A Maple implementation of a modular algorithm for computing the common zeros of a polynomial and a regular chain

Maple Conference 2021

Matt Calder, Juan Pablo Gonzalez Trochez, Marc Moreno Maza and Erik Postma

University of Western Ontario, Maplesoft

October 12, 2021

jgonza55@uwo.ca

Agenda

Outline

Solving polynomial systems incrementally

- Most algorithms for solving polynomial systems symbolically proceed
    - *incrementally*, that is, solving one equation after another, against the solutions of the previously solved equations, or
    - by *projection and lifting*, that is, by successively eliminating one variable after another, and then proceeding by back-substitution as in linear system solving.
- The algorithm Triangularize of the Maple's `RegularChains` library belongs to the category of incremental solving.
- Without entering technical details, we illustrate this algorithm in the following slides.

Incremental solving: a toy example

$$F = \begin{cases} y + w \\ 5w^2 + y \\ xz + z^3 + z \\ x^5 + x^3 + z \end{cases}$$

$$\emptyset$$

$$\xrightarrow{F[1]}$$

$$\{y + w\}$$

$F[2]$

$$\left\{ \begin{array}{c} 5y + 1 \\ 5w - 1 \end{array} \right\}, \quad \left\{ \begin{array}{c} y \\ w \end{array} \right\}$$

$F[3]$

$$\left\{ \begin{array}{r} x + z^2 + 1 \\ 5y + 1 \\ 5w - 1 \end{array} \right\}, \quad \left\{ \begin{array}{r} 5y + 1 \\ z \\ 5w - 1 \end{array} \right\}, \quad \left\{ \begin{array}{r} x + z^2 + 1 \\ y \\ w \end{array} \right\}, \quad \left\{ \begin{array}{r} y \\ z \\ w \end{array} \right\}$$

$F[4]$

$$\left\{ \begin{array}{r} x + z^2 + 1 \\ 5y + 1 \\ z^8 + \cdots \\ 5w - 1 \end{array} \right., \quad \left\{ \begin{array}{r} x - z \\ 5y + 1 \\ z^2 + z + 1 \\ 5w - 1 \end{array} \right., \quad \left\{ \begin{array}{r} x \\ 5y + 1 \\ z \\ 5w - 1 \end{array} \right., \quad \left\{ \begin{array}{r} x^2 + 1 \\ 5y + 1 \\ z \\ 5w - 1 \end{array} \right., \quad \left\{ \begin{array}{r} x + z^2 + 1 \\ y \\ z^8 + \cdots \\ w \end{array} \right., \quad \left\{ \begin{array}{r} x - z \\ y \\ z^2 + z + 1 \\ w \end{array} \right.$$

$$\left\{ \begin{array}{r} x^2 + 1 \\ y \end{array} \right. \quad \left\{ \begin{array}{r} x \\ y \end{array} \right.$$

Incremental solving: a real-life example

- As illustrated below, the algorithm Triangularize proceeds by repeated calls to a procedure called Intersect.
- This procedure computes the common zeros of a polynomial and a solution set (termed *regular chain* in polynomial system theory).

```
> R := PolynomialRing([x,y,z,t,u]);
```

$$R := polynomial\_ring$$

```
> F := [2*x + 2*y + 2*z + 2*t + u - 1, 2*x^2 + 2*y^2 + 2*z^2 + 2*t^2 + u^2 - u, 2*x*y + 2*y*z + 2*z*t + 2*t*u - t, 2*x*z +
  2*y*t + t^2 + 2*z*u - z, 2*x*t + 2*z*t + 2*y*u - y];rc := Empty(R); lrc := [rc];
```

$$F := [2\,x+2\,y+2\,z+2\,t+u-1, 2\,x^2+2\,y^2+2\,z^2+2\,t^2+u^2-u, 2\,x\,y+2\,y\,z+2\,z\,t+2\,t\,u-t, 2\,x\,z+2\,y\,t+2\,z\,u+t^2-z, 2\,x\,t+2\,y\,u+2\,z\,t-y]$$

