Computing and Using Minimal Polynomials

Presentation · June 2018

CITATIONS
0

READS
72

4 authors:

John Abbott
Università degli Studi di Genova
85 PUBLICATIONS **452** CITATIONS
SEE PROFILE

Anna Maria Bigatti
Università degli Studi di Genova
99 PUBLICATIONS **727** CITATIONS
SEE PROFILE

Elisa Palezzato
Università degli Studi di Genova
8 PUBLICATIONS **10** CITATIONS
SEE PROFILE

Lorenzo Robbiano
Università degli Studi di Genova
133 PUBLICATIONS **2,545** CITATIONS
SEE PROFILE

Computing and Using Minimal Polynomials

J. Abbott
A.M. Bigatti
E. Palezzato
L. Robbiano

Anna Maria Bigatti
Università di Genova

SC^2: H2020-FETOPEN-2016-2017-CSA project 712689 `www.sc-square.org`

Definition

Definition: Minimal Polynomial

K a field, $P = K[x_1, \ldots, x_n]$, I zero-dimensional ideal in P.

The **minimal polynomial** of a polynomial f modulo I, $\mu_{f,I}(z) \in K[z]$,

is the monic polynomial in $K[z]$ of minimum degree such that

$$\mu_{f,I}(f) \in I \qquad \text{or equiv.} \qquad \mu_{f,I}(\bar{f}) = \bar{0} \text{ in } P/I$$

```
/**/ I := ideal(x^2, y^2);
/**/ MinPolyQuot(x+y,I, t);
t^3     ----------> (x+y)^3  is in I
/**/ f := x^2- 3*x*y +1;
/**/ MinPolyQuot(f,I, t);
t^2 -2*t +1  ----> f^2 -2*f +1  is in I
```

Remark

If x_i an indeterminate in $P = K[x_1, \ldots, x_n]$
$\mu_{x_i,I}(x_i)$ is the lowest degree x_i-univariate polynomial in I
i.e. $I \cap K[x_i] = \langle\, \mu_{x_i,I}(x_i) \,\rangle$.

```
/**/ I := IdealOfPoints(P, mat([[1,2], [3,2], [5,4]]));
/**/ MinPolyQuot(x,I, x);
x^3 -9*x^2 +23*x -15   ----> (x-1)(x-3)(x-5)
/**/ MinPolyQuot(y,I, y);
y^2 -6*y +8            ----> (y-2)(y-4)
```

Remark

For a CAS like CoCoA → Gröbner Bases → elimination:
well known solution, simple and elegant 😀
... but slow and memory hungry 😫

⟶ worth implementing a dedicated algorithm

"by definition" –> Linear algebra

ALGORITHM MINPOLYQUOTDEF $P = K[x_1, \ldots, x_n]$, term-ordering σ

Input: I a zero-dimensional ideal in P, f polynomial in P

- compute GB, the σ-Gröbner basis for I
  from GB compute QB, the monomial quotient basis of P/I
- let $r_0 = f^0$ $(= 1)$
- *Main Loop:* for $i = 1, 2, \ldots, \mathrm{len}(QB)$ do
  - compute $r_i = \mathrm{NF}(f^i)$ $\quad [= \mathrm{NF}(f \cdot r_{i-1})]$
  - if there is a linear dependency $r_i = \sum_{j=0}^{i-1} c_j r_j$ with coefficients $c_j \in K$
    return $z^i - \sum_{j=0}^{i-1} c_j z^j$

Output: $\mu_{f,I}(z) \in K[z]$

```
/**/ QuotientBasis(I); --------> [1,   y,   x]
/**/ y^0;          --> 1         [1,   0,   0]
/**/ NF(y^1, I);   --> y         [0,   1,   0]
/**/ NF(y^2, I);   --> 6*y -8    [-8,  6,   0]
```

Timings over $\mathbb{F}_p$

MinPolyQuotDef carefully optimized $\longrightarrow$ 

Example	GB	MinPoly			MinPoly degree
		Def	Mat	Elim	
charp-deg500 f_1	0.38	4.10	7.06	50.54	500
charp-deg500 f_2	0.38	5.77	9.14	∞	500
charp-split6	0.00	2.43	12.29	∞	720
1000000007-randomp	0.17	4.43	9.02	∞	590
23largeCI	0.00	1.06	20.68	∞	880

Def: "by definition"
Mat: by multiplication matrix
Elim: by elimination

... and with rational coefficients? 

