Polynomial systems, sparsity, and applications

Matías R. Bender

February 25, 2022

Short CV

- Since 2019, PostDoc at Institut für Mathematik, TU Berlin.
  (Mentor: P. Bürgisser)
- In 2019, PhD in Informatics from Sorbonne Université.
  (Advisors: J.-C. Faugère & E. Tsigaridas)
- In 2015, Bsc+Msc in Computer Science from Univ. de Buenos Aires.

Supported by the ERC under the European's Horizon 2020 research and innovation programme (grant no. 787840)

It is all about the balance

	Model	Computation
Linear algebra	Linear	Easy
Non-linear algebra	Non-linear	Hard

It is all about the balance

	Model	Computation
Linear algebra	Linear	Easy
Non-linear algebra	Non-linear	Hard

Pragmatic approach $\rightarrow$ Use more powerful models,
if we can compute efficiently with them.

It is all about the balance

	Model	Computation
Linear algebra	Linear	Easy
Non-linear algebra	Non-linear	Hard

Pragmatic approach $\rightarrow$ Use more powerful models,
if we can compute efficiently with them.

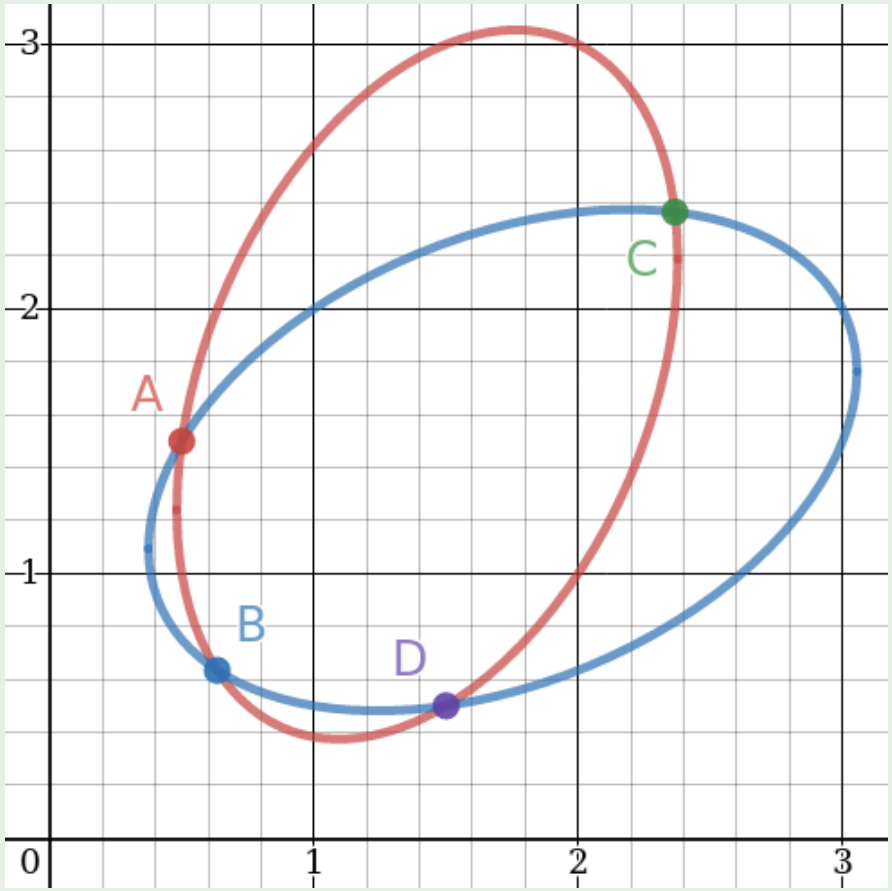
I study how to exploit ther inputs' structure to compute faster.

I work in computer algebra. I focus on

I work in computer algebra. I focus on

Solving structured polynomial systems

- Solving polynomial systems of degree at most d in $\mathbb{C}[x_1, \ldots, x_n]$.



$$\begin{cases} 2x^2 - xy + y^2 - 4x - 2y + 3 = 0 \\ x^2 - xy + 2y^2 - 2x - 4y + 3 = 0 \end{cases}$$

$$\boxed{\begin{array}{c} d = 2 \\ n = 2 \end{array}}$$

I work in computer algebra. I focus on

Solving structured polynomial systems

- Solving polynomial systems of degree at most d in $\mathbb{C}[x_1, \ldots, x_n]$.

$$\begin{cases} 2x^2 - xy + y^2 - 4x - 2y + 3 = 0 \\ x^2 - xy + 2y^2 - 2x - 4y + 3 = 0 \end{cases} \qquad \boxed{\begin{array}{c} d = 2 \\ n = 2 \end{array}}$$

- Different methods: symbolic and numerical

I work in computer algebra. I focus on

Solving structured polynomial systems

- Solving polynomial systems of degree at most d in $\mathbb{C}[x_1, \ldots, x_n]$.

$$\begin{cases} 2x^2 - xy + y^2 - 4x - 2y + 3 = 0 \\ x^2 - xy + 2y^2 - 2x - 4y + 3 = 0 \end{cases} \qquad \boxed{\begin{array}{c} d = 2 \\ n = 2 \end{array}}$$

- Different methods: symbolic and numerical

  (Today: Gröbner bases, resultants, numerical linear algebra approaches)

I work in computer algebra. I focus on

Solving structured polynomial systems

- Solving polynomial systems of degree at most d in $\mathbb{C}[x_1, \ldots, x_n]$.

$$\begin{cases} 2x^2 - xy + y^2 - 4x - 2y + 3 = 0 \\ x^2 - xy + 2y^2 - 2x - 4y + 3 = 0 \end{cases} \qquad \boxed{\begin{array}{c} d = 2 \\ n = 2 \end{array}}$$

- Different methods: symbolic and numerical

  (Today: Gröbner bases, resultants, numerical linear algebra approaches)

- It is an intrinsically hard problem $\rightarrow$ complexity $d^{O(n)}$.

I work in computer algebra. I focus on

Solving structured polynomial systems

- Solving polynomial systems of degree at most d in $\mathbb{C}[x_1, \ldots, x_n]$.

Dense
$$\begin{cases} 2x^2 - xy + y^2 - 4x - 2y + 3 = 0 \\ x^2 - xy + 2y^2 - 2x - 4y + 3 = 0 \end{cases} \qquad \boxed{\begin{array}{c} d = 2 \\ n = 2 \end{array}}$$

- Different methods: symbolic and numerical

  (Today: Gröbner bases, resultants, numerical linear algebra approaches)

- It is an intrinsically hard problem $\rightarrow$ complexity $d^{O(n)}$.

- In applications $\rightarrow$ the problems have some structure.

Sparse
$$\begin{cases} -xy - 4x + 3 = 0 \\ 2y^2 - 2x - 4y + 3 = 0 \end{cases}$$

I work in computer algebra. I focus on

Solving structured polynomial systems

- Solving polynomial systems of degree at most d in $\mathbb{C}[x_1, \ldots, x_n]$.

Dense
$$\begin{cases} 2x^2 - xy + y^2 - 4x - 2y + 3 = 0 \\ x^2 - xy + 2y^2 - 2x - 4y + 3 = 0 \end{cases} \qquad \boxed{\begin{array}{c} d = 2 \\ n = 2 \end{array}}$$

- Different methods: symbolic and numerical

  (Today: Gröbner bases, resultants, numerical linear algebra approaches)

- It is an intrinsically hard problem $\rightarrow$ complexity $d^{O(n)}$.

- In applications $\rightarrow$ the problems have some structure.

Sparse
$$\begin{cases} -xy - 4x + 3 = 0 \\ 2y^2 - 2x - 4y + 3 = 0 \end{cases}$$

- To improve complexity $\rightarrow$ exploit sparsity.

I work in computer algebra. I focus on

Solving structured polynomial systems

- Solving polynomial systems of degree at most d in $\mathbb{C}[x_1, \ldots, x_n]$.