$$rc := regular\_chain$$

$$lrc := [rc]$$

```
> ## solving F[1] against lrc
> a:= time(): lrc := [ seq ( op(Intersect(F[1], ts, R)), ts=lrc ) ]; map(Dimension, lrc, R); time() - a;
```

$$lrc := [regular\_chain]$$

$$[4]$$

$$0.003$$

```
> ## solving F[2] against lrc
> a := time() ; lrc := [ seq ( op(Intersect(F[2], ts, R)), ts=lrc ) ]; map(Dimension, lrc, R);time() - a;
```

$$a := 19.378$$

$$lrc := [regular\_chain]$$

$$[3]$$

$$0.006$$

```
> ## solving F[3] against lrc
> a := time(): lrc := [ seq ( op(Intersect(F[3], ts, R)), ts=lrc ) ]; map(Dimension, lrc, R);time() - a;
```

$$lrc := [regular\_chain, regular\_chain]$$

$$[2, 1]$$

```
> ## solving F[4] against lrc
> a:= time(): lrc := [ seq ( op(Intersect(F[4], ts, R)), ts=lrc ) ]; map(Dimension, lrc, R);time() - a;
```

$lrc := [regular\_chain, regular\_chain, regular\_chain, regular\_chain, regular\_chain, regular\_chain, regular\_chain, regular\_chain, regular\_chain]$

$$[1, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$0.270$$

```
> ## solving F[5] against the first regular chain in lrc
> a := time() : Intersect(F[5], lrc[1], R); time() -a ;
```

$$[regular\_chain, regular\_chain]$$

$$1.639$$

```
> ## solving F[5] against the other chains in lrc
> a:= time() ; [ seq ( op(Intersect(F[5], ts, R)), ts=lrc[2..-1] )]; time() - a;
```

$$a := 21.316$$

$$[regular\_chain, regular\_chain]$$

$$0.145$$

- The above example shows that most of the time is spent in computing the common zeroes of a polynomial and a regular chain of dimension 1 (that is, a space curve).
- This motivates the work presented here, where we propose a new algorithm for computing such intersections.
- Our new algorithm is based on *curve fitting techniques*.

Outline

Regular chains

Let **k** be a perfect field, and $\mathbf{k}[X]$ have ordered vars. $X = X_1 < \cdots < X_n$

A triangular set $T \subset \mathbf{k}[X]$ is a regular chain if either T is empty, or T_v^- is a regular chain and h is regular modulo $\mathrm{sat}(T_v^-)$

$$T = \left\{ \begin{array}{l} T_v = h\, v^d + \mathrm{tail}(T_v) \\ \\ T_v^- = \left\{ \quad \right\} \end{array} \right\} \qquad T = \left\{ \begin{array}{r} (2y + ba)x - by + a^2 \\ 2y^2 - by - a^2 \\ a + b \end{array} \right\}$$

$$\subset \mathbb{Q}[b < a < y < x]$$

Saturated ideal of a regular chain:

- $\mathrm{sat}(T) = (\mathrm{sat}(T_v^-) + T_v) : h^\infty$
- $\mathrm{sat}(\emptyset) = \langle 0 \rangle$

Quasi-component of a regular chain:

- $W(T) := V(T) \setminus V(h_T)$, $h_T := \prod_{p \in T} h_p$
- $\overline{W(T)} = V(\mathrm{sat}(T))$

The algorithms Intersect and Triangularize

Intersect

Let $p \in \mathbf{k}[X]$ and let $T \subseteq \mathbf{k}[X]$ be a regular chain. The function call Intersect(p, T) computes regular chains $T_1, \ldots, T_e \subseteq \mathbf{k}[X]$ such that:

$$V(p) \cap W(T) \subseteq W(T_1) \cup \cdots \cup W(T_e) \subseteq V(p) \cap \overline{W(T)}. \quad (1)$$

Triangularize

Given a finite set $F = \{f_0, f_1, f_2, \ldots\} \ldots, \subseteq \mathbf{k}[X]$, Triangularize$(F)$ compute regular chains $T_1, \ldots, T_e \subseteq \mathbf{k}[X]$ encoding the solutions of $V(F)$:

$$V(F) = W(T_1) \cup \cdots \cup W(T_e). \quad (2)$$

This is achieved by successively applying Intersect to $f_0, f_1, f_2, \ldots$ on the previously obtained regular chains.

