Fundamental Algorithms and Implementation Techniques for Computing with Regular Chains

Marc Moreno Maza
(Ontario Research Center for Computer Algebra)
(Univ. of Western Ontario)

SSSC'09
Chengdu, China,
August 12, 2009

How did regular chains emerge? (1/3)

- Let $\mathbf{K}$ be an algebraically closed field, say $\mathbb{C}$, and $\mathbf{k}$ be a subfield of $\mathbf{K}$, say $\mathbb{Q}$. Consider n variables $x_1 < \cdots < x_n$.

How did regular chains emerge? (1/3)

- Let $\mathbf{K}$ be an algebraically closed field, say $\mathbb{C}$, and $\mathbf{k}$ be a subfield of $\mathbf{K}$, say $\mathbb{Q}$. Consider n variables $x_1 < \cdots < x_n$.
- A subset $V \subset \mathbf{K}^n$ is a **(affine) variety over** $\mathbf{k}$ if there exists $F \subset \mathbf{k}[x_1, \cdots, x_n]$ such that $V = V(F)$ where

$$V(F) := \{z \in \mathbf{K}^n \mid f(z) = 0 \ \ (\forall\, f \in F)\}.$$

The variety V is **irreducible** if for all varieties $V_1, V_2 \subset \mathbf{K}^n$

$$V = V_1 \cup V_2 \quad \Rightarrow \quad V = V_1 \ \text{or} \ V = V_2.$$

How did regular chains emerge? (1/3)

- Let $\mathbf{K}$ be an algebraically closed field, say $\mathbb{C}$, and $\mathbf{k}$ be a subfield of $\mathbf{K}$, say $\mathbb{Q}$. Consider n variables $x_1 < \cdots < x_n$.
- A subset $V \subset \mathbf{K}^n$ is a **(affine) variety over k** if there exists $F \subset \mathbf{k}[x_1, \cdots, x_n]$ such that $V = V(F)$ where

$$V(F) := \{z \in \mathbf{K}^n \mid f(z) = 0 \ \ (\forall f \in F)\}.$$

  The variety V is **irreducible** if for all varieties $V_1, V_2 \subset \mathbf{K}^n$

$$V = V_1 \cup V_2 \quad \Rightarrow \quad V = V_1 \ \text{or} \ V = V_2.$$

- **Theorem** (E. Lasker) *For each variety $V \subset \mathbf{K}^n$ there exist finitely many irreducible varieties $V_1, \ldots, V_e \subset \mathbf{K}^n$* such that

$$V = V_1 \cup \cdots \cup V_e.$$

  Moreover, if $V_i \not\subseteq V_j$ for $1 \le i < j \le e$ then $\{V_1, \ldots, V_e\}$ is unique. This is the **irreducible decomposition of V**.

How did regular chains emerge? (2/3)

- **Theorem** (J.F. Ritt) *Let* $V \subset \mathbf{K}^n$ *be an irreducible non-empty variety and let* $F \subset \mathbf{k}[x_1, \cdots, x_n]$ *s.t.* $V = V(F)$*. Then, one can compute a (reduced) triangular set* $T \subset \langle F \rangle$ *s.t.*

  $$(\forall\ g \in \langle F \rangle)\ \ \mathrm{prem}(g, T) = 0.$$

  Combined with algebraic factorization one can (in theory) compute irreducible decompositions.

How did regular chains emerge? (2/3)

- **Theorem** (J.F. Ritt) *Let $V \subset \mathbf{K}^n$ be an* irreducible *non-empty variety and let $F \subset \mathbf{k}[x_1, \cdots, x_n]$ s.t. $V = V(F)$. Then, one can compute a (reduced) triangular set $T \subset \langle F \rangle$ s.t.*

$$(\forall\ g \in \langle F \rangle)\ \ \mathrm{prem}(g, T) = 0.$$

  Combined with algebraic factorization one can (in theory) compute irreducible decompositions.

- **Theorem** (W.T. Wu) *Let $V \subset \mathbf{K}^n$ be a variety and let $F \subset \mathbf{k}[x_1, \cdots, x_n]$ s.t. $V = V(F)$. Then, one can compute a (reduced) triangular set $T \subset \langle F \rangle$ s.t.*

$$(\forall\ g \in F)\ \ \mathrm{prem}(g, T) = 0.$$

  This leads to a factorization free *algorithm for decomposing varieties (but not into irreducible components).*

How did regular chains emerge? (3/3)

- **Example.** Applying the `charset` procedure to $F = \{x_2^2 - x_1, x_1 x_3^2 - 2x_2x_3 + 1, (x_2x_3 - 1)x_4^2 + x_2^2\}$ produces $T = F$. However $V(F) = \emptyset$. Indeed

$$x_1 x_3^2 - 2x_2x_3 + 1 \equiv (x_2x_3 - 1)^2 \mod x_2^2 - x_1.$$

Thus, the initial $(x_2x_3 - 1)$ is a zero-divisor modulo $\langle x_2^2 - x_1, x_1x_3^2 - 2x_2x_3 + 1\rangle$.

How did regular chains emerge? (3/3)

- **Example.** Applying the `charset` procedure to $F = \{x_2^2 - x_1, x_1x_3^2 - 2x_2x_3 + 1, (x_2x_3 - 1)x_4^2 + x_2^2\}$ produces $T = F$. However $V(F) = \emptyset$. Indeed

$$x_1x_3^2 - 2x_2x_3 + 1 \equiv (x_2x_3 - 1)^2 \mod x_2^2 - x_1.$$

Thus, the initial $(x_2x_3 - 1)$ is a zero-divisor modulo $\langle x_2^2 - x_1, x_1x_3^2 - 2x_2x_3 + 1\rangle$.

- The notion of a regular chain (Lu Yang, Jingzhong Zhang 91) (Michael Kalkbrener 91) solves this difficulty: for any input $F \subseteq \mathbf{k}[x_1, \ldots, x_n]$ one can compute regular chains $T_1, \ldots, T_e$ such that a point $z \in \mathbf{K}^n$ is a zero of F if and only if z is a zero of one of the $T_1, \ldots, T_e$ (in some technical sense). (Dong Ming Wang 2000) (Marc Moreno Maza 2000)

Outline

- Regular chains
- Normal Forms
- Regular GCDs
- Regularity test
- The `RegularChains` library

Outline

- Regular chains
- Normal Forms : using fast polynomial arithmetic
- Regular GCDs
- Regularity test
- The `RegularChains` library

Outline

- Regular chains
- Normal Forms : using fast polynomial arithmetic
- Regular GCDs : using modular techniques
- Regularity test
- The `RegularChains` library

Outline

- Regular chains
- Normal Forms : using fast polynomial arithmetic
- Regular GCDs : using modular techniques
- Regularity test : recycling intermediate computations
- The `RegularChains` library

Part I: The Notion of a Regular Chain

- Regular chain, saturated ideal
- Algorithmic properties
- Zero-dimensional case (as many equations as variables)

Regular Chain, Saturated Ideal

- Let $T \subset \mathbf{k}[x_1 < \cdots < x_n] \setminus \mathbf{k}$ be a triangular set, hence the polynomials of T have pairwise distinct main variables.

Regular Chain, Saturated Ideal

- Let $T \subset \mathbf{k}[x_1 < \cdots < x_n] \setminus \mathbf{k}$ be a triangular set, hence the polynomials of T have pairwise distinct main variables.

- Let $\mathrm{mvar}(T) := \{\mathrm{mvar}(t) \mid t \in T\}$, $\mathrm{init}(t) := \mathrm{lc}(t, \mathrm{mvar}(t))$ for all $t \in T$, and $h_T := \prod_{t \in T} \mathrm{init}(t)$.

Regular Chain, Saturated Ideal

- Let $T \subset \mathbf{k}[x_1 < \cdots < x_n] \setminus \mathbf{k}$ be a triangular set, hence the polynomials of T have pairwise distinct main variables.
- Let $\mathrm{mvar}(T) := \{\mathrm{mvar}(t) \mid t \in T\}$, $\mathrm{init}(t) := \mathrm{lc}(t, \mathrm{mvar}(t))$ for all $t \in T$, and $h_T := \prod_{t \in T} \mathrm{init}(t)$.
- T_v is the polynomial of T with main variable v, for $v \in \mathrm{mvar}(T)$, and $T_{<v} := \{t \in T \mid \mathrm{mvar}(t) < v\}$.

Regular Chain, Saturated Ideal

- Let $T \subset \mathbf{k}[x_1 < \cdots < x_n] \setminus \mathbf{k}$ be a triangular set, hence the polynomials of T have pairwise distinct main variables.
- Let $\mathrm{mvar}(T) := \{\mathrm{mvar}(t) \mid t \in T\}$, $\mathrm{init}(t) := \mathrm{lc}(t, \mathrm{mvar}(t))$ for all $t \in T$, and $h_T := \prod_{t \in T} \mathrm{init}(t)$.
- T_v is the polynomial of T with main variable v, for $v \in \mathrm{mvar}(T)$, and $T_{<v} := \{t \in T \mid \mathrm{mvar}(t) < v\}$.
- The quasi-component of T is $W(T) = V(T) \setminus V(h_T)$.

Regular Chain, Saturated Ideal

- Let $T \subset \mathbf{k}[x_1 < \cdots < x_n] \setminus \mathbf{k}$ be a triangular set, hence the polynomials of T have pairwise distinct main variables.
- Let $\mathrm{mvar}(T) := \{\mathrm{mvar}(t) \mid t \in T\}$, $\mathrm{init}(t) := \mathrm{lc}(t, \mathrm{mvar}(t))$ for all $t \in T$, and $h_T := \prod_{t \in T} \mathrm{init}(t)$.
- T_v is the polynomial of T with main variable v, for $v \in \mathrm{mvar}(T)$, and $T_{<v} := \{t \in T \mid \mathrm{mvar}(t) < v\}$.
- The quasi-component of T is $W(T) = V(T) \setminus V(h_T)$.
- The saturated ideal of T is the ideal of $\mathbf{k}[x_1 < \cdots < x_n]$

$$\mathrm{sat}(T) := \langle T \rangle : (h_T)^{\infty}.$$

Regular Chain, Saturated Ideal

- Let $T \subset \mathbf{k}[x_1 < \cdots < x_n] \setminus \mathbf{k}$ be a triangular set, hence the polynomials of T have pairwise distinct main variables.
- Let $\mathrm{mvar}(T) := \{\mathrm{mvar}(t) \mid t \in T\}$, $\mathrm{init}(t) := \mathrm{lc}(t, \mathrm{mvar}(t))$ for all $t \in T$, and $h_T := \prod_{t \in T} \mathrm{init}(t)$.
- T_v is the polynomial of T with main variable v, for $v \in \mathrm{mvar}(T)$, and $T_{<v} := \{t \in T \mid \mathrm{mvar}(t) < v\}$.
- The quasi-component of T is $W(T) = V(T) \setminus V(h_T)$.
- The saturated ideal of T is the ideal of $\mathbf{k}[x_1 < \cdots < x_n]$

$$\mathrm{sat}(T) := \langle T \rangle : (h_T)^\infty.$$

- T is a regular chain if for each $v \in \mathrm{mvar}(T)$ the initial of T_v is regular modulo $\mathrm{sat}(T_{<v})$ (Michael Kalkbrener 91).