Dense
$$\begin{cases} 2x^2 - xy + y^2 - 4x - 2y + 3 = 0 \\ x^2 - xy + 2y^2 - 2x - 4y + 3 = 0 \end{cases} \qquad \boxed{\begin{array}{c} d = 2 \\ n = 2 \end{array}}$$

- Different methods: symbolic and numerical

  (Today: Gröbner bases, resultants, numerical linear algebra approaches)

- It is an intrinsically hard problem $\rightarrow$ complexity $d^{O(n)}$.

- In applications $\rightarrow$ the problems have some structure.

Sparse
$$\begin{cases} -xy - 4x + 3 = 0 \\ 2y^2 - 2x - 4y + 3 = 0 \end{cases}$$

- To improve complexity $\rightarrow$ exploit sparsity.

Applications

Tensor decomposition, topological data analysis, computational biology.

Outline of the talk

1. Classical tools to solve dense polynomial systems
   - Gröbner bases
   - Resultants
2. Sparse polynomials
3. Solving sparse systems
   - Gröbner bases
   - Numerical linear algebra approaches
   - Complete intersections over toric varieties
   - Determinantal formulas for mixed multilinear systems
4. Applications
5. Future work

Gröbner basics

- $\mathbb{C}[\boldsymbol{x}] = \mathbb{C}[x_1, \ldots, x_n]$, polynomial ring in n indeterminates over $\mathbb{C}$.
- Polynomial $\rightarrow \sum_\alpha c_\alpha \boldsymbol{x}^\alpha \in \mathbb{C}[\boldsymbol{x}]$.
- Monomial $\rightarrow \boldsymbol{x}^\alpha$, for $\alpha \in \mathbb{N}^n$.

Gröbner basics

- $\mathbb{C}[\boldsymbol{x}] = \mathbb{C}[x_1, \ldots, x_n]$, polynomial ring in n indeterminates over $\mathbb{C}$.
- Polynomial $\rightarrow \sum_\alpha c_\alpha \boldsymbol{x}^\alpha \in \mathbb{C}[\boldsymbol{x}]$.
- Monomial $\rightarrow \boldsymbol{x}^\alpha$, for $\alpha \in \mathbb{N}^n$.

- Gröbner basis (GB) of $\langle f_1, \ldots, f_r \rangle := \{\sum_{i=1}^r h_i f_i : h_i \in \mathbb{C}[\boldsymbol{x}]\}$
  $\rightarrow$ set of generators $(g_1, \ldots, g_s)$ with special properties.

$$\langle f_1, \ldots, f_r \rangle = \langle g_1, \ldots, g_s \rangle$$

Gröbner basics

- $\mathbb{C}[\boldsymbol{x}] = \mathbb{C}[x_1, \ldots, x_n]$, polynomial ring in n indeterminates over $\mathbb{C}$.
- Polynomial $\rightarrow \sum_\alpha c_\alpha \boldsymbol{x}^\alpha \in \mathbb{C}[\boldsymbol{x}]$.
- Monomial $\rightarrow \boldsymbol{x}^\alpha$, for $\alpha \in \mathbb{N}^n$.

- Gröbner basis (GB) of $\langle f_1, \ldots, f_r \rangle := \{\sum_{i=1}^r h_i f_i : h_i \in \mathbb{C}[\boldsymbol{x}]\}$
  $\rightarrow$ set of generators $(g_1, \ldots, g_s)$ with special properties.

$$\langle f_1, \ldots, f_r \rangle = \langle g_1, \ldots, g_s \rangle$$

It depends on a monomial ordering.

Gröbner basics

- $\mathbb{C}[\boldsymbol{x}] = \mathbb{C}[x_1, \ldots, x_n]$, polynomial ring in n indeterminates over $\mathbb{C}$.
- Polynomial $\rightarrow \sum_\alpha c_\alpha \boldsymbol{x}^\alpha \in \mathbb{C}[\boldsymbol{x}]$.
- Monomial $\rightarrow \boldsymbol{x}^\alpha$, for $\alpha \in \mathbb{N}^n$.

- Gröbner basis (GB) of $\langle f_1, \ldots, f_r \rangle := \{\sum_{i=1}^r h_i f_i : h_i \in \mathbb{C}[\boldsymbol{x}]\}$
  $\rightarrow$ set of generators $(g_1, \ldots, g_s)$ with special properties.

- Generalize Row Echelon Form for linear systems.

Gröbner basics

- $\mathbb{C}[\boldsymbol{x}] = \mathbb{C}[x_1, \ldots, x_n]$, polynomial ring in n indeterminates over $\mathbb{C}$.
- Polynomial $\rightarrow \sum_\alpha c_\alpha \boldsymbol{x}^\alpha \in \mathbb{C}[\boldsymbol{x}]$.
- Monomial $\rightarrow \boldsymbol{x}^\alpha$, for $\alpha \in \mathbb{N}^n$.

- Gröbner basis (GB) of $\langle f_1, \ldots, f_r \rangle := \{\sum_{i=1}^r h_i f_i : h_i \in \mathbb{C}[\boldsymbol{x}]\}$
  $\rightarrow$ set of generators $(g_1, \ldots, g_s)$ with special properties.

- Generalize Row Echelon Form for linear systems.

- We can use it to compute
  - Ideal membership: Given h', there exists $h_1, \ldots, h_r$ such $h' = \sum_i h_i f_i$?

Gröbner basics

- $\mathbb{C}[\boldsymbol{x}] = \mathbb{C}[x_1, \ldots, x_n]$, polynomial ring in n indeterminates over $\mathbb{C}$.
- Polynomial $\rightarrow \sum_\alpha c_\alpha \boldsymbol{x}^\alpha \in \mathbb{C}[\boldsymbol{x}]$.
- Monomial $\rightarrow \boldsymbol{x}^\alpha$, for $\alpha \in \mathbb{N}^n$.

- Gröbner basis (GB) of $\langle f_1, \ldots, f_r \rangle := \{\sum_{i=1}^r h_i f_i : h_i \in \mathbb{C}[\boldsymbol{x}]\}$
  $\rightarrow$ set of generators $(g_1, \ldots, g_s)$ with special properties.

- Generalize Row Echelon Form for linear systems.

- We can use it to compute
  - Ideal membership: Given h', there exists $h_1, \ldots, h_r$ such $h' = \sum_i h_i f_i$?
  - Certify no common solutions over $\mathbb{C}$: $1 = \sum_i h_i f_i$?

Gröbner basics

- $\mathbb{C}[\boldsymbol{x}] = \mathbb{C}[x_1, \ldots, x_n]$, polynomial ring in n indeterminates over $\mathbb{C}$.
- Polynomial $\rightarrow \sum_\alpha c_\alpha \boldsymbol{x}^\alpha \in \mathbb{C}[\boldsymbol{x}]$.
- Monomial $\rightarrow \boldsymbol{x}^\alpha$, for $\alpha \in \mathbb{N}^n$.

- Gröbner basis (GB) of $\langle f_1, \ldots, f_r \rangle := \{\sum_{i=1}^r h_i f_i : h_i \in \mathbb{C}[\boldsymbol{x}]\}$
  $\rightarrow$ set of generators $(g_1, \ldots, g_s)$ with special properties.

- Generalize Row Echelon Form for linear systems.

- We can use it to compute
  - Ideal membership: Given h', there exists $h_1, \ldots, h_r$ such $h' = \sum_i h_i f_i$?
  - Certify no common solutions over $\mathbb{C}$: $1 = \sum_i h_i f_i$?
  - Solve: $\{p \in \mathbb{C}^n : f_1(p) = \cdots = f_r(p) = 0\}$.