Outline

Outline

The goal

- Let $\mathbf{k}$ be a field of characteristic zero or a prime field of *sufficiently large* characteristic.

The goal

- Let $\mathbf{k}$ be a field of characteristic zero or a prime field of *sufficiently large* characteristic.
- Let $f, t, b \in \mathbf{k}[x, y, z]$ be non-constant polynomials in the ordered variables $x > y > z$.

The goal

- Let $\mathbf{k}$ be a field of characteristic zero or a prime field of *sufficiently large* characteristic.
- Let $f, t, b \in \mathbf{k}[x, y, z]$ be non-constant polynomials in the ordered variables $x > y > z$.
- Assume that $T := \{t, b\}$ is a regular chain with

$$\mathrm{mvar}(t) = x \text{ and } \mathrm{mvar}(b) = y.$$

The goal

- Let $\mathbf{k}$ be a field of characteristic zero or a prime field of *sufficiently large* characteristic.
- Let $f, t, b \in \mathbf{k}[x, y, z]$ be non-constant polynomials in the ordered variables $x > y > z$.
- Assume that $T := \{t, b\}$ is a regular chain with

$$\mathrm{mvar}(t) = x \text{ and } \mathrm{mvar}(b) = y.$$

- Assume that

$$\mathrm{mvar}(f) = x.$$

The goal

- Let $\mathbf{k}$ be a field of characteristic zero or a prime field of *sufficiently large* characteristic.
- Let $f, t, b \in \mathbf{k}[x, y, z]$ be non-constant polynomials in the ordered variables $x > y > z$.
- Assume that $T := \{t, b\}$ is a regular chain with
$$\mathrm{mvar}(t) = x \text{ and } \mathrm{mvar}(b) = y.$$
- Assume that
$$\mathrm{mvar}(f) = x.$$
- **Our goal:** compute the intersection $V(f) \cap W(T)$ in the sense of the function call Intersect(f, T), i.e., we want to compute regular chains $T_1, \ldots, T_e \subseteq \mathbf{k}[x, y, z]$ such that:
$$V(f) \cap W(T) \subseteq W(T_1) \cup \cdots \cup W(T_e) \subseteq V(f) \cap \overline{W(T)}. \quad (3)$$

Outline

Hypothesis 1

- Let $S(t, f, x)$ be the subresultant chain of t and f (resp. f and t) regarded as polynomials in $(\mathbf{k}[y, z])[x]$ if $\mathrm{mdeg}(t) \geq \mathrm{mdeg}(f)$ (resp. $\mathrm{mdeg}(t) < \mathrm{mdeg}(f)$).

Hypothesis 1

- Let $S(t, f, x)$ be the subresultant chain of t and f (resp. f and t) regarded as polynomials in $(\mathbf{k}[y, z])[x]$ if $\mathrm{mdeg}(t) \geq \mathrm{mdeg}(f)$ (resp. $\mathrm{mdeg}(t) < \mathrm{mdeg}(f)$).
- Let $S_0(t, f, x)$ and $S_1(t, f, x)$ be the subresultants of index 0 and 1 from $S(t, f, x)$.