Algorithmic Properties

- Let $p \in \mathbf{k}[x_1 < \cdots < x_n]$ and $T \subset \mathbf{k}[x_1 < \cdots < x_n]$ be a triangular set. If T is empty then, the *iterated resultant* of p w.r.t. T is $\mathrm{res}(T, p) = p$. Otherwise, writing $T = T_{<w} \cup T_w$

$$\mathrm{res}(T, p) = \begin{cases} p & \text{if } \deg(p, w) = 0 \\ \mathrm{res}(T_{<w}, \mathrm{res}(T_w, p, w)) & \text{otherwise} \end{cases}$$

Algorithmic Properties

- Let $p \in \mathbf{k}[x_1 < \cdots < x_n]$ and $T \subset \mathbf{k}[x_1 < \cdots < x_n]$ be a triangular set. If T is empty then, the *iterated resultant* of p w.r.t. T is $\mathrm{res}(T, p) = p$. Otherwise, writing $T = T_{<w} \cup T_w$

$$\mathrm{res}(T, p) = \begin{cases} p & \text{if } \deg(p, w) = 0 \\ \mathrm{res}(T_{<w}, \mathrm{res}(T_w, p, w)) & \text{otherwise} \end{cases}$$

- T is a regular chain iff

$$\mathrm{res}(T, h_T) \neq 0$$

(Lu Yang, Jingzhong Zhang 91).

Algorithmic Properties

- Let $p \in \mathbf{k}[x_1 < \cdots < x_n]$ and $T \subset \mathbf{k}[x_1 < \cdots < x_n]$ be a triangular set. If T is empty then, the *iterated resultant* of p w.r.t. T is $\mathrm{res}(T, p) = p$. Otherwise, writing $T = T_{<w} \cup T_w$

$$\mathrm{res}(T, p) \quad = \quad \begin{cases} p & \text{if } \deg(p, w) = 0 \\ \mathrm{res}(T_{<w}, \mathrm{res}(T_w, p, w)) & \text{otherwise} \end{cases}$$

- T is a regular chain iff

$$\mathrm{res}(T, h_T) \neq 0$$

(Lu Yang, Jingzhong Zhang 91).

- T is a regular chain iff

$$\{p \mid \mathrm{prem}(p, T) = 0\} \quad = \quad \mathrm{sat}(T)$$

(Philippe Aubry, Daniel Lazard, Marc Moreno Maza 97).

Zero-dimensional Regular Chains

- Let $T \subset \mathbf{k}[x_1, \ldots, x_n]$ be a regular chain such that $|T| = n$.

Zero-dimensional Regular Chains

- Let $T \subset \mathbf{k}[x_1, \ldots, x_n]$ be a regular chain such that $|T| = n$.
- Then each $\mathrm{init}(t)$ for $t \in T$ is invertible modulo $\langle T \rangle$ (using GCD computations)

Zero-dimensional Regular Chains

- Let $T \subset \mathbf{k}[x_1, \ldots, x_n]$ be a regular chain such that $|T| = n$.
- Then each $\mathrm{init}(t)$ for $t \in T$ is invertible modulo $\langle T \rangle$ (using GCD computations)
- Thus $\mathrm{sat}(T) = \langle T \rangle$ (consider the primary components of $\langle T \rangle$)

Zero-dimensional Regular Chains

- Let $T \subset \mathbf{k}[x_1, \ldots, x_n]$ be a regular chain such that $|T| = n$.
- Then each $\mathrm{init}(t)$ for $t \in T$ is invertible modulo $\langle T \rangle$ (using GCD computations)
- Thus $\mathrm{sat}(T) = \langle T \rangle$ (consider the primary components of $\langle T \rangle$)
- Let N be the regular chain obtained from T by normalization: multiplying each $t \in T$ by the inverse of $\mathrm{init}(t)$ modulo $\langle T \rangle$.

Zero-dimensional Regular Chains

- Let $T \subset \mathbf{k}[x_1, \ldots, x_n]$ be a regular chain such that $|T| = n$.
- Then each $\mathrm{init}(t)$ for $t \in T$ is invertible modulo $\langle T \rangle$ (using GCD computations)
- Thus $\mathrm{sat}(T) = \langle T \rangle$ (consider the primary components of $\langle T \rangle$)
- Let N be the regular chain obtained from T by normalization: multiplying each $t \in T$ by the inverse of $\mathrm{init}(t)$ modulo $\langle T \rangle$.
- Let G be the regular chain obtained from N by auto-reduction in Gröbner basis sense. Then G is a reduced Gröbner basis.

Zero-dimensional Regular Chains

- Let $T \subset \mathbf{k}[x_1, \ldots, x_n]$ be a regular chain such that $|T| = n$.
- Then each $\mathrm{init}(t)$ for $t \in T$ is invertible modulo $\langle T \rangle$ (using GCD computations)
- Thus $\mathrm{sat}(T) = \langle T \rangle$ (consider the primary components of $\langle T \rangle$)
- Let N be the regular chain obtained from T by normalization: multiplying each $t \in T$ by the inverse of $\mathrm{init}(t)$ modulo $\langle T \rangle$.
- Let G be the regular chain obtained from N by auto-reduction in Gröbner basis sense. Then G is a reduced Gröbner basis.
- Example:
  $T = \{x_1^2 + 1, x_1 x_2^2 + 1\} \quad \Rightarrow \quad G = \{x_1^2 + 1, x_2^2 - x_1\}$.

Zero-dimensional Regular Chains

- Let $T \subset \mathbf{k}[x_1, \ldots, x_n]$ be a regular chain such that $|T| = n$.
- Then each $\mathrm{init}(t)$ for $t \in T$ is invertible modulo $\langle T \rangle$ (using GCD computations)
- Thus $\mathrm{sat}(T) = \langle T \rangle$ (consider the primary components of $\langle T \rangle$)
- Let N be the regular chain obtained from T by normalization: multiplying each $t \in T$ by the inverse of $\mathrm{init}(t)$ modulo $\langle T \rangle$.
- Let G be the regular chain obtained from N by auto-reduction in Gröbner basis sense. Then G is a reduced Gröbner basis.
- Example:
  $T = \{x_1^2 + 1, x_1 x_2^2 + 1\} \quad \Rightarrow \quad G = \{x_1^2 + 1, x_2^2 - x_1\}$.
- **Unless k is finite, normalization blows up coefficients.**

Part II: Normal Forms

- Ideal membership, normal form computation
- The fast division trick
- FFT-based multiplication
- Fast Normal form computation

Ideal membership, normal form computation

- Let $T \subset \mathbf{k}[x_1 < \cdots < x_n]$ be a regular chain s.t. $|T| = n$, $h_T = 1$ and T is auto-reduced. Hence T is a Gröbner basis.

Ideal membership, normal form computation

- Let $T \subset \mathbf{k}[x_1 < \cdots < x_n]$ be a regular chain s.t. $|T| = n$, $h_T = 1$ and T is auto-reduced. Hence T is a Gröbner basis.
- For $p \in \mathbf{k}[x_1, \ldots, x_n]$, we want to compute NormalForm(p, T) as fast as possible.

Ideal membership, normal form computation

- Let $T \subset \mathbf{k}[x_1 < \cdots < x_n]$ be a regular chain s.t. $|T| = n$, $h_T = 1$ and T is auto-reduced. Hence T is a Gröbner basis.
- For $p \in \mathbf{k}[x_1, \ldots, x_n]$, we want to compute $\mathrm{NormalForm}(p, T)$ as fast as possible.
- A *descending* approach

$$\mathrm{rem}(\mathrm{rem}(\cdots, \mathrm{rem}(p, T_{x_n}), \cdots, T_{x_2}), T_{x_1})$$

blows up intermediate expression

Ideal membership, normal form computation

- Let $T \subset \mathbf{k}[x_1 < \cdots < x_n]$ be a regular chain s.t. $|T| = n$, $h_T = 1$ and T is auto-reduced. Hence T is a Gröbner basis.
- For $p \in \mathbf{k}[x_1, \ldots, x_n]$, we want to compute $\mathrm{NormalForm}(p, T)$ as fast as possible.
- A *descending* approach

$$\mathrm{rem}(\mathrm{rem}(\cdots, \mathrm{rem}(p, T_{x_n}), \cdots, T_{x_2}), T_{x_1})$$

blows up intermediate expression

- A *naive ascending* approach

$$\mathrm{rem}(\cdots, \mathrm{rem}(\mathrm{rem}(\mathrm{rem}(p, T_{x_1}), T_{x_2}), T_{x_1}), \cdots T_{x_1})$$

blows up algebraic complexity

The fast division trick (1/2)

- Let $a, b \in \mathbb{A}[x]$ with $n := \deg(a) \geq m := \deg(b) > 0$, b monic and $\mathbb{A}$ any commutative ring with 1.