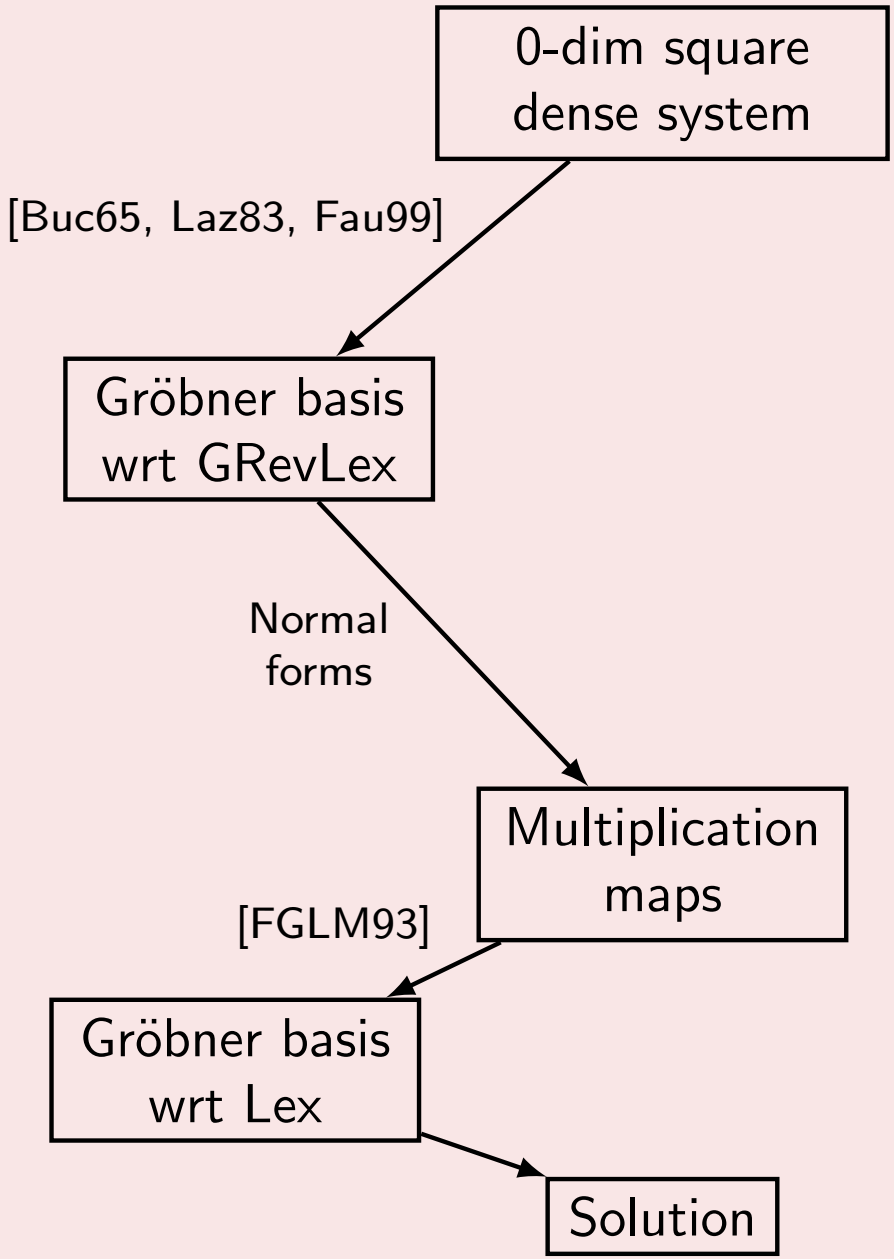
Gröbner basics

- $\mathbb{C}[\boldsymbol{x}] = \mathbb{C}[x_1, \ldots, x_n]$, polynomial ring in n indeterminates over $\mathbb{C}$.
- Polynomial $\rightarrow \sum_\alpha c_\alpha \boldsymbol{x}^\alpha \in \mathbb{C}[\boldsymbol{x}]$.
- Monomial $\rightarrow \boldsymbol{x}^\alpha$, for $\alpha \in \mathbb{N}^n$.

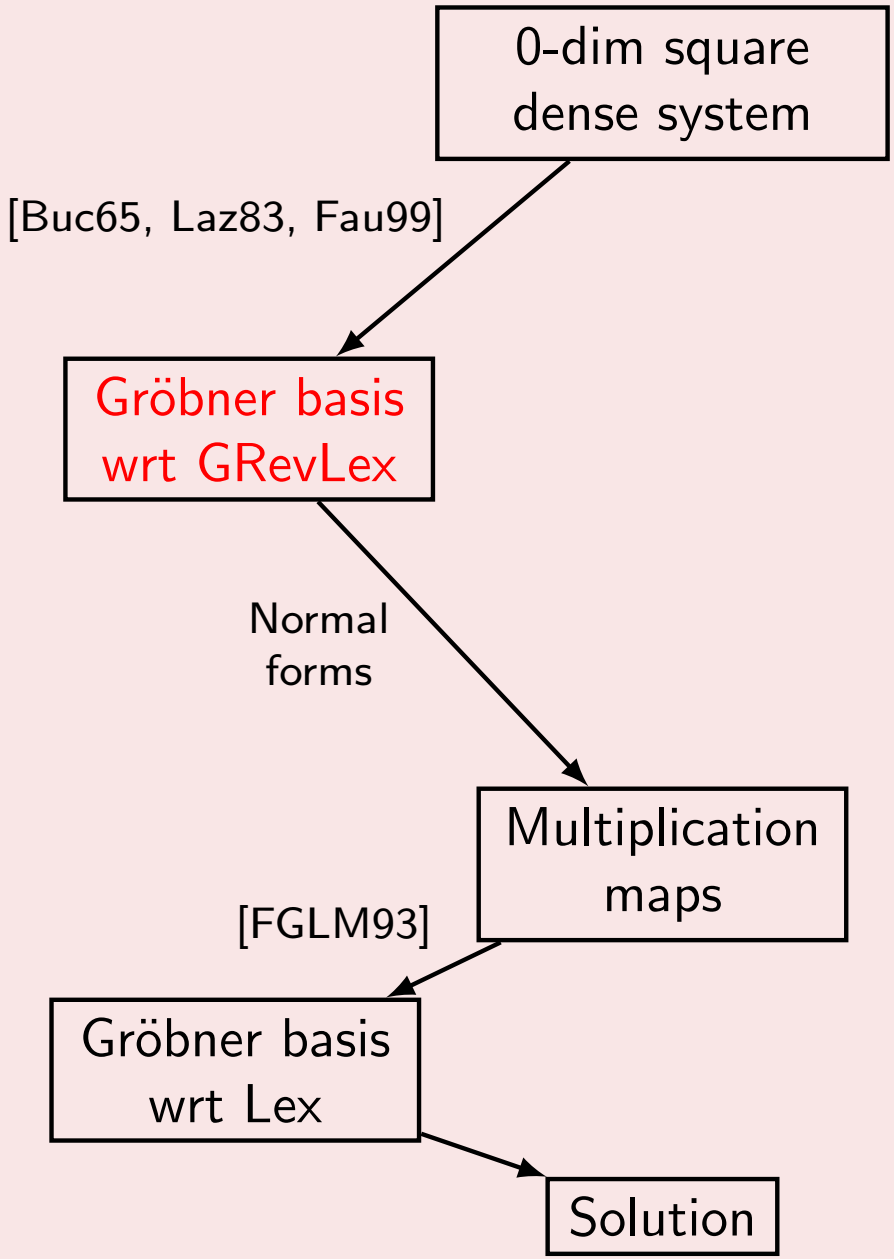
- Gröbner basis (GB) of $\langle f_1, \ldots, f_r \rangle := \{\sum_{i=1}^r h_i f_i : h_i \in \mathbb{C}[\boldsymbol{x}]\}$
  $\rightarrow$ set of generators $(g_1, \ldots, g_s)$ with special properties.

- Generalize Row Echelon Form for linear systems.