Hypothesis 1

- Let $S(t, f, x)$ be the subresultant chain of t and f (resp. f and t) regarded as polynomials in $(\mathbf{k}[y, z])[x]$ if $\mathrm{mdeg}(t) \geq \mathrm{mdeg}(f)$ (resp. $\mathrm{mdeg}(t) < \mathrm{mdeg}(f)$).
- Let $S_0(t, f, x)$ and $S_1(t, f, x)$ be the subresultants of index 0 and 1 from $S(t, f, x)$.
- We let

$$r := S_0(t, f, x) \quad \text{and} \quad \ell := S_1(t, f, x). \tag{4}$$

Hypothesis 1

- Let $S(t, f, x)$ be the subresultant chain of t and f (resp. f and t) regarded as polynomials in $(\mathbf{k}[y, z])[x]$ if $\mathrm{mdeg}(t) \geq \mathrm{mdeg}(f)$ (resp. $\mathrm{mdeg}(t) < \mathrm{mdeg}(f)$).
- Let $S_0(t, f, x)$ and $S_1(t, f, x)$ be the subresultants of index 0 and 1 from $S(t, f, x)$.
- We let

$$r := S_0(t, f, x) \quad \text{and} \quad \ell := S_1(t, f, x). \tag{4}$$

Hypothesis

$$r \notin \mathbf{k} \quad \text{and} \quad \mathrm{mvar}(r) = y. \tag{5}$$

Hypothesis 2

- Let $S(r, b, y)$ be the subresultant chain of r and b (resp. b and r) regarded as polynomials in $\mathbf{k}[y, z]$ if $\mathrm{mdeg}(r) \geq \mathrm{mdeg}(b)$ (resp. $\mathrm{mdeg}(r) < \mathrm{mdeg}(b)$).

Hypothesis 2

- Let $S(r, b, y)$ be the subresultant chain of r and b (resp. b and r) regarded as polynomials in $\mathbf{k}[y, z]$ if $\mathrm{mdeg}(r) \geq \mathrm{mdeg}(b)$ (resp. $\mathrm{mdeg}(r) < \mathrm{mdeg}(b)$).
- Let $s := S_0(r, b, y)$ and $g := S_1(r, b, y)$ be the subresultants of index 0 and 1 from $S(r, b, y)$.

Hypothesis 2

- Let $S(r, b, y)$ be the subresultant chain of r and b (resp. b and r) regarded as polynomials in $\mathbf{k}[y, z]$ if $\mathrm{mdeg}(r) \geq \mathrm{mdeg}(b)$ (resp. $\mathrm{mdeg}(r) < \mathrm{mdeg}(b)$).
- Let $s := S_0(r, b, y)$ and $g := S_1(r, b, y)$ be the subresultants of index 0 and 1 from $S(r, b, y)$.
- We denote $\overline{s}$ the squarefree part of s.

Hypothesis 2

- Let $S(r, b, y)$ be the subresultant chain of r and b (resp. b and r) regarded as polynomials in $\mathbf{k}[y, z]$ if $\mathrm{mdeg}(r) \geq \mathrm{mdeg}(b)$ (resp. $\mathrm{mdeg}(r) < \mathrm{mdeg}(b)$).
- Let $s := S_0(r, b, y)$ and $g := S_1(r, b, y)$ be the subresultants of index 0 and 1 from $S(r, b, y)$.
- We denote $\overline{s}$ the squarefree part of s.

Hypothesis

The polynomial set $C := \{\overline{s}, g, \ell\}$ is a regular chain.

Hypothesis 2

- Let $S(r, b, y)$ be the subresultant chain of r and b (resp. b and r) regarded as polynomials in $\mathbf{k}[y, z]$ if $\mathrm{mdeg}(r) \geq \mathrm{mdeg}(b)$ (resp. $\mathrm{mdeg}(r) < \mathrm{mdeg}(b)$).
- Let $s := S_0(r, b, y)$ and $g := S_1(r, b, y)$ be the subresultants of index 0 and 1 from $S(r, b, y)$.
- We denote $\overline{s}$ the squarefree part of s.

Hypothesis

The polynomial set $C := \{\overline{s}, g, \ell\}$ is a regular chain.

- The polynomials $\overline{s}, g, \ell$ are non-constant with respective main variables z, y, x.