The fast division trick (1/2)

- Let $a, b \in \mathbb{A}[x]$ with $n := \deg(a) \geq m := \deg(b) > 0$, b monic and $\mathbb{A}$ any commutative ring with 1.
- We want the quotient q and the remainder r of a w.r.t. b:

$$a(x) \;=\; q(x)\, b(x) + r(x)$$

The fast division trick (1/2)

- Let $a, b \in \mathbb{A}[x]$ with $n := \deg(a) \geq m := \deg(b) > 0$, b monic and $\mathbb{A}$ any commutative ring with 1.
- We want the quotient q and the remainder r of a w.r.t. b:
$$a(x) \; = \; q(x)\, b(x) + r(x)$$
- Replacing x by $1/x$ and multiplying the equation by x^n:
$$x^n\, a(1/x) \; = \; \left(x^{n-m} q(1/x)\right)\, \left(x^m\, b(1/x)\right) + x^{n-m+1}\, \left(x^{m-1}\, r(1/x)\right)$$
That is:
$$\mathrm{rev}_n(a) \; = \; \mathrm{rev}_{n-m}(q)\, \mathrm{rev}_m(b) + x^{n-m+1}\, \mathrm{rev}_{m-1}(r)$$

The fast division trick (1/2)

- Let $a, b \in \mathbb{A}[x]$ with $n := \deg(a) \geq m := \deg(b) > 0$, b monic and $\mathbb{A}$ any commutative ring with 1.
- We want the quotient q and the remainder r of a w.r.t. b:
$$a(x) \;=\; q(x)\, b(x) + r(x)$$
- Replacing x by $1/x$ and multiplying the equation by x^n:
$$x^n\, a(1/x) \;=\; \left(x^{n-m} q(1/x)\right)\, \left(x^m\, b(1/x)\right) + x^{n-m+1}\, \left(x^{m-1}\, r(1/x)\right)$$
That is:
$$\mathrm{rev}_n(a) \;=\; \mathrm{rev}_{n-m}(q)\, \mathrm{rev}_m(b) + x^{n-m+1}\, \mathrm{rev}_{m-1}(r)$$
- Computing $(\mathrm{rev}_m(b))^{-1} \mod x^{n-m+1}$ is a truncated inverse of a power series. (S. Cook, 1966) (H. T. Kung, 1974) and (M. Sieveking, 1972)

The fast division trick (2/2)

Input: $f \in \mathbb{A}[x]$ such that $f(0) = 1$ and $\ell \in \mathcal{N}$.

Output: $g \in \mathbb{A}[x]$ such that $f\,g \equiv 1 \mod x^\ell$

$g_0 := 1$
$r := \lceil \log_2(\ell) \rceil$
for $i = 1 \cdots r$ **repeat**
  $g_i := \left(2g_{i-1} - f\,{g_{i-1}}^2\right) \mod x^{2^i}$
return g_r

- This algorithm runs in $3\mathrm{M}(\ell) + 0(\ell)$ operations in $\mathbb{A}$.
- Improved versions run in $2\,\mathrm{M}(\ell) + O(\ell)$ operations in $\mathbb{A}$.
- Finally, the quotient q and the remainder r are computed in $3\,\mathrm{M}(n-m) + \mathrm{M}(\max(n-m, m)) + \mathcal{O}(n)$ operations in $\mathbb{A}$
- *Modern Computed Algebra* (Gathen Gerhard 99)

FFT-based multiplication

M(d) number of coefficient operations in degree less than d.

Classical Multiplication	$M(d) = 2d^2$
Karatsuba Multiplication	$M(d) = 9d^{1.59}$
FFT over appropriate ring	$M(d) = 9/2d \log d + 3\,d$

Input: $f, g \in \mathbf{k}[x]$ and ω a s-primitive root of unity for $s > \deg(f) + \deg(g)$ and s is a power of 2.

Output: the product fg

(1) Evaluate f and g at ω^i for $i = 0 \cdots s-1$

(2) Evaluate fg at ω^i for $i = 0 \cdots s-1$

(3) Interpolate and **return** fg

See (M.M.M. Yuzhen Xie 2009) for implementation techniques.

Fast Normal form computation (1/3)

- Let A and B in $\mathbf{k}[x_1, \ldots, x_n]$ reduced w.r.t. $T := \{T_1, \ldots, T_n\}$ 0-dimensional, reduced and all $\mathrm{init}(T_i) = 1$.

Fast Normal form computation (1/3)

- Let A and B in $\mathbf{k}[x_1, \ldots, x_n]$ reduced w.r.t. $T := \{T_1, \ldots, T_n\}$ 0-dimensional, reduced and all $\mathrm{init}(T_i) = 1$.
- The size of input is $\delta_{\mathbf{T}} = \deg(T_1, x_1) \cdots \deg(T_n, x_n)$.

Fast Normal form computation (1/3)

- Let A and B in $\mathbf{k}[x_1, \ldots, x_n]$ reduced w.r.t. $T := \{T_1, \ldots, T_n\}$ 0-dimensional, reduced and all $\mathrm{init}(T_i) = 1$.
- The size of input is $\delta_{\mathbf{T}} = \deg(T_1, x_1) \cdots \deg(T_n, x_n)$.
- One can compute $A\,B \mod \langle T_1, \ldots, T_n\}$ in $O\tilde{~}(4^n \delta_{\mathbf{T}})$ operations in $\mathbf{k}$ (Xin Li, M.M.M., É. Schost 07).

Fast Normal form computation (1/3)

- Let A and B in $\mathbf{k}[x_1, \ldots, x_n]$ reduced w.r.t. $T := \{T_1, \ldots, T_n\}$ 0-dimensional, reduced and all $\mathrm{init}(T_i) = 1$.
- The size of input is $\delta_{\mathbf{T}} = \deg(T_1, x_1) \cdots \deg(T_n, x_n)$.
- One can compute $A\,B \mod \langle T_1, \ldots, T_n\}$ in $O\tilde{~}(4^n \delta_{\mathbf{T}})$ operations in $\mathbf{k}$ (Xin Li, M.M.M., É. Schost 07).
- Three key ideas: using the fast division trick and avoid mod $\langle T_1, \ldots, T_n\}$ as much as possible and reduce to multiplying polynomials over the base field $\mathbf{k}$ using FFT.

ModMul($A, B, \{T_1, \ldots, T_n\}$)
1 $D := AB$ computed in $\mathbf{k}[x_1, \ldots, x_n]$
2 **return** $\mathrm{NormalForm}_n(D, \{T_1, \ldots, T_n\})$

Fast Normal form computation (2/3)

$\underline{\mathrm{NormalForm}_1(A : R[x_1], \{T_1 : R[x_1]\})}$

$S_1 := \mathrm{Rev}(T_1)^{-1} \mod x_1^{\deg(A)-\deg(T_1)+1}$

$D := \mathrm{Rev}(A) S_1 \mod x_1^{\deg(A)-\deg(T_1)+1}$

$D := T_1 \, \mathrm{Rev}(D)$

return $A - D$

$\underline{\mathrm{NormalForm}_2(A : R[x_1, x_2], \{T_1 : R[x_1], T_2 : R[x_1, x_2]\})}$

$A := \mathrm{map}(\mathrm{NormalForm}_1, \mathrm{Coeffs}(A, x_2), \{T_1\})$

$S_2 := \mathrm{Rev}(T_2)^{-1} \mod T_1, x_2^{\deg(A,x_2)-\deg(T_2,x_2)+1}$

$D := \mathrm{Rev}(A) S_2 \mod x_2^{\deg(A,x_2)-\deg(T_2,x_2)+1}$

$D := \mathrm{map}(\mathrm{NormalForm}_1, \mathrm{Coeffs}(D, x_2), \{T_1\})$

$D := T_2 \, \mathrm{Rev}(D)$

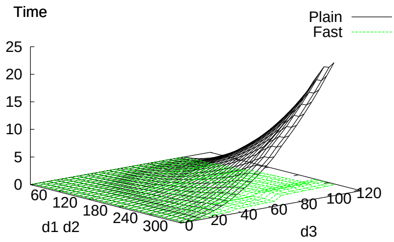
$D := \mathrm{map}(\mathrm{NormalForm}_1, \mathrm{Coeffs}(D, x_2), \{T_1\})$

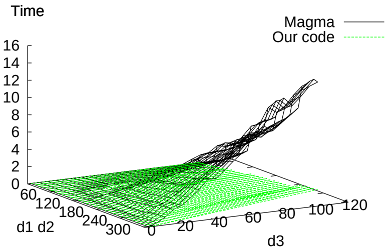
return $A - D$

Fast Normal form computation (3/3)

[left] comparison of classical (plain) and asymptotically fast strategies.

[right] comparison with MAGMA.





- Asymptotically fast strategy dominates the classical one.
- Our fast implementation is better than Magma's one (the best known implementation).

Part III: Regular GCDs

- Plane curve intersection
- The notion of a regular GCD
- Subresultants
- Regular GCDs via subresultants
- Complexity estimates
- Experimental results

Plane curve intersection

A historical application of the resultant is to compute the intersection of two plane curves. Up to details, there are two steps:

- eliminate one variable by computing a resultant,
- compute a GCD modulo this resultant.

Example (From *Modern Computer Algebra*, Chapter 6)

. Let $P = \left(y^2+6\right)(x-1) - y\left(x^2+1\right)$ and
$Q = \left(x^2+6\right)(y-1) - x\left(y^2+1\right)$

- $\mathrm{res}(\mathrm{P}, \mathrm{Q}, \mathrm{y}) = 2\,\left(\mathrm{x}^2 - \mathrm{x} + 4\right)(\mathrm{x}-2)^2(\mathrm{x}-3)^2.$
- $\mathrm{gcd}(\mathrm{P}, \mathrm{Q}, \mathrm{x}-2=0) = (\mathrm{y}-2)(\mathrm{y}-3).$
- $\mathrm{gcd}(\mathrm{P}, \mathrm{Q}, \mathrm{x}-3=0) = (\mathrm{y}-2)(\mathrm{y}-3).$
- $\mathrm{gcd}(\mathrm{P}, \mathrm{Q}, \mathrm{x}^2-\mathrm{x}+4=0) = (2\,\mathrm{x}-1)\,\mathrm{y} - 7 - \mathrm{x}.$

Regular GCD

- Let $\mathbb{B}$ be a commutative ring with units. Let $P, Q \in \mathbb{B}[y]$ be non-constant with regular leading coefficients.
- $G \in \mathbb{B}[y]$ is a *regular GCD* of P, Q if we have:
  - (i) $\mathrm{lc}(G, y)$ is a regular element of $\mathbb{B}$,
  - (ii) $G \in \langle P, Q \rangle$ in $\mathbb{B}[y]$,
  - (iii) $\deg(G, y) > 0 \quad \Rightarrow \quad \mathrm{prem}(P, G, y) = \mathrm{prem}(Q, G, y) = 0.$

Regular GCD

- Let $\mathbb{B}$ be a commutative ring with units. Let $P, Q \in \mathbb{B}[y]$ be non-constant with regular leading coefficients.

- $G \in \mathbb{B}[y]$ is a *regular GCD* of P, Q if we have:
  - (i) $\mathrm{lc}(G, y)$ is a regular element of $\mathbb{B}$,
  - (ii) $G \in \langle P, Q \rangle$ in $\mathbb{B}[y]$,
  - (iii) $\deg(G, y) > 0 \quad \Rightarrow \quad \mathrm{prem}(P, G, y) = \mathrm{prem}(Q, G, y) = 0.$

- In practice $\mathbb{B} = \mathbf{k}[x_1, \ldots, x_n]/\mathrm{sat}(T)$, with T being a regular chain.