- We can use it to compute
  - Ideal membership: Given h', there exists $h_1, \ldots, h_r$ such $h' = \sum_i h_i f_i$?
  - Certify no common solutions over $\mathbb{C}$: $1 = \sum_i h_i f_i$?
  - Solve: $\{p \in \mathbb{C}^n : f_1(p) = \cdots = f_r(p) = 0\}$.
  - Compute other algebraic and geometric invariants.



Under suitable assumptions

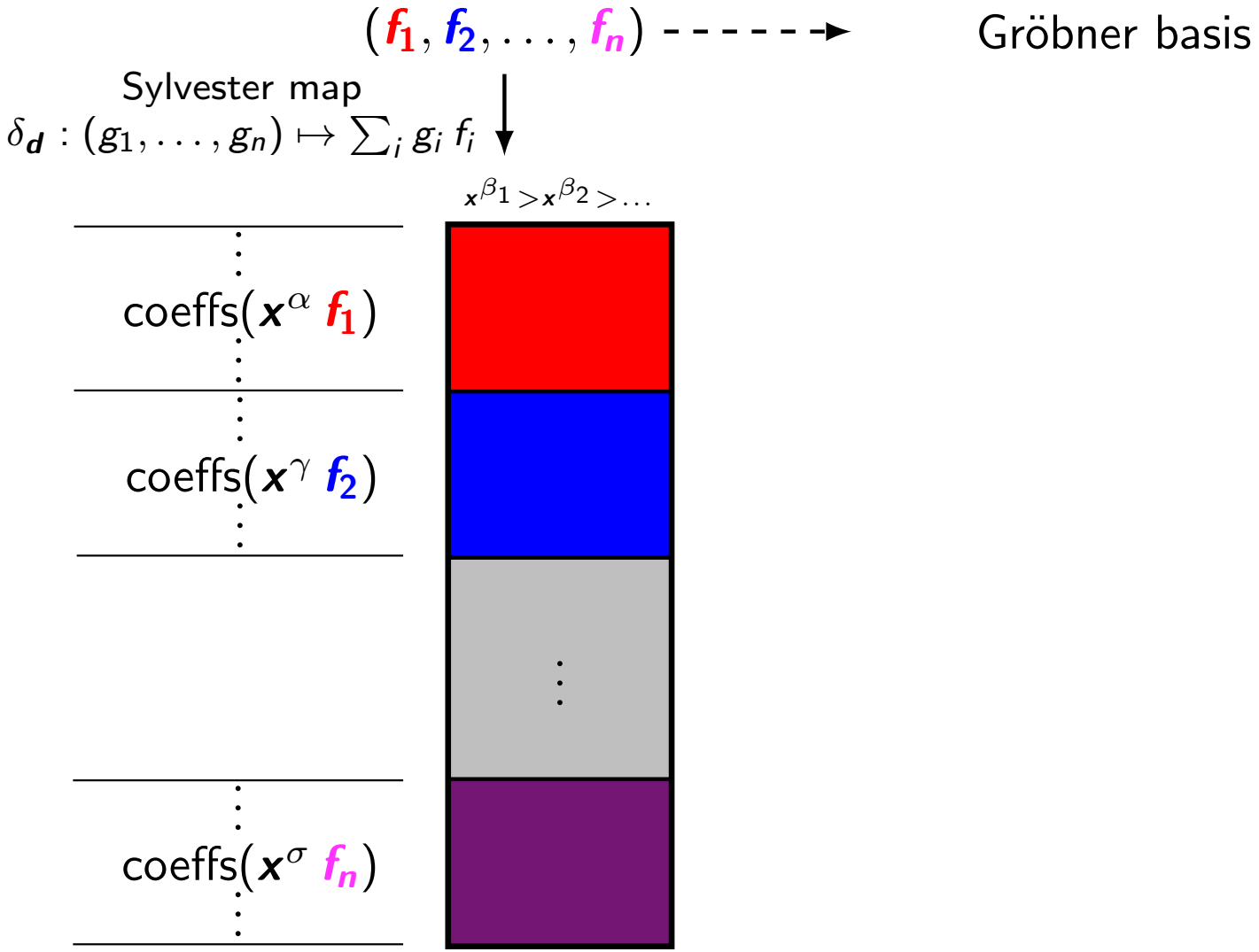


Under suitable assumptions

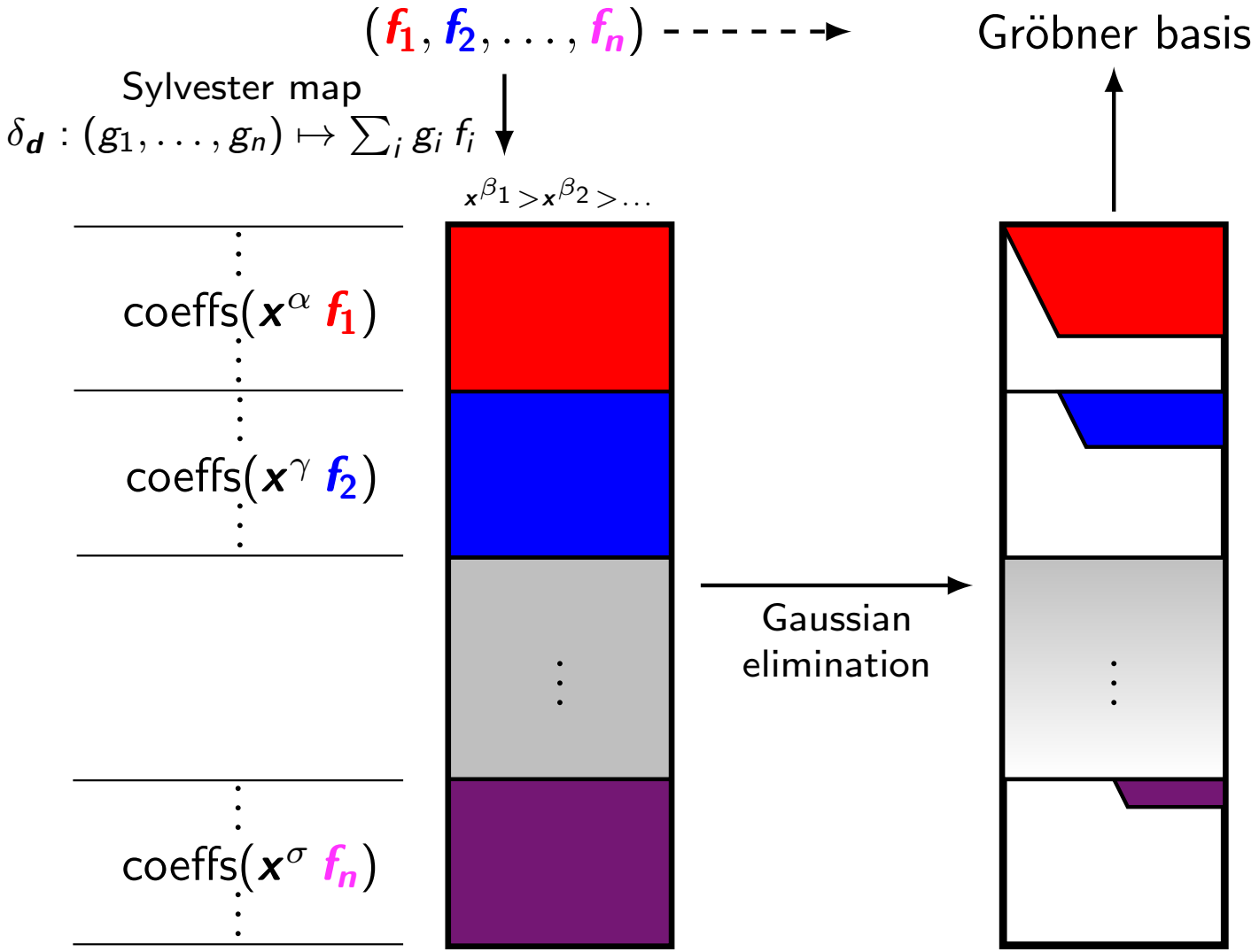
Computing Gröbner bases - [Lazard '83]