Hypothesis 2

- Let $S(r, b, y)$ be the subresultant chain of r and b (resp. b and r) regarded as polynomials in $\mathbf{k}[y, z]$ if $\mathrm{mdeg}(r) \geq \mathrm{mdeg}(b)$ (resp. $\mathrm{mdeg}(r) < \mathrm{mdeg}(b)$).
- Let $s := S_0(r, b, y)$ and $g := S_1(r, b, y)$ be the subresultants of index 0 and 1 from $S(r, b, y)$.
- We denote $\overline{s}$ the squarefree part of s.

Hypothesis

The polynomial set $C := \{\overline{s}, g, \ell\}$ is a regular chain.

- The polynomials $\overline{s}, g, \ell$ are non-constant with respective main variables z, y, x.
- The initial h_g is invertible mod $\overline{s}$ and the initial h_ℓ is invertible modulo the ideal $\langle \overline{s}, g \rangle$.

Hypothesis 2

- Let $S(r, b, y)$ be the subresultant chain of r and b (resp. b and r) regarded as polynomials in $\mathbf{k}[y, z]$ if $\mathrm{mdeg}(r) \geq \mathrm{mdeg}(b)$ (resp. $\mathrm{mdeg}(r) < \mathrm{mdeg}(b)$).
- Let $s := S_0(r, b, y)$ and $g := S_1(r, b, y)$ be the subresultants of index 0 and 1 from $S(r, b, y)$.
- We denote $\bar{s}$ the squarefree part of s.

Hypothesis

The polynomial set $C := \{\bar{s}, g, \ell\}$ is a regular chain.

- The polynomials $\bar{s}, g, \ell$ are non-constant with respective main variables z, y, x.
- The initial h_g is invertible mod $\bar{s}$ and the initial h_ℓ is invertible modulo the ideal $\langle \bar{s}, g \rangle$.
- $V(\bar{s}, r, b) = V(\bar{s}, g)$.

Hypothesis 3

Our last genericity assumption is the following

Hypothesis

The initial of the polynomial t is invertible modulo the ideal

$$\mathrm{sat}(\{\bar{s}, g\}) = \langle \bar{s}, g \rangle.$$

Hypothesis 3

Our last genericity assumption is the following

Hypothesis

The initial of the polynomial t is invertible modulo the ideal

$$\mathrm{sat}(\{\bar{s}, g\}) = \langle \bar{s}, g \rangle.$$

- As a consequence we have:

$$V(f) \cap V(\{\bar{s}, g, t\}) \; = \; V(\{\bar{s}, g, \ell\}).$$

Hypothesis 3

Our last genericity assumption is the following

Hypothesis

The initial of the polynomial t is invertible modulo the ideal

$$\mathrm{sat}(\{\bar{s}, g\}) = \langle \bar{s}, g \rangle.$$

- As a consequence we have:

$$V(f) \cap V(\{\bar{s}, g, t\}) \ = \ V(\{\bar{s}, g, \ell\}).$$

- Putting all hypotheses together yield the following theorem:

$$V(f, t, b) = V(\{\bar{s}, g, \ell\}).$$

Hypothesis 3

Our last genericity assumption is the following

Hypothesis

The initial of the polynomial t is invertible modulo the ideal

$$\mathrm{sat}(\{\overline{s}, g\}) = \langle \overline{s}, g \rangle.$$

- As a consequence we have:

$$V(f) \cap V(\{\overline{s}, g, t\}) \ = \ V(\{\overline{s}, g, \ell\}).$$

- Putting all hypotheses together yield the following theorem:

$$V(f, t, b) = V(\{\overline{s}, g, \ell\}).$$

- Therefore, under our hypothese, we have:

$$\mathrm{Intersect}(f, \{t, b\}) = V(\{\overline{s}, g, \ell\}).$$

Example

$R := PolynomialRing([x, y, z]):$

$f := (y + z) * x^\wedge 2 + x + 1;$
$t := z * x^\wedge 2 + y * x + 1;$
$b := (z+1) * y^\wedge 2 + y + 2;$

$$f := (y+z)\, x^2 + x + 1$$
$$t := z\, x^2 + y\, x + 1$$
$$b := (z+1)\, y^2 + y + 2 \qquad (1)$$