Regular GCD

- Let $\mathbb{B}$ be a commutative ring with units. Let $P, Q \in \mathbb{B}[y]$ be non-constant with regular leading coefficients.

- $G \in \mathbb{B}[y]$ is a *regular GCD* of P, Q if we have:
  - (i) $\mathrm{lc}(G, y)$ is a regular element of $\mathbb{B}$,
  - (ii) $G \in \langle P, Q \rangle$ in $\mathbb{B}[y]$,
  - (iii) $\deg(G, y) > 0 \quad \Rightarrow \quad \mathrm{prem}(P, G, y) = \mathrm{prem}(Q, G, y) = 0$.

- In practice $\mathbb{B} = \mathbf{k}[x_1, \ldots, x_n]/\mathrm{sat}(T)$, with T being a regular chain.

- Such a regular GCD may not exist. However one can compute $\mathcal{I}_i = \mathrm{sat}(T_i)$ and non-zero polynomials G_i such that

$$\sqrt{\mathcal{I}} = \cap_{i=0}^{e} \sqrt{\mathcal{I}_i} \quad \text{and} \quad G_i \quad \text{regular GCD of} \quad P, Q \bmod \mathcal{I}_i$$

Regularity test

- Regularity test is a fundamental operation:

$$\mathrm{Regularize}(p, \mathcal{I}) \quad \longmapsto \quad (\mathcal{I}_1, \ldots, \mathcal{I}_e)$$

  such that:

$$\sqrt{\mathcal{I}} = \cap_{i=0}^{e} \sqrt{\mathcal{I}_i} \;\text{ and }\; p \in \mathcal{I}_i \text{ or } p \text{ regular modulo } \mathcal{I}_i$$

- Regularity test reduces to regular GCD computation.

Related work

- This notion of a regular GCD was proposed in (M. M. 2000)
- In previous work (Kalkbrener 1993) and (Rioboo & M. M. 1995), other regular GCDs modulo regular chains were introduced, but with limitations.
- In other work (Wang 2000), (Yang etc. 1995) and (Jean Della Dora, Claire Dicrescenzo, Dominique Duval 85), related techniques are used to construct triangular decompositions.
- Regular GCDs modulo regular chains generalize GCDs over towers of field extensions for which specialized algorithms are available, (van Hoeij and Monagan 2002 & 2004).
- Asymptotically fast algorithms (when $\mathrm{sat}(T)$ is zero-dimensional and radical) appear in (Xavier Dahan, M. M. , Éric Schost, Yuzhen Xie, 2006)
- The next results appear in (Xin Li, M. M. , Wei Pan, 2009).

Subresultants (1/2)

- Let $P, Q \in (\mathbf{k}[x_1])[x_2]$ with $\deg(P, x_2) > \deg(Q, x_2)$.

Subresultants (1/2)

- Let $P, Q \in (\mathbf{k}[x_1])[x_2]$ with $\deg(P, x_2) > \deg(Q, x_2)$.
- The polynomials computed by $\mathsf{SubresultantPRS}(P, Q)$ form a sequence, denoted by $\mathsf{Chain}(P, Q)$, starting at Q and ending at $\mathrm{res}(P, Q, x_2)$

Subresultants (1/2)

- Let $P, Q \in (\mathbf{k}[x_1])[x_2]$ with $\deg(P, x_2) > \deg(Q, x_2)$.
- The polynomials computed by SubresultantPRS(P, Q) form a sequence, denoted by Chain(P, Q), starting at Q and ending at $\mathrm{res}(P, Q, x_2)$
- This chain contains $\deg(Q, x_2) + 1$ polynomials for each $j \in \deg(Q, x_2) \cdots 0$, the polynomial of index j is called the subresultant of index j, denoted by S_j.

Subresultants (1/2)

- Let $P, Q \in (\mathbf{k}[x_1])[x_2]$ with $\deg(P, x_2) > \deg(Q, x_2)$.
- The polynomials computed by SubresultantPRS(P, Q) form a sequence, denoted by Chain(P, Q), starting at Q and ending at $\mathrm{res}(P, Q, x_2)$
- This chain contains $\deg(Q, x_2) + 1$ polynomials for each $j \in \deg(Q, x_2) \cdots 0$, the polynomial of index j is called the subresultant of index j, denoted by S_j.
- The coefficients of $S_j \in \mathbf{k}[x_1]$ are minors of the Sylvester Matrix.

Subresultants (1/2)

- Let $P, Q \in (\mathbf{k}[x_1])[x_2]$ with $\deg(P, x_2) > \deg(Q, x_2)$.
- The polynomials computed by SubresultantPRS(P, Q) form a sequence, denoted by Chain(P, Q), starting at Q and ending at $\operatorname{res}(P, Q, x_2)$
- This chain contains $\deg(Q, x_2) + 1$ polynomials for each $j \in \deg(Q, x_2) \cdots 0$, the polynomial of index j is called the subresultant of index j, denoted by S_j.
- The coefficients of $S_j \in \mathbf{k}[x_1]$ are minors of the Sylvester Matrix.
- If $S_j \neq 0$, then $\deg(S_j) \leq j$. If $\deg(S_j) = j$ then S_j is said non-defective, otherwise it is said defective.

Subresultants (1/2)

- Let $P, Q \in (\mathbf{k}[x_1])[x_2]$ with $\deg(P, x_2) > \deg(Q, x_2)$.
- The polynomials computed by SubresultantPRS(P, Q) form a sequence, denoted by Chain(P, Q), starting at Q and ending at $\mathrm{res}(P, Q, x_2)$
- This chain contains $\deg(Q, x_2) + 1$ polynomials for each $j \in \deg(Q, x_2) \cdots 0$, the polynomial of index j is called the subresultant of index j, denoted by S_j.
- The coefficients of $S_j \in \mathbf{k}[x_1]$ are minors of the Sylvester Matrix.
- If $S_j \neq 0$, then $\deg(S_j) \leq j$. If $\deg(S_j) = j$ then S_j is said non-defective, otherwise it is said defective.
- (Chee K. Yap 1993) (Lionel Ducos 1997) (M'hammed El Kahoui, 2003)

Subresultants (2/2)

- **Example.**
  The Chain of $P = X_2^4 + X_1X_2 + 1$ and $Q = 4X_2^3 + X_1$ in $(\mathbb{Q}[X_1])[X_2]$ produces the following sequence of polynomials:

$$\begin{aligned}
S_4 &= X_2^4 + X_1X_2 + 1 \\
S_3 &= 4X_2^3 + X_1 \\
S_2 &= -4(3X_1X_2 + 4) \\
S_1 &= -12X_1(3X_1X_2 + 4) \\
S_0 &= -27X_1^4 + 256
\end{aligned}$$

Subresultants (2/2)

- **Example.**
  The Chain of $P = X_2^4 + X_1X_2 + 1$ and $Q = 4X_2^3 + X_1$ in $(\mathbb{Q}[X_1])[X_2]$ produces the following sequence of polynomials:

$$\begin{aligned} S_4 &= X_2^4 + X_1X_2 + 1 \\ S_3 &= 4X_2^3 + X_1 \\ S_2 &= -4(3X_1X_2 + 4) \\ S_1 &= -12X_1(3X_1X_2 + 4) \\ S_0 &= -27X_1^4 + 256 \end{aligned}$$

- Let Φ be a homomorphism from $\mathbf{k}[x_1, x_2]$ to $\mathbf{K}[x_2]$. Assume $\Phi(a) \neq 0$ where $a = \mathrm{lc}(P, X_2)$. Then we have the specialization property of subresultants:

$$\Phi(sres_i(P, Q)) = \Phi(a)^{n-k} sres_i(\Phi(P), \Phi(Q))$$

  where $n = \deg(Q, x_2)$ and $k = \deg(\Phi(Q), x_2)$.

Regular GCDs (1/6)

- Let $P, Q \in \mathbf{k}[\mathbf{x}][y]$ with $\mathrm{mvar}(P) = \mathrm{mvar}(Q) = y$.
- Define $R = \mathrm{res}(P, Q, y)$.

Regular GCDs (1/6)

- Let $P, Q \in \mathbf{k}[\mathbf{x}][y]$ with $\mathrm{mvar}(P) = \mathrm{mvar}(Q) = y$.
- Define $R = \mathrm{res}(P, Q, y)$.
- Let $T \subset \mathbf{k}[x_1, \ldots, x_n]$ be a regular chain such that
  - $R \in \mathrm{sat}(T)$,
  - $\mathrm{init}(P)$ and $\mathrm{init}(Q)$ are regular modulo $\mathrm{sat}(T)$.

Regular GCDs (1/6)

- Let $P, Q \in \mathbf{k}[\mathbf{x}][y]$ with $\mathrm{mvar}(P) = \mathrm{mvar}(Q) = y$.
- Define $R = \mathrm{res}(P, Q, y)$.
- Let $T \subset \mathbf{k}[x_1, \ldots, x_n]$ be a regular chain such that
  - $R \in \mathrm{sat}(T)$,
  - $\mathrm{init}(P)$ and $\mathrm{init}(Q)$ are regular modulo $\mathrm{sat}(T)$.
- $\mathbb{A} = \mathbf{k}[x_1, \ldots, x_n]$ and $\mathbb{B} = \mathbf{k}[x_1, \ldots, x_n]/\mathrm{sat}(T)$.
- For $0 \leq j \leq \mathrm{mdeg}(Q)$, we write S_j for the j-th subresultant of P, Q in $\mathbb{A}[y]$.

Regular GCDs (1/6)

- Let $P, Q \in \mathbf{k}[\mathbf{x}][y]$ with $\mathrm{mvar}(P) = \mathrm{mvar}(Q) = y$.
- Define $R = \mathrm{res}(P, Q, y)$.
- Let $T \subset \mathbf{k}[x_1, \ldots, x_n]$ be a regular chain such that
  - $R \in \mathrm{sat}(T)$,
  - $\mathrm{init}(P)$ and $\mathrm{init}(Q)$ are regular modulo $\mathrm{sat}(T)$.
- $\mathbb{A} = \mathbf{k}[x_1, \ldots, x_n]$ and $\mathbb{B} = \mathbf{k}[x_1, \ldots, x_n]/\mathrm{sat}(T)$.
- For $0 \leq j \leq \mathrm{mdeg}(Q)$, we write S_j for the j-th subresultant of P, Q in $\mathbb{A}[y]$.
- Recall that S_d regular GCD of P, Q modulo $\mathrm{sat}(T)$ means
  (*i*) $\mathrm{lc}(S_d, y)$ is a regular element of $\mathbb{B}$,
  (*ii*) $S_d \in \langle P, Q \rangle$ in $\mathbb{B}[y]$,
  (*iii*) $\deg(S_d, y) > 0 \;\Rightarrow\; \mathrm{prem}(P, S_d, y) = \mathrm{prem}(Q, S_d, y) = 0$.