$(f_1, f_2, \ldots, f_n)$ - - - - - - ➤ Gröbner basis

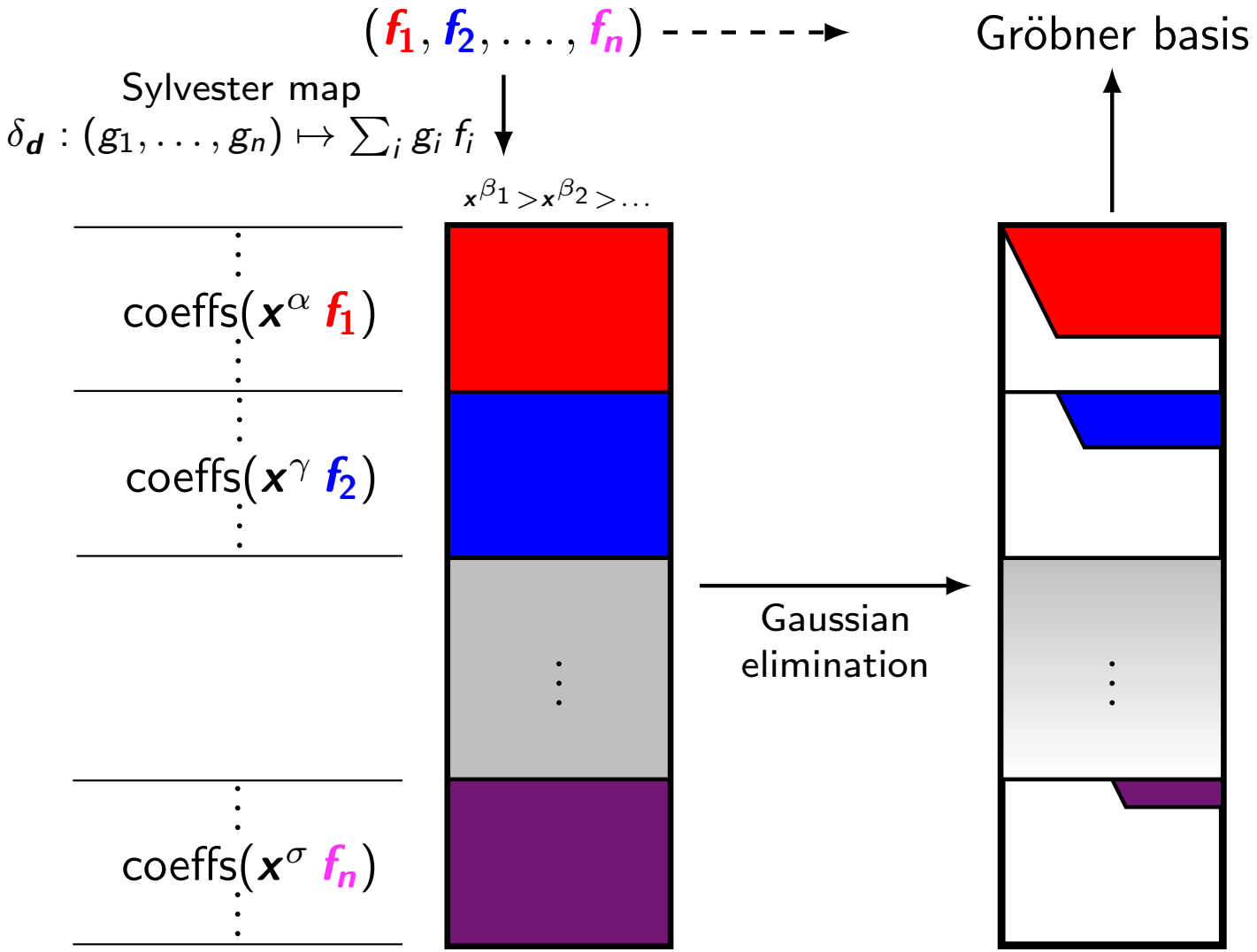
Computing Gröbner bases - [Lazard '83]



Computing Gröbner bases - [Lazard '83]

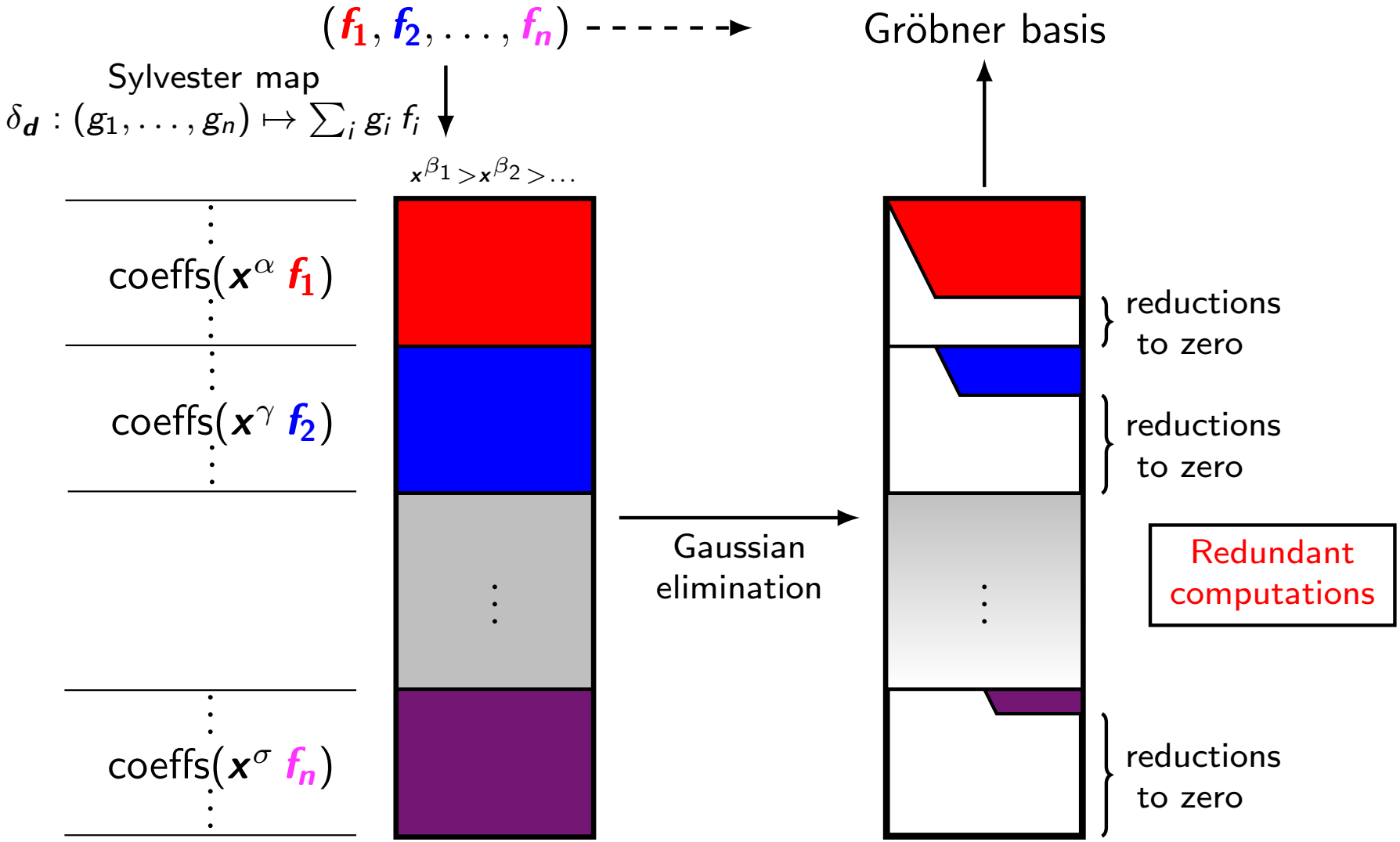


Computing Gröbner bases - [Lazard '83]