$src1 := SubresultantChain(f, t, x, R):$
$l := SubresultantOfIndex(1, src1, R);\ r := SubresultantOfIndex(0, src1, R);$

$$l := x\, y^2 + x\, y\, z - x\, z + y$$
$$r := y^3 + y^2\, z - 2\, y\, z + z \qquad (2)$$

$src2 := SubresultantChain(r, b, y, R):$
$g := SubresultantOfIndex(1, src2, R);\ \ s := SubresultantOfIndex(0, src2, R);$

$$g := -2\, y\, z^3 - 5\, y\, z^2 + z^3 - 5\, y\, z - y - z + 2$$
$$s := z^5 + 9\, z^4 + 24\, z^3 + 38\, z^2 + 13\, z + 8 \qquad (3)$$

$sol := Chain([s], Empty(R), R) : IsRegular(Initial(g, R), sol, R);$

$$true \tag{4}$$

$sol2 := Chain([g], sol, R) : IsRegular(Initial(l, R), sol2, R);$

$$true \tag{5}$$

$IsRegular(Initial(t, R), sol2, R);$

$$true \tag{6}$$

$sol3 := Chain([l], sol2, R) : Display(sol3, R);$

$$\begin{cases} (y^2 + yz - z)\, x + y = 0 \\ (-2z^3 - 5z^2 - 5z - 1)\, y + z^3 - z + 2 = 0 \\ z^5 + 9z^4 + 24z^3 + 38z^2 + 13z + 8 = 0 \\ y^2 + yz - z \neq 0 \\ -2z^3 - 5z^2 - 5z - 1 \neq 0 \end{cases} \tag{7}$$

$dec3 := Triangularize([f, t, b], R) : Display(dec3[1], R);$

$$\begin{cases} (y^2 + yz - z)\, x + y = 0 \\ (2z^3 + 5z^2 + 5z + 1)\, y - z^3 + z - 2 = 0 \\ z^5 + 9z^4 + 24z^3 + 38z^2 + 13z + 8 = 0 \\ y^2 + yz - z \neq 0 \\ 2z^3 + 5z^2 + 5z + 1 \neq 0 \end{cases} \tag{8}$$

Outline

The Modular Method

The key ideas of that method are

- Computing the subresultants $r = S_0(t, f, x)$, $\ell = S_1(t, f, x)$, $s = S_0(r, b, y)$, $g = S_1(r, b, y)$ by evaluation and interpolation.

The Modular Method

The key ideas of that method are

- Computing the subresultants $r = S_0(t, f, x)$, $\ell = S_1(t, f, x)$, $s = S_0(r, b, y)$, $g = S_1(r, b, y)$ by evaluation and interpolation.
- Use the Bézout bound of the (zero-dimensional) variety $V(f, t, b)$ for this evaluation and interpolation process.

The Modular Method

The key ideas of that method are

- Computing the subresultants $r = S_0(t, f, x)$, $\ell = S_1(t, f, x)$, $s = S_0(r, b, y)$, $g = S_1(r, b, y)$ by evaluation and interpolation.
- Use the Bézout bound of the (zero-dimensional) variety $V(f, t, b)$ for this evaluation and interpolation process.
- Verify the genericity assumptions as we recover ℓ, s, g from the evaluation and interpolation process, returning an error if one of those assumptions is not met.

The Modular Method

The key ideas of that method are

- Computing the subresultants $r = S_0(t, f, x)$, $\ell = S_1(t, f, x)$, $s = S_0(r, b, y)$, $g = S_1(r, b, y)$ by evaluation and interpolation.
- Use the Bézout bound of the (zero-dimensional) variety $V(f, t, b)$ for this evaluation and interpolation process.
- Verify the genericity assumptions as we recover ℓ, s, g from the evaluation and interpolation process, returning an error if one of those assumptions is not met.
- Use only basic well optimized functions of maple. In particular the **modp1** library.

