal{I} \}.$$

Proposition 10. *$\mathcal{V}(\mathcal{I}: g^\infty)$ is the Zariski closure of $\mathcal{V}(\mathcal{I}) \setminus \mathcal{V}(g)$.*

Example 11. $\mathcal{I} = x_1 (x_1^2 + x_2^2 - 1)$, $g = x_1$, $\mathcal{I}: g^\infty = (x_1^2 + x_2^2 - 1)$.

Overview of the Kronecker solver

$\mathbb{K}$: any field of characteritic 0, or sufficiently large.
$f_1, \ldots, f_s, g$: polynomials in $\mathbb{K}[x_1, \ldots, x_n]$.

$$f_1(x_1, \ldots, x_n) = \cdots = f_s(x_1, \ldots, x_n) = 0, \qquad g(x_1, \ldots, x_n) \neq 0$$

Notations

$\mathcal{I}_i = (f_1, \ldots, f_i): g^\infty$, $\quad \mathcal{J}_i = \mathcal{I}_i + (x_1, \ldots, x_{n-i})$, $\quad \mathcal{K}_i = \mathcal{I}_i + (x_1, \ldots, x_{n-i-1})$

Assumptions

For simplicity we assume that the system is regular and reduced:

- $s = n$,
- f_{i+1} is a nonzerodivisor modulo $\mathcal{I}_i$: $f_{i+1} h \in \mathcal{I}_i \Rightarrow h \in \mathcal{I}_i$,
- $\mathcal{I}_i$ is radical: $\mathcal{I}_i = \sqrt{\mathcal{I}_i}$.

Consequences

- $\dim \mathcal{I}_i = n - i$.
- The system admits a finite number of solutions.

Main idea

With sufficiently generic coordinates $\mathcal{V}(\mathcal{J}_i)$ is a finite set of points, and $\mathcal{V}(\mathcal{K}_i)$ is a curve. We construct a symbolic deformation from

$$\mathcal{J}_i: f_1(x_1, \ldots, x_n) = \cdots = f_i(x_1, \ldots, x_n) = x_1 = \cdots = x_{n-i} = 0, \, g(x_1, \ldots, x_n) \neq 0,$$

to

$$\mathcal{J}_{i+1}: f_1(x_1, \ldots, x_n) = \cdots = f_{i+1}(x_1, \ldots, x_n) = x_1 = \cdots = x_{n-i-1} = 0, \, g(x_1, \ldots, x_n) \neq 0,$$

by following the curve

$$\mathcal{K}_i: f_1(x_1, \ldots, x_n) = \cdots = f_i(x_1, \ldots, x_n) = x_1 = \cdots = x_{n-i-1} = 0, \, g(x_1, \ldots, x_n) \neq 0.$$

Representation of the finite solution sets

A zero-dimensional solution set $\mathcal{E}$ can be parametrized in this way:

$$\mathcal{E} = \{ (v_1(\alpha), \ldots, v_n(\alpha)) \,|\, q(\alpha) = 0 \},$$

with q and the v_i in $\mathbb{K}[T]$.

Definition 12. *The data of sufficiently generic coordinates, and of such a representation for $\mathcal{J}_i$ is called a lifting fiber for $\mathcal{I}_i$.*

Remark 13. Also known as witness sets in the numerical algorithms works by Sommese *et al.*

Representation of the solution curves

A solution curve $\mathcal{C}$ can be parametrized in this way:

$$\mathcal{C} = \overline{\left\{\left(\frac{w_1(\alpha,\beta)}{q'(\alpha,\beta)}, \ldots, \frac{w_n(\alpha,\beta)}{q'(\alpha,\beta)}\right) \mid q(\alpha,\beta) = 0, q'(\alpha,\beta) \neq 0\right\}},$$

where q and the w_i are in $\mathbb{K}[t,T]$, $q' = \frac{\partial q}{\partial T}$.
With sufficiently generic coordinates we have: $\deg_t q$ and $\deg_t w_i \leq \deg_T q$.

Definition 14. *The data of sufficiently generic coordinates, and of such a representation for $\mathcal{K}_i$ is called a lifting curve for $\mathcal{I}_i$.*

Overview of the algorithm

1. Perform a random affine change of the variables.
   – this makes $\mathcal{J}_i$ have a finite set of solutions that are all regular.
2. Initialize the process with the solution set of $\mathcal{J}_0 = (x_1, \ldots, x_n)$.

   From the finite solution set of $\mathcal{J}_i$ compute the one of $\mathcal{J}_{i+1}$ as follows:

   a) Lifting step: compute a representation of the curve $\mathcal{K}_i$.

   b) Intersection step: compute a representation of the finite set of points of $\mathcal{K}_i + (f_{i+1})$, that is the intersection of the latter curve with the hypersurface defined by f_{i+1}.

   c) Cleaning step: deduce $\mathcal{J}_{i+1} = (\mathcal{K}_i + (f_{i+1}))$: g^∞, by removing from the previous set the points in the hypersurface $g = 0$.
3. Rewrite the solutions of $\mathcal{J}_n$ in terms of the orginal variables.