Regular GCDs (2/6)

- Let $1 \leq d \leq q$ such that $S_j \in \mathrm{sat}(T)$ for all $0 \leq j < d$.

Lemma

If $\mathrm{lc}(S_d, y)$ is regular modulo $\mathrm{sat}(T)$, then S_d is non-defective over $\mathbf{k}[\mathbf{x}]$.

Regular GCDs (2/6)

- Let $1 \leq d \leq q$ such that $S_j \in \mathrm{sat}(T)$ for all $0 \leq j < d$.

Lemma

If $\mathrm{lc}(S_d, y)$ *is regular modulo* $\mathrm{sat}(T)$*, then* S_d *is non-defective over* $\mathbf{k}[\mathbf{x}]$.

- Consequently, S_d is the last nonzero subresultant over $\mathbb{B}$, and it is also non-defective over $\mathbb{B}$.

Regular GCDs (2/6)

- Let $1 \leq d \leq q$ such that $S_j \in \mathrm{sat}(T)$ for all $0 \leq j < d$.

Lemma
If $\mathrm{lc}(S_d, y)$ *is regular modulo* $\mathrm{sat}(T)$, *then* S_d *is non-defective over* $\mathbf{k}[\mathbf{x}]$.

- Consequently, S_d is the last nonzero subresultant over $\mathbb{B}$, and it is also non-defective over $\mathbb{B}$.
- If $\mathrm{lc}(S_d, x_n)$ is not regular modulo $\mathrm{sat}(T)$ then S_d may be defective over $\mathbb{B}$.

Regular GCDs (3/6)

- Let $1 \leq d \leq q$ such that $S_j \in \mathrm{sat}(T)$ for all $0 \leq j < d$.

Lemma

If $\mathrm{lc}(S_d, y)$ *is in* $\mathrm{sat}(T)$*, then* S_d *is nilpotent modulo* $\mathrm{sat}(T)$*.*

Regular GCDs (3/6)

- Let $1 \leq d \leq q$ such that $S_j \in \mathrm{sat}(T)$ for all $0 \leq j < d$.

Lemma

If $\mathrm{lc}(S_d, y)$ *is in* $\mathrm{sat}(T)$, *then* S_d *is nilpotent modulo* $\mathrm{sat}(T)$.

- Up to sufficient splitting of $\mathrm{sat}(T)$, S_d will vanish on all the components of $\mathrm{sat}(T)$.

Regular GCDs (3/6)

- Let $1 \leq d \leq q$ such that $S_j \in \mathrm{sat}(T)$ for all $0 \leq j < d$.

Lemma

If $\mathrm{lc}(S_d, y)$ is in $\mathrm{sat}(T)$, then S_d is nilpotent modulo $\mathrm{sat}(T)$.

- Up to sufficient splitting of $\mathrm{sat}(T)$, S_d will vanish on all the components of $\mathrm{sat}(T)$.
- The above two lemmas completely characterize the last non-zero subresultant of P and Q over $\mathbb{B}$.

Regular GCDs (4/6)

Example

- Consider P and Q in $\mathbb{Q}[x_1, x_2][y]$:

$$P = x_2^2 y^2 - x_1^4 \ \text{and} \ Q = x_1^2 y^2 - x_2^4.$$

- We have:

$$S_1 = x_1^6 - x_2^6 \ \text{and} \ R = (x_1^6 - x_2^6)^2.$$

- Let $T = \{R\}$. Then we observe:
  - The last subresultant of P, Q modulo $\mathrm{sat}(T)$ is S_1, which is a defective one.
  - S_1 is nilpotent modulo $\mathrm{sat}(T)$.
- P and Q do not admit a regular GCD over $\mathbb{Q}[x_1, x_2]/\mathrm{sat}(T)$.

Regular GCDs (5/6)

- Let $1 \leq d \leq q$ such that $S_j \in \mathrm{sat}(T)$ for all $0 \leq j < d$.

Proposition

Assume

- $\mathrm{lc}(S_d, y)$ *is regular modulo* $\mathrm{sat}(T)$,
- $\mathrm{sat}(T)$ *is radical.*

Then, S_d *is a regular GCD of* P, Q *modulo* $\mathrm{sat}(T)$.

Regular GCDs (5/6)

- Let $1 \leq d \leq q$ such that $S_j \in \mathrm{sat}(T)$ for all $0 \leq j < d$.

Proposition

Assume

- $\mathrm{lc}(S_d, y)$ *is regular modulo* $\mathrm{sat}(T)$,
- $\mathrm{sat}(T)$ *is radical.*

Then, S_d *is a regular GCD of* P, Q *modulo* $\mathrm{sat}(T)$.

Recall that S_d regular GCD of P, Q modulo $\mathrm{sat}(T)$ means

(i) $\mathrm{lc}(S_d, y)$ is a regular element of $\mathbb{B}$,
(ii) $S_d \in \langle P, Q \rangle$ in $\mathbb{B}[y]$,
(iii) $\deg(S_d, y) > 0 \;\Rightarrow\; \mathrm{prem}(P, S_d, y) = \mathrm{prem}(Q, S_d, y) = 0$.

Regular GCDs (5/6)

- Let $1 \leq d \leq q$ such that $S_j \in \mathrm{sat}(T)$ for all $0 \leq j < d$.

Proposition

Assume

- $\mathrm{lc}(S_d, y)$ *is regular modulo* $\mathrm{sat}(T)$*,*
- $\mathrm{sat}(T)$ *is radical.*

Then, S_d *is a regular GCD of* P, Q *modulo* $\mathrm{sat}(T)$*.*

Proposition

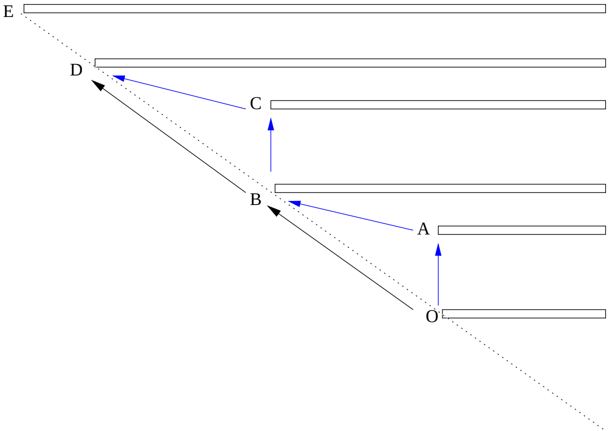
Assume

- $\mathrm{lc}(S_d, y)$ *is regular modulo* $\mathrm{sat}(T)$*,*
- *for all* $d < k \leq q$*,* $\mathrm{coeff}(S_k, y^k)$ *is either* 0 *or regular modulo* $\mathrm{sat}(T)$*.*

Then, S_d *is a regular GCD of* P, Q *modulo* $\mathrm{sat}(T)$*.*

Regular GCDs (6/6)

- Assume that the subresultants S_j for $1 \leq j < q$ are computed.
- Then one can compute a regular GCD of P, Q modulo $\mathrm{sat}(T)$ by performing a bottom-up search.



Implementation and Complexity Estimates (1/2)

We assume that the the base field **k** supports FFT.

- Recall $P, Q \in \mathbf{k}[x_1, \ldots, x_n][y]$. Let $x_{n+1} := y$.

Implementation and Complexity Estimates (1/2)

We assume that the the base field **k** supports FFT.

- Recall $P, Q \in \mathbf{k}[x_1, \ldots, x_n][y]$. Let $x_{n+1} := y$.
- We regard P, Q as univariate polynomials in x_{n+1}.

Implementation and Complexity Estimates (1/2)

We assume that the the base field $\mathbf{k}$ supports FFT.

- Recall $P, Q \in \mathbf{k}[x_1, \ldots, x_n][y]$. Let $x_{n+1} := y$.
- We regard P, Q as univariate polynomials in x_{n+1}.
- We evaluate their coefficients at sufficiently many points s.t. Chain(P, Q) can be computed by evaluation / interpolation (thus via Chinese Remaindering Theorem).

Implementation and Complexity Estimates (1/2)

We assume that the the base field $\mathbf{k}$ supports FFT.

- Recall $P, Q \in \mathbf{k}[x_1, \ldots, x_n][y]$. Let $x_{n+1} := y$.
- We regard P, Q as univariate polynomials in x_{n+1}.
- We evaluate their coefficients at sufficiently many points s.t. Chain(P, Q) can be computed by evaluation / interpolation (thus via Chinese Remaindering Theorem).
- To do so, we need bounds. We consider the Sylvester Matrix. Define $d_i := \max(\deg(P, x_i), \deg(Q, x_i))$. We have

$$\deg(R, x_i) \leq b_i := 2 d_i d_{n+1}.$$

Implementation and Complexity Estimates (1/2)

We assume that the the base field $\mathbf{k}$ supports FFT.

- Recall $P, Q \in \mathbf{k}[x_1, \ldots, x_n][y]$. Let $x_{n+1} := y$.
- We regard P, Q as univariate polynomials in x_{n+1}.
- We evaluate their coefficients at sufficiently many points s.t. Chain(P, Q) can be computed by evaluation / interpolation (thus via Chinese Remaindering Theorem).
- To do so, we need bounds. We consider the Sylvester Matrix. Define $d_i := \max(\deg(P, x_i), \deg(Q, x_i))$. We have
$$\deg(R, x_i) \leq b_i := 2 d_i d_{n+1}.$$
- $B := (b_1 + 1) \cdots (b_n + 1)$ is the number of points at which we need to evaluate P, Q.

Implementation and Complexity Estimates (2/2)

- We choose the B points not cancelling $\mathrm{init}(P)$ and $\mathrm{init}(Q)$.

Implementation and Complexity Estimates (2/2)

- We choose the B points not cancelling $\mathrm{init}(P)$ and $\mathrm{init}(Q)$.
- We evaluate P, Q at these B points via n-dimensional FFT in time $O(d_{n+1} B \log(B))$.

Implementation and Complexity Estimates (2/2)

- We choose the B points not cancelling $\mathrm{init}(P)$ and $\mathrm{init}(Q)$.
- We evaluate P, Q at these B points via n-dimensional FFT in time $O(d_{n+1} B \log(B))$.
- At each of the B points we compute the subresultants of P and Q in time $O(d_{n+1}^2 B)$.