The implementation

while $i \leq bnd := 2 * BezoutBdn + 1$ **do**

  Select a point v and specialize f, t, b at $z = v$.

  if f, t, b does not specialize well **then**

    Next

  end if

  Normalize T to $T_v = \{t_v, b_v\}$.

  Compute $r_v = S_0(t_v, f_v, x)$, $\ell_v = S_1(t_v, f_v, x)$.

  Check assumptions about r.

  Compute $s_v = S_0(r_v, b_v, y)$, $g_v = S_1(r_v, b_v, y)$.

end while

Interpolate s_v, g_v, ℓ_v into s, g, ℓ.

Apply Rational Function Reconstruction to s, g, ℓ.

Get the numerator of s, g, ℓ.

Compute the squarefree part of s, $\bar{s}$.

Check that $C = \{\bar{s}, g, \ell\}$ is a regular chain and the initial of t.

Outline

A first example

- Prime characteristic 469762049.

A first example

- Prime characteristic 469762049.
- $t = zx^2 + yx + 1$,
- $b = (z+1)y^2 + y + 2$,
- $f = (y+z)x^2 + x + 1$.

A first example

- Prime characteristic 469762049.
- $t = zx^2 + yx + 1$,
- $b = (z + 1)y^2 + y + 2$,
- $f = (y + z)x^2 + x + 1$.
- **Intersect time:** 0.010s, **Intersect by Specialization time:** 0.114s

A first example

- Prime characteristic 469762049.
- $t = zx^2 + yx + 1$,
- $b = (z + 1)y^2 + y + 2$,
- $f = (y + z)x^2 + x + 1$.
- **Intersect time:** 0.010s, **Intersect by Specialization time:** 0.114s
- $s = z^5 + 9z^4 + 2 * z^3 + 38z^2 + 13z + 8$.

A first example

- Prime characteristic 469762049.
- $t = zx^2 + yx + 1$,
- $b = (z+1)y^2 + y + 2$,
- $f = (y+z)x^2 + x + 1$.
- **Intersect time:** 0.010s, **Intersect by Specialization time:** 0.114s
- $s = z^5 + 9z^4 + 2 * z^3 + 38z^2 + 13z + 8$.
- $g = z^3 + 469762048z + 2 + (469762047z^3 + 469762044z^2 + 469762044z + 469762048)y$.

A first example

- Prime characteristic 469762049.
- $t = zx^2 + yx + 1$,
- $b = (z+1)y^2 + y + 2$,
- $f = (y+z)x^2 + x + 1$.
- **Intersect time:** 0.010s, **Intersect by Specialization time:** 0.114s
- $s = z^5 + 9z^4 + 2 * z^3 + 38z^2 + 13z + 8$.
- $g = z^3 + 469762048z + 2 + (469762047z^3 + 469762044z^2 + 469762044z + 469762048)y$.
- $\ell = z^3 + 469762048z + 2 + (469762047z^3 + 469762044z^2 + 469762044z + 469762048)y$.

A second example

- Prime characteristic 469762049.

A second example

- Prime characteristic 469762049.
- $t = 4x^9 - 40x^5y^2z + 6x^3y^3z + 27xy^6 + 68xy^3z^2 - 11z^5$,
- $b = -33y^8z + 8y^5z^2 - 69y^4z^2 - 34z^6 - 58y^5 - 53yz^2$,
- $f = -7x^3y^2z^4 - 50y^4z^5 - 70x^3y^5 + 19xy^5 - 5y^3z + 48x$.