Example 15. (with no inequation). $\mathcal{J}_0 = (x_1, \ldots, x_n)$, the only solution is 0.

1. First step, $i = 0$.

   a) lifting: we obtain $\mathcal{K}_0 = (x_1, \ldots, x_{n-1})$ that defines a line.

   b) intersection: $\mathcal{K}_0 + (f_1) = (f_1) + (x_1, \ldots, x_{n-1})$ defines the solutions of $f_1(0, \ldots, 0, x_n) = 0$.
2. Second step, $i = 1$.

   a) lifting: $\mathcal{K}_1 = (f_1) + (x_1, \ldots, x_{n-2})$ corresponds to the curve defined by $f_1(0, \ldots, 0, x_{n-1}, x_n) = 0$.

   b) intersection: $\mathcal{K}_1 + (f_2) = (f_1, f_2) + (x_1, \ldots, x_{n-2})$ corresponds to the intersection of the two plane curves $f_1(0, \ldots, 0, x_{n-1}, x_n) = 0$ and $f_2(0, \ldots, 0, x_{n-1}, x_n) = 0$.
3. Third step, $i = 2$.

   a) lifting: $\mathcal{K}_2 = (f_1, f_2) + (x_1, \ldots, x_{n-3})$ corresponds to the curve defined by $f_1(0, \ldots, 0, x_{n-2}, x_{n-1}, x_n) = f_2(0, \ldots, 0, x_{n-2}, x_{n-1}, x_n) = 0$.

   b) intersection: $\mathcal{K}_2 + (f_3) = (f_1, f_2, f_3) + (x_1, \ldots, x_{n-3})$ corresponds to the intersection of the latter curve with $f_3(0, \ldots, 0, x_{n-2}, x_{n-1}, x_n) = 0$.
4. ...

Examples

Example 16. Graphical example with `Axel` (Mourrain et al., `http://axel.inria.fr`).

Example 17. Naive implementation with Mathemagix (van der Hoeven, Lecerf, Mourrain, Ruatta, *et al.* http://www.mathemagix.org) – currently used graphical interface is GNU TeXmacs (van der Hoeven *at al.*, http://www.texmacs.org).

```
Mmx] include "gregorix/kronecker_naive.mmx";
Mmx] n:= 3;
     f == [ x1^2 + x2^2 + x3^2 - 2,
            x1^2 + x2^2 - 1,
            x1 - x2 + 3 * x3 ];
     x == [x1, x2, x3];
     y == [x1, x2 - 2 * x3, x3];
     f == replace (f, x, y)
```

$$\left[(x2-2\,x3)^2+x1^2+x3^2-2,(x2-2\,x3)^2+x1^2-1,x1-x2+5\,x3\right]$$

```
Mmx] T == polynomial (rational 0, 1);
     q == monic_part evaluate (replace (f[0], [x1,x2],
                                          [0:>Symbolic,0]),
                               [x3], [T], polynomial);
     v == [T];
Mmx] $lifting_fiber (x, q, v) // for f1(0,0,x3)= 0
```

$$x^2-\frac{2}{5}=0,\quad \begin{array}{rcl} x1 & = & 0\\ x2 & = & 0\\ x3 & = & x \end{array}$$

```
Mmx] K1 == lift_curve (f[0,1], x, q, v);
     q == car K1; w == car cdr K1;
Mmx] $lifting_curve (x, q, w) // for f1(0,x2,x3)= 0
```

$$y^2-\frac{4}{5}x\,y+\frac{1}{5}x^2-\frac{2}{5}=0,\quad \begin{array}{rcl} x1 & = & 0\\ x2 & = & x\\ x3 & = & \dfrac{\frac{4}{5}x\,y-\frac{2}{5}x^2+\frac{4}{5}}{2\,y-\frac{4}{5}x} \end{array}$$

```
Mmx] J2 == intersect (f[1], x, q, w);
     q == car J2; v == car cdr J2;
Mmx] $lifting_fiber (x, q, v) // for f1(0,x2,x3)= f2(0,x2,x3)= 0
```

$$x^4-10\,x^2+9=0,\quad \begin{array}{rcl} x1 & = & 0\\ x2 & = & x\\ x3 & = & \dfrac{-1}{12}x^3+\dfrac{13}{12}x \end{array}$$

```
Mmx] K2 == lift_curve (f[0,2], x, q, v);
     q == car K2; w == car cdr K2;
Mmx] $lifting_curve (x, q, w)
        // for f1(x1,x2,x3)= f2(x1,x2,x3)= 0
```

$$y^4+(2\,x^2-10)\,y^2+x^4+6\,x^2+9=0,\quad \begin{array}{rcl} x1 & = & x\\ x2 & = & \dfrac{(-4\,x^2+20)\,y^2-4\,x^4-24\,x^2-36}{4\,y^3+(4\,x^2-20)\,y}\\ x3 & = & \dfrac{8\,y^2-8\,x^2-24}{4\,y^3+(4\,x^2-20)\,y} \end{array}$$

```
Mmx] J3 == intersect (f[2], x, q, w);
     q == car J3; v == car cdr J3;
Mmx] $lifting_fiber (x, q, v) // for f1= f2= f3= 0
```

$$x^4 - x^2 + 16 = 0, \quad \begin{array}{rcl} x1 & = & x \\ x2 & = & \frac{5}{12} x^3 - \frac{13}{12} x \\ x3 & = & \frac{1}{12} x^3 - \frac{5}{12} x \end{array}$$

Feature summary

- Take advantage of the evaluation properties of the system.
- Handle easily $g \neq 0$.
- Cost depends on a geometric degree.
- Use dense polynomials in two variables only – fast arithmetic available.
- High probability of success.