Implementation and Complexity Estimates (2/2)

- We choose the B points not cancelling $\mathrm{init}(P)$ and $\mathrm{init}(Q)$.
- We evaluate P, Q at these B points via n-dimensional FFT in time $O(d_{n+1} B \log(B))$.
- At each of the B points we compute the subresultants of P and Q in time $O(d_{n+1}^2 B)$.
- We interpolate $\mathrm{res}(P, Q, y) = S_0$ in time $O(B \log(B))$ via n-dimensional FFT .

Implementation and Complexity Estimates (2/2)

- We choose the B points not cancelling $\mathrm{init}(P)$ and $\mathrm{init}(Q)$.
- We evaluate P, Q at these B points via n-dimensional FFT in time $O(d_{n+1} B \log(B))$.
- At each of the B points we compute the subresultants of P and Q in time $O(d_{n+1}^2 B)$.
- We interpolate $\mathrm{res}(P, Q, y) = S_0$ in time $O(B \log(B))$ via n-dimensional FFT .
- If $\mathrm{sat}(T)$ is radical, a regular GCD is interpolated within $O(d_{n+1} B \log(B))$; otherwise $O(d_{n+1}^2 B \log(B))$.

Implementation and Complexity Estimates (2/2)

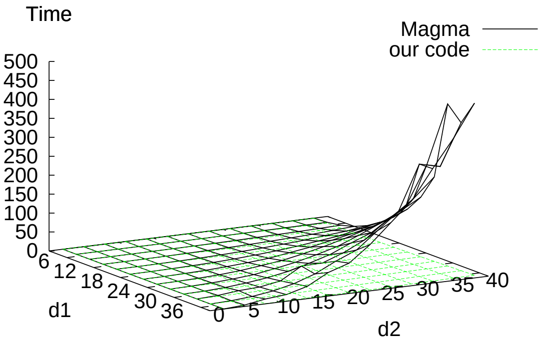
- We choose the B points not cancelling $\mathrm{init}(P)$ and $\mathrm{init}(Q)$.
- We evaluate P, Q at these B points via n-dimensional FFT in time $O(d_{n+1} B \log(B))$.
- At each of the B points we compute the subresultants of P and Q in time $O(d_{n+1}^2 B)$.
- We interpolate $\mathrm{res}(P, Q, y) = S_0$ in time $O(B \log(B))$ via n-dimensional FFT .
- If $\mathrm{sat}(T)$ is radical, a regular GCD is interpolated within $O(d_{n+1} B \log(B))$; otherwise $O(d_{n+1}^2 B \log(B))$.
- Regularity tests (and normal forms) also fit these bounds.

Implementation and Complexity Estimates (2/2)

- We choose the B points not cancelling $\mathrm{init}(P)$ and $\mathrm{init}(Q)$.
- We evaluate P, Q at these B points via n-dimensional FFT in time $O(d_{n+1} B \log(B))$.
- At each of the B points we compute the subresultants of P and Q in time $O(d_{n+1}^2 B)$.
- We interpolate $\mathrm{res}(P, Q, y) = S_0$ in time $O(B \log(B))$ via n-dimensional FFT .
- If $\mathrm{sat}(T)$ is radical, a regular GCD is interpolated within $O(d_{n+1} B \log(B))$; otherwise $O(d_{n+1}^2 B \log(B))$.
- Regularity tests (and normal forms) also fit these bounds.
- If a regular GCD is expected to have degree 1 in y all computations fit in $O\tilde{~}(d_{n+1} B)$.

Generic Bivariate Systems

- "our code" means `BivariateModularTriangularize` in MAPLE 13.
- Random generic input systems, thus equiprojectable.
- For the largest examples (having about 5700 solutions), the ratio is about 460/7 in our favor.



Non-generic Bivariate Systems

- Examples designed to enforce many "splittings" (many equiprojectable components).
- For the largest examples, the ratio is 5260/80, in our favor.

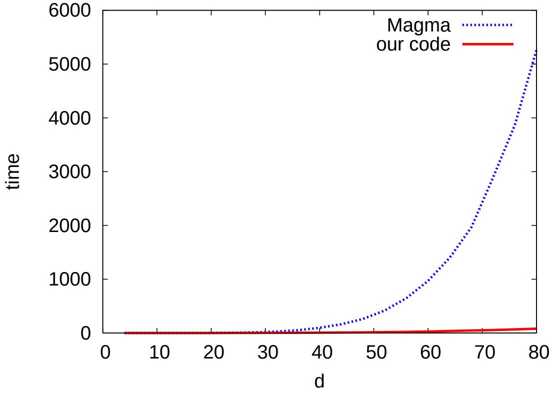


Figure: Non-generic bivariate systems: MAGMA vs. us.

Generic Trivariate Systems

- MAPLE means the experimental and fast version of `Triangularize` to be integrated in MAPLE 14.

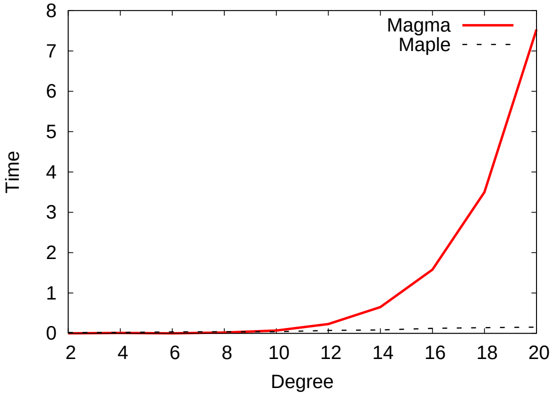


Figure: *Generic dense 3-variable.*

Part IV: Regularity test

- Testing regularity
- Experimental results

Regularity Test

For T 0-dim, auto-reduced and with $h_T = 1$ this procedure returns $T^1, \ldots, T^e$ such that Q is either zero or invertible modulo T^i.

RegularizeDim0(Q, T) ==

(0) **if** $Q \in \mathbf{k}$ **then return** $[T]$

(1) $\quad$ $Results := []$; $v := \mathrm{mvar}(Q)$

(2) $\quad$ $R := \mathrm{res}(Q, T_v, v)$

(3) $\quad$ **for** $D \in$ RegularizeDim0$(R, T_{<v})$ **do**

(4) $\quad\quad$ $s :=$ NormalForm(R, D)

(5) $\quad\quad$ **if** $s \neq 0$ **then**

(7) $\quad\quad\quad$ $Results := \{\{D \cup \{T_v\} \cup T_{>v}\}\} \cup Results$

(8) $\quad\quad$ **else for** $(g, E) \in$ RegularGcd(Q, T_v, D) **do**

(9) $\quad\quad\quad$ $g :=$ NormalForm(g, E)

(11) $\quad\quad\quad$ $Results := \{\{E \cup \{g\} \cup T_{>v}\}\} \cup Results$

(12) $\quad\quad\quad$ $c :=$ NormalForm$(\mathrm{quo}(T_v, g), E)$

(13) $\quad\quad\quad$ **if** $\deg(c, v) > 0$ **then**

(14) $\quad\quad\quad\quad$ $Results :=$ RegularizeDim0$(q, E \cup c \cup T_{>}v) \cup Results$

(15) **return** $Results$

Regularity Test (= Saturation)

d_1	d_2	d_3	Regularize	Fast Regularize	Magma
2	2	3	0.032	0.004	0.010
3	4	6	0.160	0.016	0.020
4	6	9	0.404	0.024	0.060
5	8	12	>100	0.129	0.330
6	10	15	>100	0.272	1.300
7	12	18	>100	0.704	5.100
8	14	21	>100	1.276	14.530
9	16	24	>100	5.836	40.770
10	18	27	>100	9.332	107.280
11	20	30	>100	15.904	229.950
12	22	33	>100	33.146	493.490

Table: Generic dense 3-variable.

- In the non-generic case, both gaps are even larger.
- "Fast Regularize" means `RegularizeDim0` in MAPLE 13.

Conclusions

- Modular methods help reducing expression swell and algebraic complexity.
- Modular methods create opportunities for fast arithmetic and parallelism.
- Fast arithmetic reduces algebraic complexity further.
- Performance improvements can come also from other factors: avoiding re-computations, controlling memory traffic
- Controlling expression swell may require to understand the structure of the computed objects.

Xie Xie! Thank You!

Positive Dimensional Regular Chains

- Let $\mathbf{u} = u_1, \ldots, u_d$ be parameters, $\mathbf{y} = y_1, \ldots, y_m$ be unknowns, Π_U be the projection from $\mathbf{K}^{m+d}$ to $\mathbf{K}^d$.

Positive Dimensional Regular Chains

- Let $\mathbf{u} = u_1, \ldots, u_d$ be parameters, $\mathbf{y} = y_1, \ldots, y_m$ be unknowns, Π_U be the projection from $\mathbf{K}^{m+d}$ to $\mathbf{K}^d$.
- A regular chain $T \subset \mathbf{k}[\mathbf{u}, \mathbf{y}]$ *specializes well* at $u \in \mathbf{K}^d$ if $T(u)$ is a regular chain in $\mathbf{K}[\mathbf{y}]$ and $\mathrm{rank}(T(u)) = \mathrm{rank}(T_{>U_d})$.

Positive Dimensional Regular Chains

- Let $\mathbf{u} = u_1, \ldots, u_d$ be parameters, $\mathbf{y} = y_1, \ldots, y_m$ be unknowns, Π_U be the projection from $\mathbf{K}^{m+d}$ to $\mathbf{K}^d$.
- A regular chain $T \subset \mathbf{k}[\mathbf{u}, \mathbf{y}]$ *specializes well* at $u \in \mathbf{K}^d$ if $T(u)$ is a regular chain in $\mathbf{K}[\mathbf{y}]$ and $\mathrm{rank}(T(u)) = \mathrm{rank}(T_{>U_d})$.
- Let $T \subset \mathbf{k}[\mathbf{u}, \mathbf{y}]$ be a reg. chain and $u \in \Pi_U(\mathbf{W}(T \cap \mathbf{k}[U]))$. T specializes well at $u \iff \mathrm{res}(h_{T_{>U_d}}, T_{>U_d}) \neq 0$ at $\mathbf{u} = u$

Positive Dimensional Regular Chains

- Let $\mathbf{u} = u_1, \ldots, u_d$ be parameters, $\mathbf{y} = y_1, \ldots, y_m$ be unknowns, Π_U be the projection from $\mathbf{K}^{m+d}$ to $\mathbf{K}^d$.
- A regular chain $T \subset \mathbf{k}[\mathbf{u}, \mathbf{y}]$ *specializes well* at $u \in \mathbf{K}^d$ if $T(u)$ is a regular chain in $\mathbf{K}[\mathbf{y}]$ and $\mathrm{rank}(T(u)) = \mathrm{rank}(T_{>U_d})$.
- Let $T \subset \mathbf{k}[\mathbf{u}, \mathbf{y}]$ be a reg. chain and $u \in \Pi_U(\mathbf{W}(T \cap \mathbf{k}[U]))$. T specializes well at $u \iff \mathrm{res}(h_{T_{>U_d}}, T_{>U_d}) \neq 0$ at $\mathbf{u} = u$
- Replacing *regular chain* by squarefree reg. ch. in char. 0 and $h_{T_{>U_d}}$ by $\mathrm{Sep}_{T_{>U_d}}$ one obtains the *border polynomial of T*.