A second example

- Prime characteristic 469762049.
- $t = 4x^9 - 40x^5y^2z + 6x^3y^3z + 27xy^6 + 68xy^3z^2 - 11z^5$,
- $b = -33y^8z + 8y^5z^2 - 69y^4z^2 - 34z^6 - 58y^5 - 53yz^2$,
- $f = -7x^3y^2z^4 - 50y^4z^5 - 70x^3y^5 + 19xy^5 - 5y^3z + 48x$.
- **Intersect time:** 298.017s, **Intersect by Specialization time:** 30.187s

A second example

- Prime characteristic 469762049.
- $t = 4x^9 - 40x^5y^2z + 6x^3y^3z + 27xy^6 + 68xy^3z^2 - 11z^5$,
- $b = -33y^8z + 8y^5z^2 - 69y^4z^2 - 34z^6 - 58y^5 - 53yz^2$,
- $f = -7x^3y^2z^4 - 50y^4z^5 - 70x^3y^5 + 19xy^5 - 5y^3z + 48x$.
- **Intersect time:** 298.017s, **Intersect by Specialization time:** 30.187s
- $\deg(s) = 573$, $\deg(g) = 504$, $\deg(\ell) = 64$.

Benchmark

Prime characteristic 469762049.

N	deg(t)	deg(b)	deg(f)	Intersect	Intersect by Specialization
1	3	2	3	0.123s	0.476s
2	5	4	4	0.312s	1.020s
3	5	4	5	0.412s	1.406s
4	7	6	7	14.652s	12.576s
5	7	7	7	1.509s	9.755s
6	8	7	8	37.540s	35.174s
7	8	8	8	33.720s	21.716s
8	9	6	9	28.545s	13.705s

Benchmark

Prime characteristic 469762049.

N	deg(t)	deg(b)	deg(f)	Intersect	Intersect by Specialization
9	8	7	8	37.540s	35.174s
10	8	8	8	33.720s	21.716s
11	8	8	8	31.280s	19.766s
12	8	8	8	21.285s	15.244s
13	8	8	8	24.387s	13.059s
14	8	8	8	45.607s	16.406s
15	8	8	8	46.862s	18.717s
16	8	8	8	46.862s	18.717s
17	8	8	9	68.167s	23.177s
18	9	9	8	34.497s	29.235s
19	9	6	9	28.545s	13.705s

Outline

Concluding remarks

Theoretical aspects

- We have presented a modular algorithm for solving a trivariate polynomial system:
$$f = t = b = 0$$
under genericity assumptions.
- To be more precise, this is a modular method for the call Intersect($f, \{t, b\}$).
- The article to-be-submitted to the Maple conference proceedings will explain how to relax the genericity assumptions
- A follow-up article will extend this modular method to solve square systems with an arbitrary number of variables.

Concluding remarks

Practical aspects

- The preliminary implementation and experimentation in Maple bring promising results for this modular method.
- To be more precise, for generic input systems of sufficiently large Bézout bound, the modular method outperforms the non-modular implementation of the command Intersect.
- There is large room for improvement: indeed, this modular method opens the door to using speculative algorithms for computing subresultants. Those are asymptotically fast algorithms that:

Concluding remarks

Practical aspects

- The preliminary implementation and experimentation in Maple bring promising results for this modular method.
- To be more precise, for generic input systems of sufficiently large Bézout bound, the modular method outperforms the non-modular implementation of the command Intersect.
- There is large room for improvement: indeed, this modular method opens the door to using speculative algorithms for computing subresultants. Those are asymptotically fast algorithms that:
  - compute the subresultants of index 0 and 1 without computing the other subresultants,

Concluding remarks

Practical aspects

- The preliminary implementation and experimentation in Maple bring promising results for this modular method.
- To be more precise, for generic input systems of sufficiently large Bézout bound, the modular method outperforms the non-modular implementation of the command Intersect.
- There is large room for improvement: indeed, this modular method opens the door to using speculative algorithms for computing subresultants. Those are asymptotically fast algorithms that:
  - compute the subresultants of index 0 and 1 without computing the other subresultants,
  - while being able to resume the computations for obtaining the subresultants of higher index, if needed.