L: evaluation cost of the system.
d: maximum of the total degree of the f_i.
δ: maximum number of solutions of the intermediate systems $\mathcal{J}_i$.
D: final number of solutions.

Theorem 18. *[Giusti, Lecerf, Salvy, 2001] The Kronecker solver takes*

$$n(n\,L + n^4)(d\,\delta)^2 \log(d\,\delta)^{\mathcal{O}(1)}$$

operations in $\mathbb{K}$.
If $\mathbb{K} = \mathbb{Q}$, the resolution is done modulo a "suitable" prime number p. Then the solutions over $\mathbb{Q}$ are lifted with $(n\,L + n^4)\eta D \log(\eta D)^{\mathcal{O}(1)}$ bit operations, where η is the bit-size of the integers of the output.

Example 19. Random equations of degree 2, with coefficients of bit-size 2.
$\delta = 2^n$ (Bézout), $\eta \sim 2^n$ (arithmetic Bézout)
Cost of the resolution modulo p: $4^n\, n^{\mathcal{O}(1)}$ operations mod p.
Bit-cost of the lifting of the integers: $4^n\, n^{\mathcal{O}(1)}$.
Size of the output: at least $\theta(n4^n)$ (arithmetic Bézout).
Quasi optimality!

Goals of the present lectures

Rest of this lecture

- Brief history of the Kronecker solver

Lecture 2

- Incremental solving
- Computational dimension theory
- Computational degree theory

Lecture 3

- Representation of the solution sets
- Complete presentation of the Konecker solver
- Overview of the possible extensions

Historical digression

- We could discard the lifting step by using a functional representation of $\mathcal{I}_i$ as follows: $(f_1, \ldots, f_i) : g^\infty =$

$$(q(x_1, \ldots, x_r, T), x_{r+1} - v_{r+1}(x_1, \ldots, x_r, T), \ldots, x_n - v_n(x_1, \ldots, x_r, T)), \quad (1)$$

  in $\mathbb{K}(x_1, \ldots, x_r)[x_{r+1}, \ldots, x_n]$, with $r = n - i$.

- This was the way the Kronecker solver started to be designed, but it leads to a much higher cost.
- Then the Newton operator was introduced to compress the representation in (1):
  1. Specialize (1) at a random value $x_1 = a_1, \ldots, x_r = a_r$.
  2. Use a variant of the Newton iterator to get a good functional representation of (1), with size bounded in terms of L and $\deg q$ only.
- Using such a functional representation in all the intermediate steps of the solver has the following advantages:
  - easier mathematical description,
  - better control of the probabilities,
  - possibility to have deterministic algorithms for a non-uniform complexity model;

  but the following drawbacks:
  - memory managment for the functional representation,
  - non-optimized algorithms for polynomials,
  - the deterministic non-uniform model is not tracktable into practice.
- Moving from this original version of the solver to the one presented here appealed to the deforestation paradigm: elimination of useless temporary data structures.
- The Kronecker algorithm already presented is deforested: functional data structures are only used for the input polynomials.

Brief history of the Kronecker solver

- Hommage to Kronecker (1882) method, but far more sophisticated.
- Giusti, Heintz, Morais, Pardo: first symbolic algorithms exploiting functional representation at the begining of the 90s: dimension, Noether position, Nullstellensatz.
- The non-deforested version of the Kronecker solver first appeared in works by Giusti, Hägele, Heintz, Montaña, Morais, Morgenstern, Pardo, 1993, 1995, 1997, 1998: incremental solving, symbolic Newton operator, polynomial time.
- Simplifications and first extensions: Ph.D theses of Morais (1997) and Hägele (1998).
- Functional data structures implementation: Castaño, Llovet, Martìnez (1996), Hägele (1998), Bruno, Heintz, Matera, Wachenchauzer (2002).
- Practical computation of the dimension, and deforestation paradigm: Giusti, Hägele, Marchand, Lecerf, Salvy (2000).
- Practical deforested version of the Kronecker solver: Giusti, Lecerf, Salvy (2001). Implementation in `Magma`. Lecerf's Ph.D thesis (2001).
- Better probability and space analyses: Matera (1999), Heintz, Matera, Waissbein (2001).
- Extension for computing the Chow form: Jeronimo, Krick, Sabia, Sombra (2001, 2004).