Positive Dimensional Regular Chains

- Let $\mathbf{u} = u_1, \ldots, u_d$ be parameters, $\mathbf{y} = y_1, \ldots, y_m$ be unknowns, Π_U be the projection from $\mathbf{K}^{m+d}$ to $\mathbf{K}^d$.
- A regular chain $T \subset \mathbf{k}[\mathbf{u}, \mathbf{y}]$ *specializes well* at $u \in \mathbf{K}^d$ if $T(u)$ is a regular chain in $\mathbf{K}[\mathbf{y}]$ and $\mathrm{rank}(T(u)) = \mathrm{rank}(T_{>U_d})$.
- Let $T \subset \mathbf{k}[\mathbf{u}, \mathbf{y}]$ be a reg. chain and $u \in \Pi_U(\mathbf{W}(T \cap \mathbf{k}[U]))$. T specializes well at $u \iff \mathrm{res}(h_{T_{>U_d}}, T_{>U_d}) \neq 0$ at $\mathbf{u} = u$
- Replacing *regular chain* by squarefree reg. ch. in char. 0 and $h_{T_{>U_d}}$ by $\mathrm{Sep}_{T_{>U_d}}$ one obtains the *border polynomial of T*.

Related Work

On a projection theorem of quasi-varieties in elimination theory (Wen-Tsün Wu 90). (Xiao-Shan Gao, Shang-Ching Chou 92) (Dongming Wang 00 & 01) (Lu Yang, Xiaorong Hou, Bican Xia 01) (Xiao-Shan Gao, Ding-Kang Wang 03) (Changbo Chen, Oleg Golubitsky, François Lemaire, Marc Moreno Maza, Wei Pan 07)

Equiprojectable Decomposition (1/2)

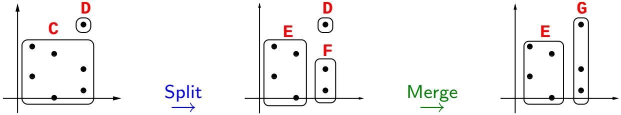
$$C \left| \begin{array}{l} C_2 = y^2 + 6yx^2 + 2y + x \\ C_1 = x^3 + 6x^2 + 5x + 2 \end{array} \right. , \quad D \left| \begin{array}{l} D_2 = y + 6 \\ D_1 = x + 6 \end{array} \right.$$

$\downarrow$ Split C : GCD $\downarrow$

$$E \left| \begin{array}{l} {C_2}' = y^2 + x \\ {C_1}' = x^2 + 5 \end{array} \right. , \quad F \left| \begin{array}{l} C_2'' = y^2 + y + 1 \\ C_1'' = x + 6 \end{array} \right. , \quad D \left| \begin{array}{l} D_2 = y + 6 \\ D_1 = x + 6 \end{array} \right.$$

$\downarrow$ Merge F and D : CRT $\downarrow$

$$E \left| \begin{array}{l} C_2' = y^2 + x \\ C_1' = x^2 + 5 \end{array} \right. , \quad G \left| \begin{array}{l} G_2 = y^3 + 6 \\ G_1 = x + 6 \end{array} \right.$$



Equiprojectable Decomposition (2/2)

$$\begin{array}{ccc}
F \subset \mathbb{Q}[x_1, \ldots, x_n] \\ p \text{ prime}, \ p > A(F) & \longrightarrow & Eq.D._{\mathbb{Q}}(F) \\
\Big\downarrow \text{Triang.}(F \mod p) & & \Big\uparrow \textit{Lifting} \\
\{T^1, \ldots, T^s\} \\ \subset \mathbb{F}_p[x_1, \ldots, x_n] & \xrightarrow{\textbf{Split} + \textbf{Merge}} & Eq.D._{\mathbb{F}_p}(F)
\end{array}$$

- $A(F) := 2n^2 d^{2n+1}(3h + 7\log(n+1) + 5n\log d + 10)$ where h and d upper bound coeff. sizes and total degrees for $f \in F$. Assumes F square and generates a 0-dimensional radical ideal.

Equiprojectable Decomposition (2/2)

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
\begin{array}{c} F \subset \mathbb{Q}[x_1, \ldots, x_n] \\ p \text{ prime}, \ p > A(F) \end{array} \arrow[r] \arrow[d, swap, "\text{Triang.}(F \mod p)"] & Eq.D._{\mathbb{Q}}(F) \\
\begin{array}{c} \{T^1, \ldots, T^s\} \\ \subset \mathbb{F}_p[x_1, \ldots, x_n] \end{array} \arrow[r, "\textbf{Split} + \textbf{Merge}"] & Eq.D._{\mathbb{F}_p}(F) \arrow[u, swap, "\textit{Lifting}"]
\end{tikzcd}
$$

- $A(F) := 2n^2 d^{2n+1}(3h + 7\log(n+1) + 5n \log d + 10)$ where h and d upper bound coeff. sizes and total degrees for $f \in F$. Assumes F square and generates a 0-dimensional radical ideal.
- If $p \not| A(F)$, the equiprojectable decomposition specializes well mod p.

Equiprojectable Decomposition (2/2)

$$
\begin{tikzcd}
\begin{array}{c} F \subset \mathbb{Q}[x_1, \ldots, x_n] \\ p \text{ prime}, \ p > A(F) \end{array} \arrow[r] \arrow[d, "\text{Triang.}(F \mod p)"'] & Eq.D._{\mathbb{Q}}(F) \\
\begin{array}{c} \{T^1, \ldots, T^s\} \\ \subset \mathbb{F}_p[x_1, \ldots, x_n] \end{array} \arrow[r, "\textbf{Split + Merge}"] & Eq.D._{\mathbb{F}_p}(F) \arrow[u, "\textit{Lifting}"']
\end{tikzcd}
$$

- $A(F) := 2n^2 d^{2n+1}(3h + 7\log(n+1) + 5n\log d + 10)$ where h and d upper bound coeff. sizes and total degrees for $f \in F$. Assumes F square and generates a 0-dimensional radical ideal.
- If $p \not| A(F)$, the equiprojectable decomposition specializes well mod p.
- In practice we choose p much smaller with a probability of success, i.e. $> 99\%$ with $p \approx ln(A(F))$ (Xavier Dahan, M. M. M., Éric Schost, Wenyuan Wu, Yuzhen Xie 05).

Incremental Solving

- Let $F \subset \mathbf{k}[\mathbf{x}]$, $f \in \mathbf{k}[\mathbf{x}]$, $T, T^m \ldots, T^e \subset \mathbf{k}[\mathbf{x}]$ reg. chains. Assume we have *solved* F as $V(F) = W(T^i) \cup \cdots \cup W(T^e)$.

Incremental Solving

- Let $F \subset \mathbf{k}[\mathbf{x}]$, $f \in \mathbf{k}[\mathbf{x}]$, $T, T^m \ldots, T^e \subset \mathbf{k}[\mathbf{x}]$ reg. chains. Assume we have *solved* F as $V(F) = W(T^i) \cup \cdots \cup W(T^e)$.
- Assume that we have an operation $(f, T) \longmapsto \mathbf{Intersect}(f, T) = (C_1, \ldots, C_d)$ such that

$$V(f) \cap W(T) \;\subseteq\; \cup_i W(C_i) \;\subseteq\; V(f) \cap \overline{W(T)}.$$

Then solving $F \cup f$ reduces to $\mathbf{Intersect}(f, T^i)$ for all i.

Incremental Solving

- Let $F \subset \mathbf{k}[\mathbf{x}]$, $f \in \mathbf{k}[\mathbf{x}]$, $T, T^m \ldots, T^e \subset \mathbf{k}[\mathbf{x}]$ reg. chains. Assume we have *solved* F as $V(F) = W(T^i) \cup \cdots \cup W(T^e)$.
- Assume that we have an operation $(f, T) \longmapsto \mathbf{Intersect}(f, T) = (C_1, \ldots, C_d)$ such that

$$V(f) \cap W(T) \ \subseteq \ \cup_i W(C_i) \ \subseteq \ V(f) \cap \overline{W(T)}.$$

Then solving $F \cup f$ reduces to $\mathbf{Intersect}(f, T^i)$ for all i.

$\Longrightarrow$ the core routine operates on well behaved objects.

Incremental Solving

- Let $F \subset \mathbf{k}[\mathbf{x}]$, $f \in \mathbf{k}[\mathbf{x}]$, $T, T^m \ldots, T^e \subset \mathbf{k}[\mathbf{x}]$ reg. chains. Assume we have *solved* F as $V(F) = W(T^i) \cup \cdots \cup W(T^e)$.
- Assume that we have an operation $(f, T) \longmapsto \textbf{Intersect}(f, T) = (C_1, \ldots, C_d)$ such that

$$V(f) \cap W(T) \;\subseteq\; \cup_i W(C_i) \;\subseteq\; V(f) \cap \overline{W(T)}.$$

Then solving $F \cup f$ reduces to $\textbf{Intersect}(f, T^i)$ for all i.

$\Longrightarrow$ the core routine operates on well behaved objects.

$\Longrightarrow$ the decomposition can be reduced to regular GCD computation, allowing modular methods and fast arithmetic.

Incremental Solving

- Let $F \subset \mathbf{k}[\mathbf{x}]$, $f \in \mathbf{k}[\mathbf{x}]$, $T, T^m \ldots, T^e \subset \mathbf{k}[\mathbf{x}]$ reg. chains. Assume we have *solved* F as $V(F) = W(T^i) \cup \cdots \cup W(T^e)$.
- Assume that we have an operation $(f, T) \longmapsto \mathbf{Intersect}(f, T) = (C_1, \ldots, C_d)$ such that

$$V(f) \cap W(T) \;\subseteq\; \cup_i W(C_i) \;\subseteq\; V(f) \cap \overline{W(T)}.$$

Then solving $F \cup f$ reduces to $\mathbf{Intersect}(f, T^i)$ for all i.