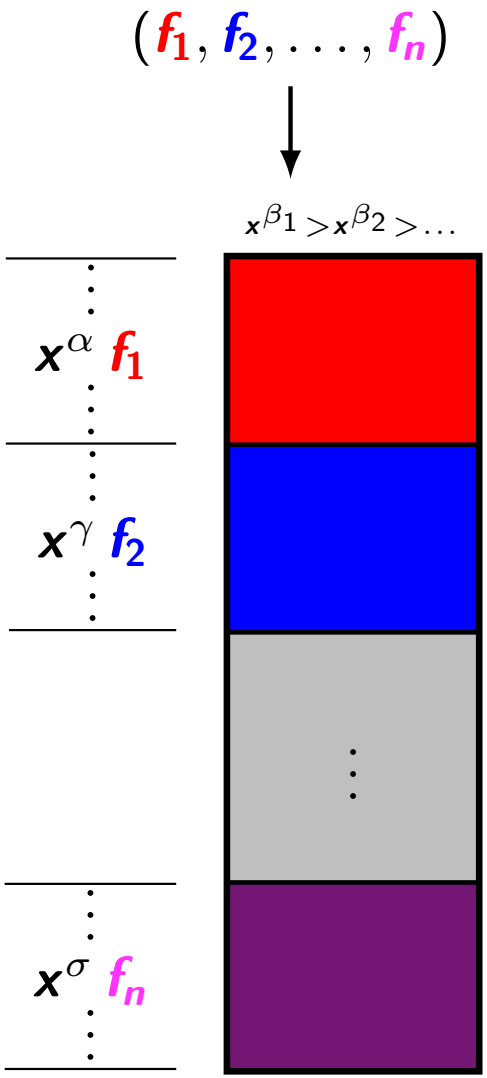
Complexity $\leftrightarrow$ Size of matrix $\rightarrow$ In generic coord., $d \leq$ Castelnuovo-Mumford (CM) reg.
[Bayer, Stillman '87]

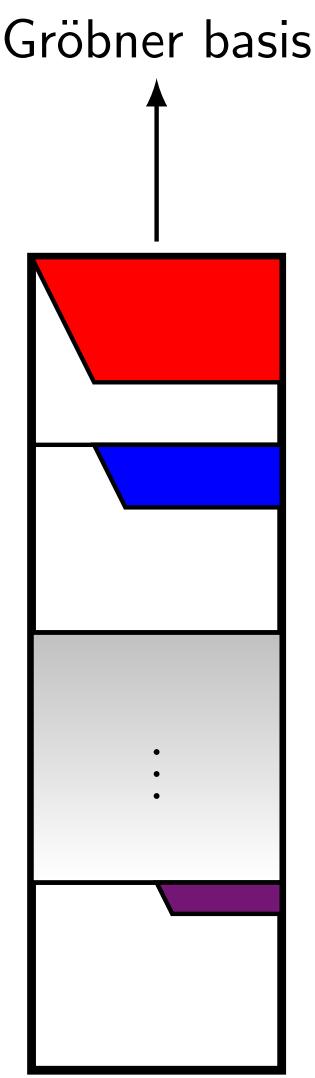
Computing Gröbner bases - [Lazard '83]



Complexity $\leftrightarrow$ Size of matrix $\rightarrow$ In generic coord., $d \leq$ Castelnuovo-Mumford (CM) reg.
[Bayer, Stillman '87]

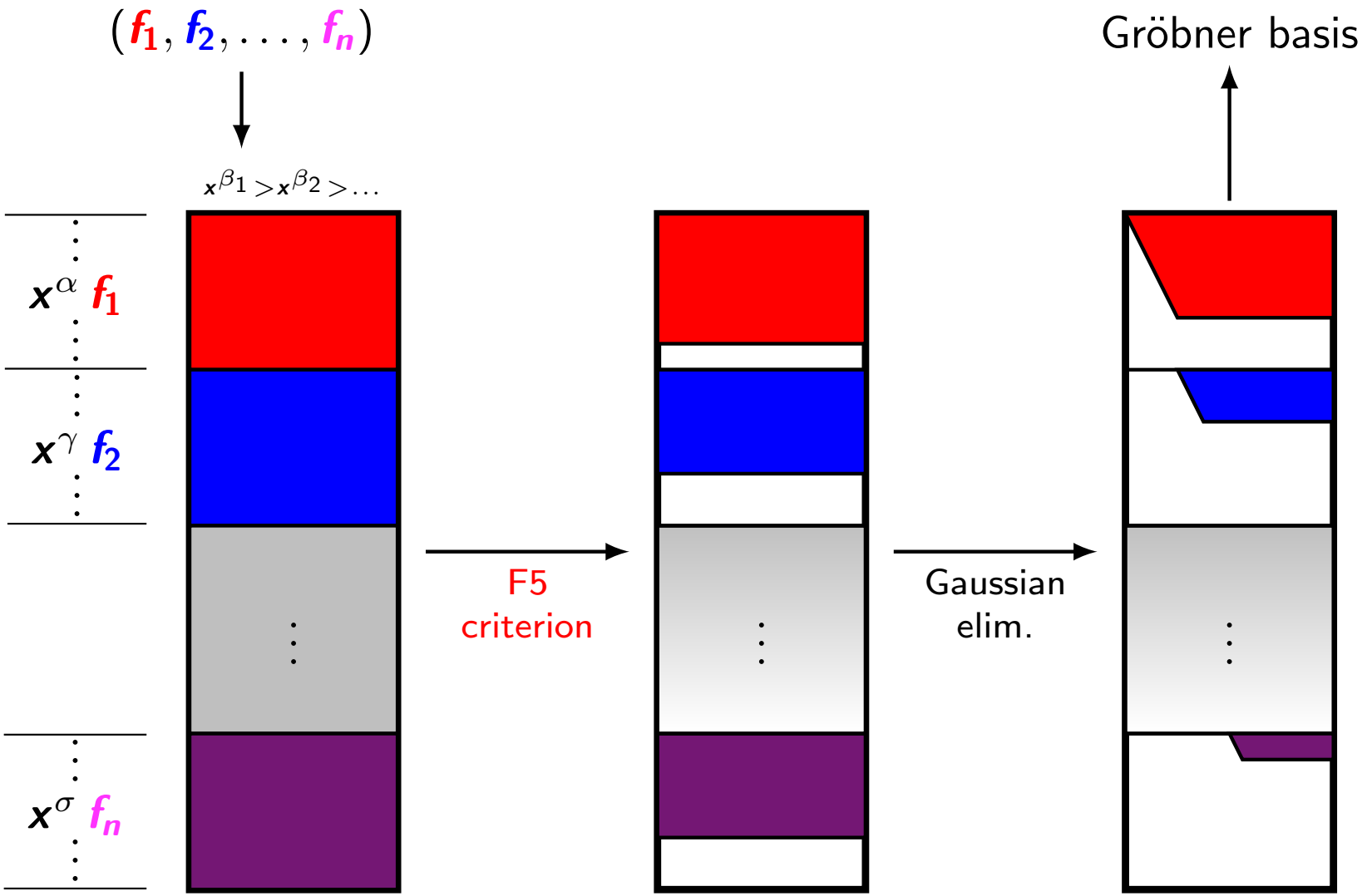
Computing Gröbner bases - [Lazard '83] + [Faugère '02]