Rational coefficients: modular methods

Definition π_p: reduction modulo p

Let $\delta \in \mathbb{N}_+$ and p a prime not dividing δ.

$$\begin{array}{lccccc} \pi_p: & \mathbb{Z}_\delta & \longrightarrow & \mathbb{F}_p & a/\delta^d & \mapsto & \bar{a}_t/\bar{\delta}^d \\ \pi_p: & \mathbb{Z}_\delta[x_1,\ldots,x_n] & \longrightarrow & \mathbb{F}_p[x_1,\ldots,x_n] & \sum_t c_t t & \mapsto & \sum_t \pi_p(c_t)t \end{array}$$


But how can we define the reduction modulo p of an **ideal**?

Theorem (Reduction modulo p of Gröbner Bases)

I non-zero ideal in $\mathbb{Q}[x_1,\ldots,x_n]$, GB its reduced σ-Gröbner basis.
Let p be any prime not dividing $\mathrm{den}(GB)$.

1. the reduced σ-Gröbner basis of $\langle \pi_p(GB) \rangle$ is $\pi_p(GB)$
2. f such that $p \not| \mathrm{den}(f) \longrightarrow$ the NF of $\pi_p(f)$ is $\pi_p(\mathrm{NF}_{\sigma,I}(f))$

$\longrightarrow$ $\boldsymbol{I_{(p,\sigma)}} = \langle \pi_p(G) \rangle$ More in Abbott,Bigatti,Robbiano: *"Ideals mod p"*

Modular methods for minimal polynomials

I zero-dimensional ideal, f polynomial in $\mathbb{Q}[x_1, \ldots, x_n]$.

Proposition

$\delta = \mathrm{den}(f) \cdot \mathrm{den}(GB_\sigma(I))$ *then* $\mu_{f,I}(z)$ *has all coefficients in* $\mathbb{Z}_\delta$.

Example 1

$P = \mathbb{Q}[x, y]$ and $I = \langle 2x + 3y,\ y^2 - 4 \rangle$.
Two possible Gröbner bases: $\{x + \frac{3}{2}y,\ y^2 - 4\}$ and $\{y + \frac{2}{3}x,\ x^2 - 9\}$
$f = 23x + 17y$ then $\mu_{f,I}(z)$ has integer coefficients $(= z^2 - 1225)$.

Theorem (Bad primes)

1. There are only finitely many bad primes.
2. $\pi_p(\mu_{f,I}(z))$ is a multiple of $\mu_{\pi_p(f),I_{(p,\sigma)}}(z)$.

$\longrightarrow$ detect bad primes 

Timings over $\mathbb{Q}$

Modular computation + CRT + rational reconstruction $\longrightarrow$ 

Example	GB	MinPoly				
		$\mathbb{Q}$	Modular verified		coeff	deg
	time	*time*	*time*	*#p*		
QQ-rand	0.15	∞	15.43	64	$10^{389}, 10^{188}$	116
QQ-CI1 ℓ_1	0.00	47.86	0.39	12	$10^{93}, 10^{0}$	107
QQ-CI1 ℓ_2	0.00	226.34	1.31	25	$10^{210}, 10^{0}$	108
QQ-CI2	0.00	∞	3.77	38	$10^{330}, 10^{0}$	144
QQ-split5	0.00	∞	0.67	9	$10^{64}, 10^{0}$	120
QQ-split6	0.00	∞	175.14	58	$10^{503}, 10^{0}$	720
QQ-largeCI	0.00	233.24	0.39	5	$10^{29}, 10^{4}$	230
twomaxhard	0.42	∞	18.12	30	$10^{234}, 10^{19}$	149
twomaxsimple	0.33	5.33	0.67	15	$10^{108}, 10^{12}$	55
PrimaryNotMax	0.00	510.85	3.45	3	$10^{11}, 10^{0}$	252