$\Longrightarrow$ the core routine operates on well behaved objects.

$\Longrightarrow$ the decomposition can be reduced to regular GCD computation, allowing modular methods and fast arithmetic.

Related Work

(D. Lazard 91) proposes the principle. (M. M. M. 00) introduces regular GCDs and gives a complete incremental algorithm which, in addition, generates components by decreasing order of dimension.

The notion of a Regular GCD

- Let $P, Q, G \in \mathbf{k}[x_1 < \cdots < x_n][y]$ and $T \subset \mathbf{k}[x_1 < \cdots < x_n]$ reg. chain. G is a *regular GCD* of P, Q modulo $\mathrm{sat}(T)$ if
  - (*i*) $\mathrm{lc}(G, y)$ is a regular modulo $\mathrm{sat}(T)$,
  - (*ii*) $G \in \langle P, Q \rangle$ modulo $\mathrm{sat}(T)$,
  - (*iii*) $\deg_y(G) > 0 \Rightarrow \mathrm{prem}_y(P, G), \mathrm{prem}_y(Q, G) \in \mathrm{sat}(T)$.

The notion of a Regular GCD

- Let $P, Q, G \in \mathbf{k}[x_1 < \cdots < x_n][y]$ and $T \subset \mathbf{k}[x_1 < \cdots < x_n]$ reg. chain. G is a *regular GCD* of P, Q modulo $\mathrm{sat}(T)$ if
  - (*i*) $\mathrm{lc}(G, y)$ is a regular modulo $\mathrm{sat}(T)$,
  - (*ii*) $G \in \langle P, Q \rangle$ modulo $\mathrm{sat}(T)$,
  - (*iii*) $\deg_y(G) > 0 \;\Rightarrow\; \mathrm{prem}_y(P, G), \mathrm{prem}_y(Q, G) \in \mathrm{sat}(T)$.
- If both $T \cup P$ and $T \cup Q$ are regular chains and if G is a GCD of P, Q modulo $\mathrm{sat}(T)$ with $\deg_y(G) > 0$ then we have

$$\begin{array}{rl} W(T \cup P) \;\cap\; V(Q) \subseteq & W(T \cup G) \cup \\ & W(T \cup P) \;\cap\; V(Q, h_G) \;\subseteq\; \overline{W(T \cup P)} \;\cap\; V(Q). \end{array}$$

The notion of a Regular GCD

- Let $P, Q, G \in \mathbf{k}[x_1 < \cdots < x_n][y]$ and $T \subset \mathbf{k}[x_1 < \cdots < x_n]$ reg. chain. G is a *regular GCD* of P, Q modulo $\mathrm{sat}(T)$ if
  - (i) $\mathrm{lc}(G, y)$ is a regular modulo $\mathrm{sat}(T)$,
  - (ii) $G \in \langle P, Q \rangle$ modulo $\mathrm{sat}(T)$,
  - (iii) $\deg_y(G) > 0 \;\Rightarrow\; \mathrm{prem}_y(P, G), \mathrm{prem}_y(Q, G) \in \mathrm{sat}(T)$.
- If both $T \cup P$ and $T \cup Q$ are regular chains and if G is a GCD of P, Q modulo $\mathrm{sat}(T)$ with $\deg_y(G) > 0$ then we have

$$
\begin{aligned}
W(T \cup P) \, \cap \, V(Q) \subseteq \;& W(T \cup G) \cup \\
& W(T \cup P) \, \cap \, V(Q, h_G) \subseteq \overline{W(T \cup P)} \, \cap \, V(Q).
\end{aligned}
$$

- One can compute $T^1, \ldots, T^e$ and $G_1, \ldots, G_e$ such that G_i is a reg. GCD of P, Q mod $\mathrm{sat}(T_i)$ and $\sqrt{\mathrm{sat}(T)} = \cap_{i=0}^{e} \sqrt{\mathrm{sat}(T^i)}$.

Regularity test

- Regularity test is a fundamental operation:

$$\mathrm{Regularize}(p, \mathcal{I}) \quad \longmapsto \quad (\mathcal{I}_1, \ldots, \mathcal{I}_e)$$

  such that:

$$\sqrt{\mathcal{I}} = \cap_{i=0}^{e} \sqrt{\mathcal{I}_i} \ \text{ and } \ p \in \mathcal{I}_i \text{ or } p \text{ regular modulo } \mathcal{I}_i$$

- Regularity test reduces to regular GCD computation.

Related work

- This notion of a regular GCD was proposed in (M. M. 2000)
- In previous work (Kalkbrener 1993) and (Rioboo & M. M. 1995), other regular GCDs modulo regular chains were introduced, but with limitations.
- In other work (Wang 2000), (Yang etc. 1995) and (Jean Della Dora, Claire Dicrescenzo, Dominique Duval 85), related techniques are used to construct triangular decompositions.
- Regular GCDs modulo regular chains generalize GCDs over towers of field extentions for which specialized algorithms are available, (van Hoeij and Monagan 2002 & 2004).
- Asymptotically fast algorithms (when $\mathrm{sat}(T)$ is zero-dimensional and radical) appear in (Xavier Dahan, M. M. , Éric Schost, Yuzhen Xie, 2006)
- The next results appear in (Xin Li, M. M. , Wei Pan, 2009).

Regular GCDs: Bottom-up or Top-down?

- Let $P, Q \in \mathbf{k}[x_1 < \cdots < x_n][y]$ and $T \subset \mathbf{k}[x_1 < \cdots < x_n]$ reg. chain. How to compute a *regular GCD* of P, Q mod $\mathrm{sat}(T)$?
- (M. Kalkbrener 91) uses Pseudo-Remainder Sequences. Inefficient!

Regular GCDs: Bottom-up or Top-down?

- Let $P, Q \in \mathbf{k}[x_1 < \cdots < x_n][y]$ and $T \subset \mathbf{k}[x_1 < \cdots < x_n]$ reg. chain. How to compute a *regular GCD* of P, Q mod $\mathrm{sat}(T)$?
- (M. Kalkbrener 91) uses Pseudo-Remainder Sequences. Inefficient!
- (Jean Della Dora, Claire Dicrescenzo, Dominique Duval 85) assume $\mathrm{sat}(T)$ radical and compute (naively) the subresultant chain of P, Q in $\mathbf{k}[x_1 < \cdots < x_n][y]$. Limited and practically inefficient!

Regular GCDs: Bottom-up or Top-down?

- Let $P, Q \in \mathbf{k}[x_1 < \cdots < x_n][y]$ and $T \subset \mathbf{k}[x_1 < \cdots < x_n]$ reg. chain. How to compute a *regular GCD* of P, Q mod $\mathrm{sat}(T)$?
- (M. Kalkbrener 91) uses Pseudo-Remainder Sequences. Inefficient!
- (Jean Della Dora, Claire Dicrescenzo, Dominique Duval 85) assume $\mathrm{sat}(T)$ radical and compute (naively) the subresultant chain of P, Q in $\mathbf{k}[x_1 < \cdots < x_n][y]$. Limited and practically inefficient!
- (M. M. M. and R. Rioboo 95) assume $\mathrm{sat}(T)$ radical + 0-dimensional and use the subresultant chain of P, Q directly in $\mathbf{k}[x_1 < \cdots < x_n][y]$ mod $\mathrm{sat}(T)$. Better but removing the assumptions removes the efficiency.

Subresultants (3/3

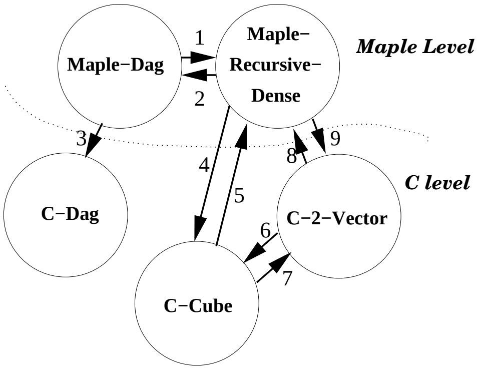
- Let $P, Q \in \mathbb{B}[y]$ with $p = \deg(P) \geq \deg(Q) = q > 0$.
- For $0 \leq d < q$ let $S_d = S_d(P, Q)$ be the d-th *subresultant* of P and Q. Let $s_d = \mathrm{coeff}(S_d, x^d)$. If $s_d = 0$ we say S_d is defective, otherwise we say S_d is non-defective.
- Let $d = q-1, \ldots, 1$. Assume S_d, S_{d-1} nonzero, with resp. degrees d and e. Assume s_d regular in $\mathbb{B}$. Then we have
$$\mathrm{lc}(S_{d-1})^{d-e-1} S_{d-1} = {s_d}^{d-e-1} S_e.$$
- Moreover, there exists $C_d \in \mathbb{B}[X]$ such that we have:
$$(-1)^{d-1}\mathrm{lc}(S_{d-1}) s_e S_d + C_d S_{d-1} = {s_d}^2 S_{e-1}.$$

  In addition $S_{d-2} = S_{d-3} = \cdots = S_{e+1} = 0$ also holds.
- (Yap 1993) (Ducos 1997) (El Kahoui, 2003)

The RegularChains library in MAPLE

- 80,000 lines of MAPLE code, 36,000 lines of C code, 121 Commands, 6 modules ChainTools, MatrixTools, ConstructibleSetTools. ParametricSystemTools, SemiAlgebraicSetTools , FastArithmeticTools .
- Main new commnands in MAPLE 13: IsPrimitive, ComplexRootClassification, RealRootClassification, RealRootIsolate RealRootCounting, BorderPolynomial, + those of FastArithmeticTools (see demo).
- Current contributors: Changbo Chen, Francçois Lemaire, Liyun Li, Xin Li, M.M.M., Wei Pan, Bican Xia, Rong Xiao, Yuzhen Xie.

The MODPN library



- *C-Dag* for straight-line program.
- *C-Cube* for FFT-based computations.
- *C-2-Vector* for compact dense representation.
- *Maple-Dag* for calling `RegularChains` library.
- *Maple-Recursive-Dense* for calling RECDEN library.

Generic Bivariate Systems

- "our code" means `BivariateModularTriangularize` in MAPLE 13.
- Random generic input systems, thus equiprojectable.
- For the largest examples (having about 5700 solutions), the ratio is about 460/7 in our favor.