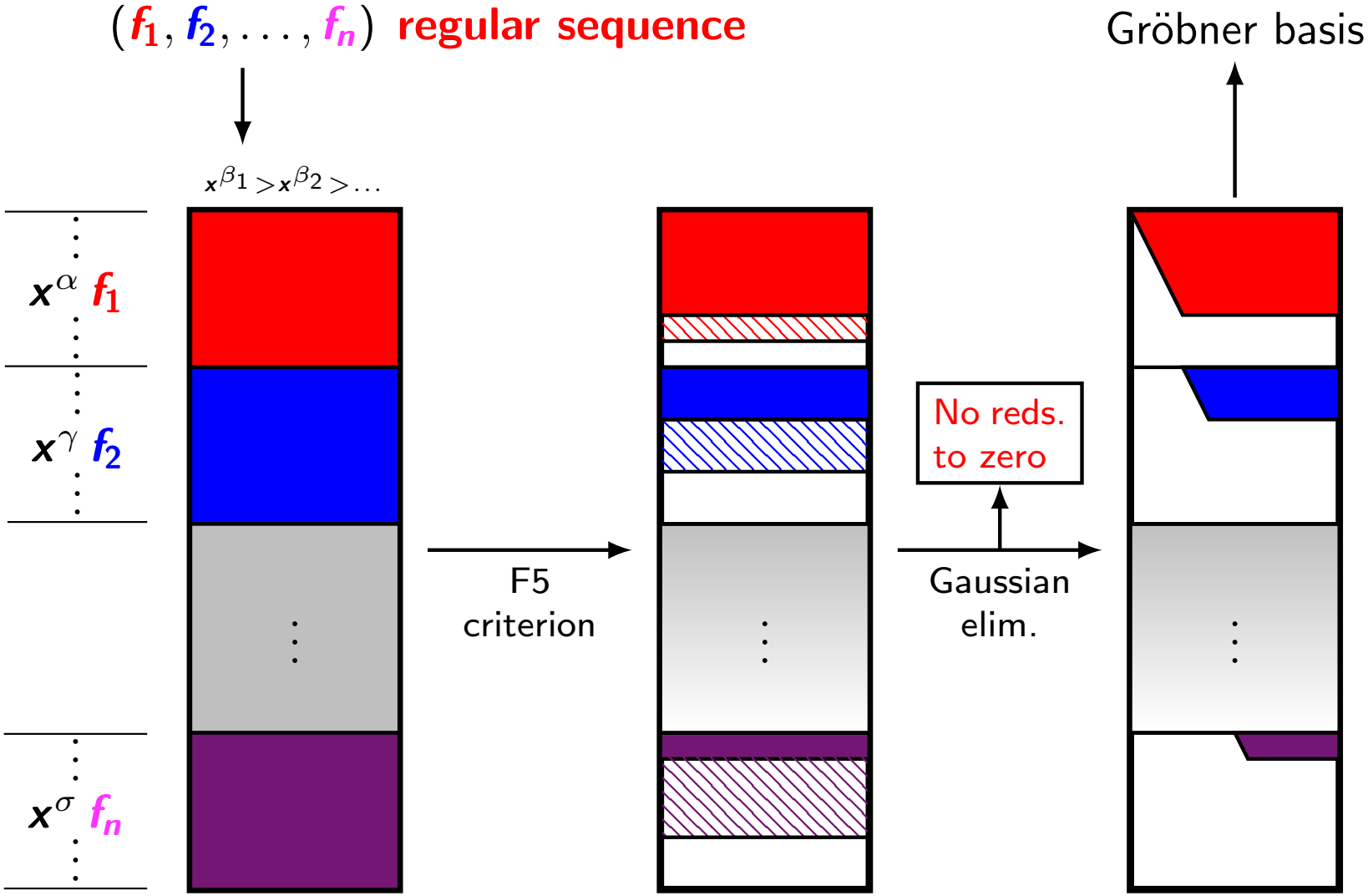
Complexity $\leftrightarrow$ Size of matrix $\rightarrow$ In generic coord., $\boldsymbol{d} \leq$ Castelnuovo-Mumford (CM) reg.

Computing Gröbner bases - [Lazard '83] + [Faugère '02]



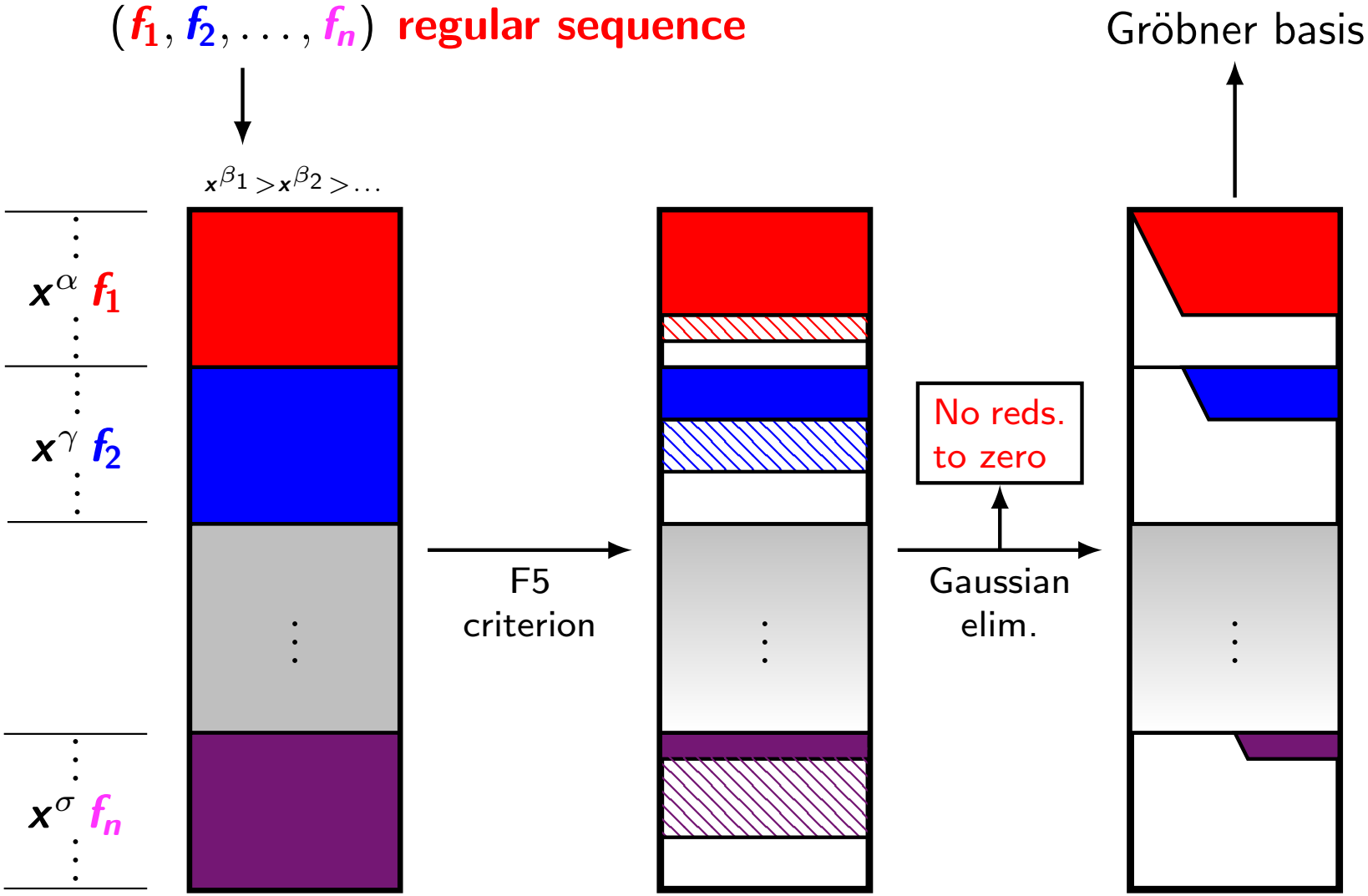
Complexity $\leftrightarrow$ Size of matrix $\rightarrow$ In generic coord., $d \leq$ Castelnuovo-Mumford (CM) reg.