Using Minimal Polynomials

Remark

$\ell \in K[x_1, \ldots, x_n]$ *a* ***generic*** *linear form,* I *zero-dimensional ideal*

- I *not radical* $\Longrightarrow$ $\mu_{\ell,I}$ *not square-free*
- I *radical* $\Longrightarrow \deg(\mu_{\ell,I}) = d$

If K *is* ***big enough*** ***generic*** $\longrightarrow$ ***random***

[More in Kreuzer,Robbiano book:
"Computational Linear and Commutative Algebra"]

😀 Some applications $\longrightarrow$

IsMaximal(I): practically effective NON-algorithm!

[ALGORITHM] ISMAXIMAL

Input I, an ideal in P

- *Loop:* repeat
  - pick a **random** linear form $\ell \in P$; compute $\mu = \mu_{\ell,I}$
  - if μ is reducible then **return false**
  - if $\deg(\mu) = d$ then **return true**

Output *true/false* indicating the maximality of I.

Remark

ISMAXIMAL *is not an algorithm because termination is not guaranteed. But in practice recall: if* ℓ *a* **random** *linear form (*K *is* ***big enough****) then*

I *not radical* $\Longrightarrow$ $\mu_{\ell,I}$ *not square-free*

I *radical* $\Longrightarrow$ $\deg(\mu_{\ell,I}) = d$

This is neat and elegant, but better faster ISMAXIMAL $\longrightarrow$ 

IsMaximal(I): a *very* effective NON-algorithm!

[ALGORITHM] ISMAXIMAL

Input I, an ideal in P

1. if I is not zero-dimensional, **return false**
2. compute $d = \dim_K(P/I)$
3. *First Loop:* for each indeterminate x_i do
   - 3.1 compute $\mu = \mu_{x_i,I}$
   - 3.2 if μ is reducible then **return false**
   - 3.3 if $\deg(\mu) = d$ then **return true**
4. if K is finite then (..Frobenius space..)
5. *Second Loop:* repeat
   - 5.1 pick a **random** linear form $\ell \in P$; compute $\mu = \mu_{\ell,I}$
   - 5.2 if μ is reducible then **return false**
   - 5.3 if $\deg(\mu) = d$ then **return true**

Output *true/false* indicating the maximality of I.

Radical

ALGORITHM RADICAL0DIM

Input I, a zero-dimensional ideal in P

1 let $J = I$ and compute $d = \dim_K(P/J)$

2 *Main Loop:* for each indeterminate x_i do

2.1 compute $\mu = \mu_{x_i,J}$

2.2 if μ is not square-free then

2.2.1 let $\mu = \mathrm{rad}(\mu)$

2.2.2 let $J = J + \langle \mu(x_i) \rangle$

2.2.3 **compute** $d = \dim_K(P/J)$
(if it is worth it $\longrightarrow$ **Timeout**)

2.3 if $\deg(\mu) = d$ then **return** J

3 **return** J

Output the radical of I

Many application of minimal polynomials

I zero-dimensional ideal

- **IsRadical(*I*)**
- **Radical(*I*)** (seen)
- **IsMaximal(*I*)** (seen)
- **IsPrimary(*I*)**:
  combination of IsMaximal and Radical
- **PrimaryDecomposition(*I*)**:
  combination of MinPoly and IsPrimary

- and probably **most of the applications** found in literature which mention *Lex Gröbner bases*!

Thank you!!

Many application of minimal polynomials

I zero-dimensional ideal

- **IsRadical(*I*)**
- **Radical(*I*)** (seen)
- **IsMaximal(*I*)** (seen)
- **IsPrimary(*I*)**:
  combination of IsMaximal and Radical
- **PrimaryDecomposition(*I*)**:
  combination of MinPoly and IsPrimary

- and probably **most of the applications** found in literature which mention *Lex Gröbner bases*!

Thank you!!