Computing Gröbner bases - [Lazard '83] + [Faugère '02]



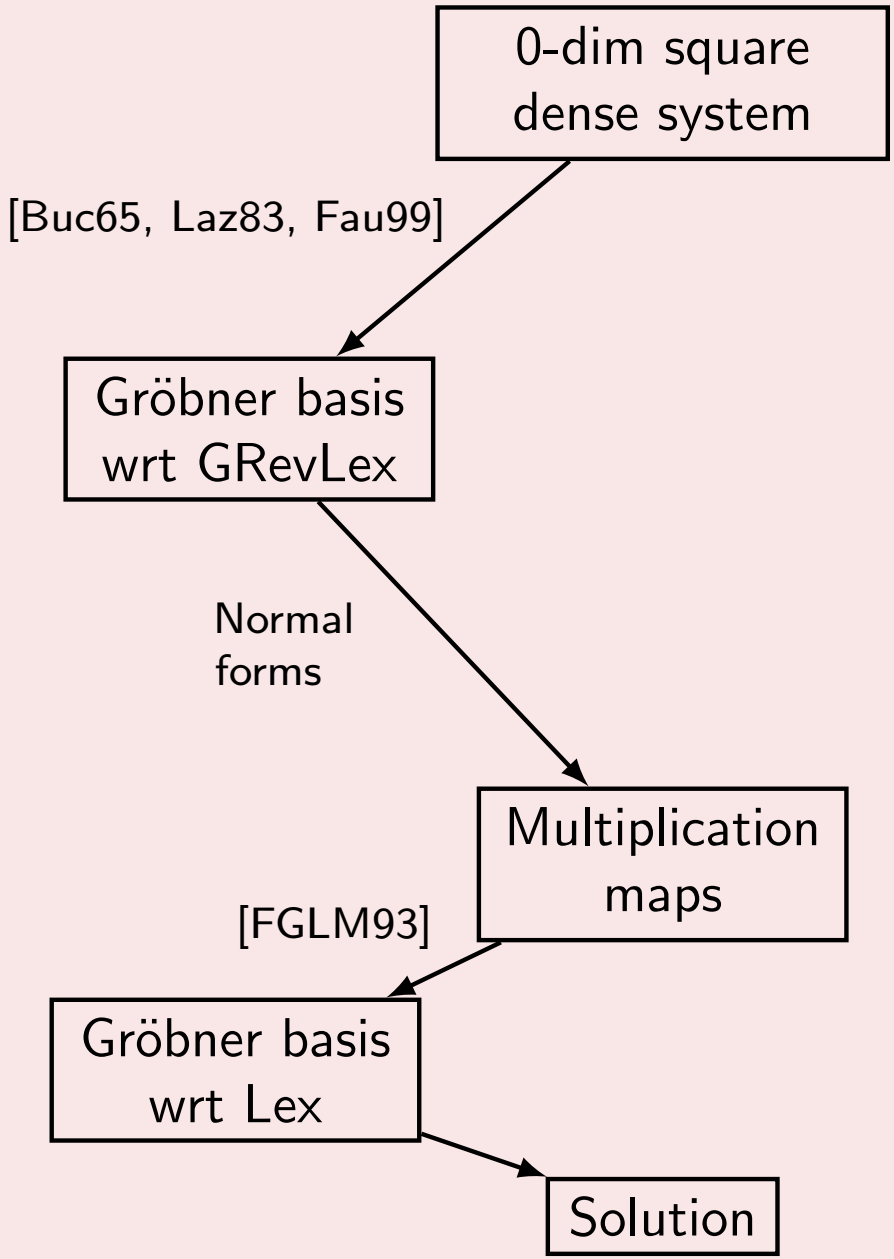
Complexity $\leftrightarrow$ Size of matrix $\rightarrow$ In generic coord., $d \leq$ Castelnuovo-Mumford (CM) reg.

Computing Gröbner bases - [Lazard '83] + [Faugère '02]

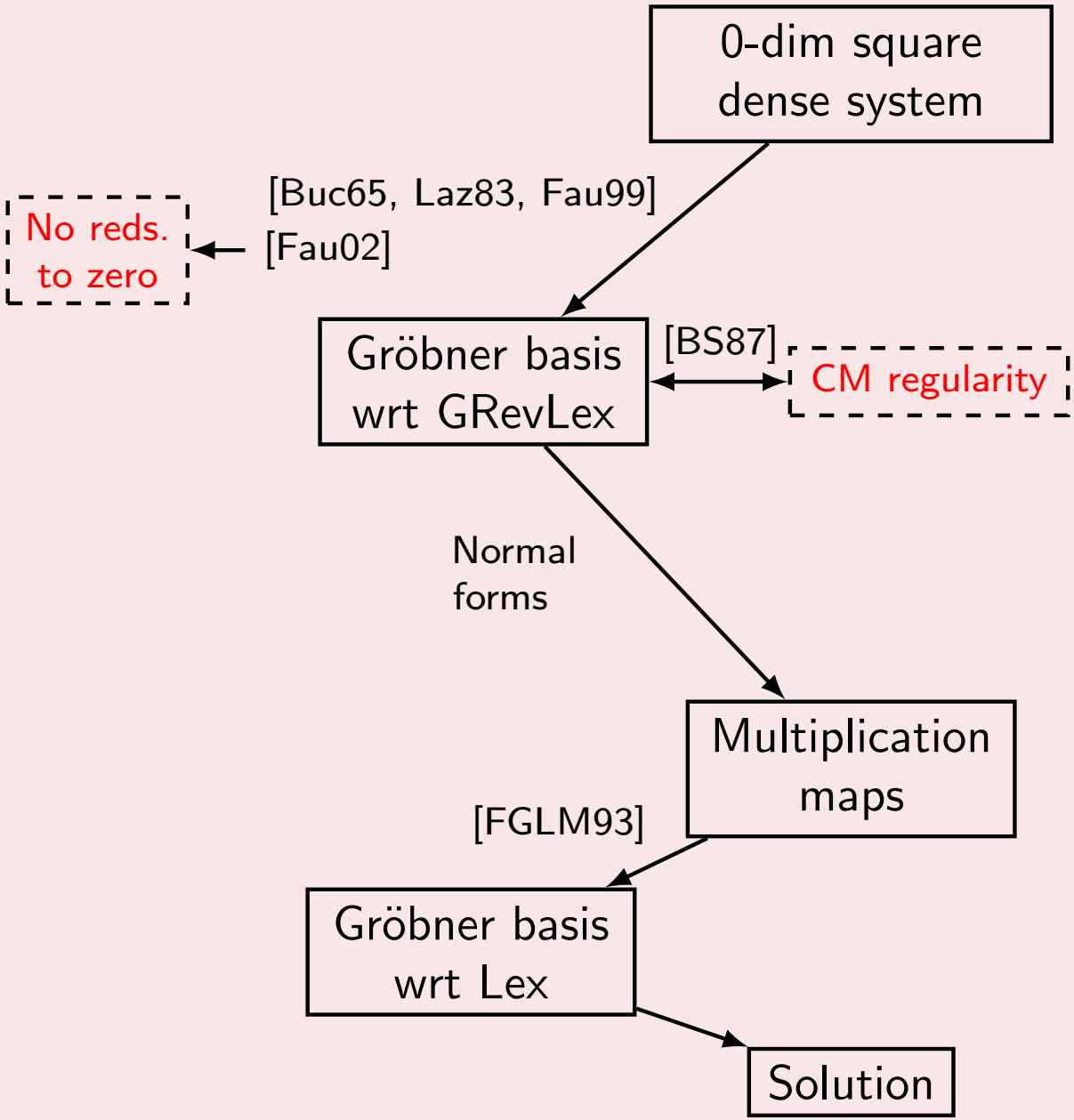


Complexity $\leftrightarrow$ Size of matrix $\rightarrow$ In generic coord., $d \leq$ Castelnuovo-Mumford (CM) reg.

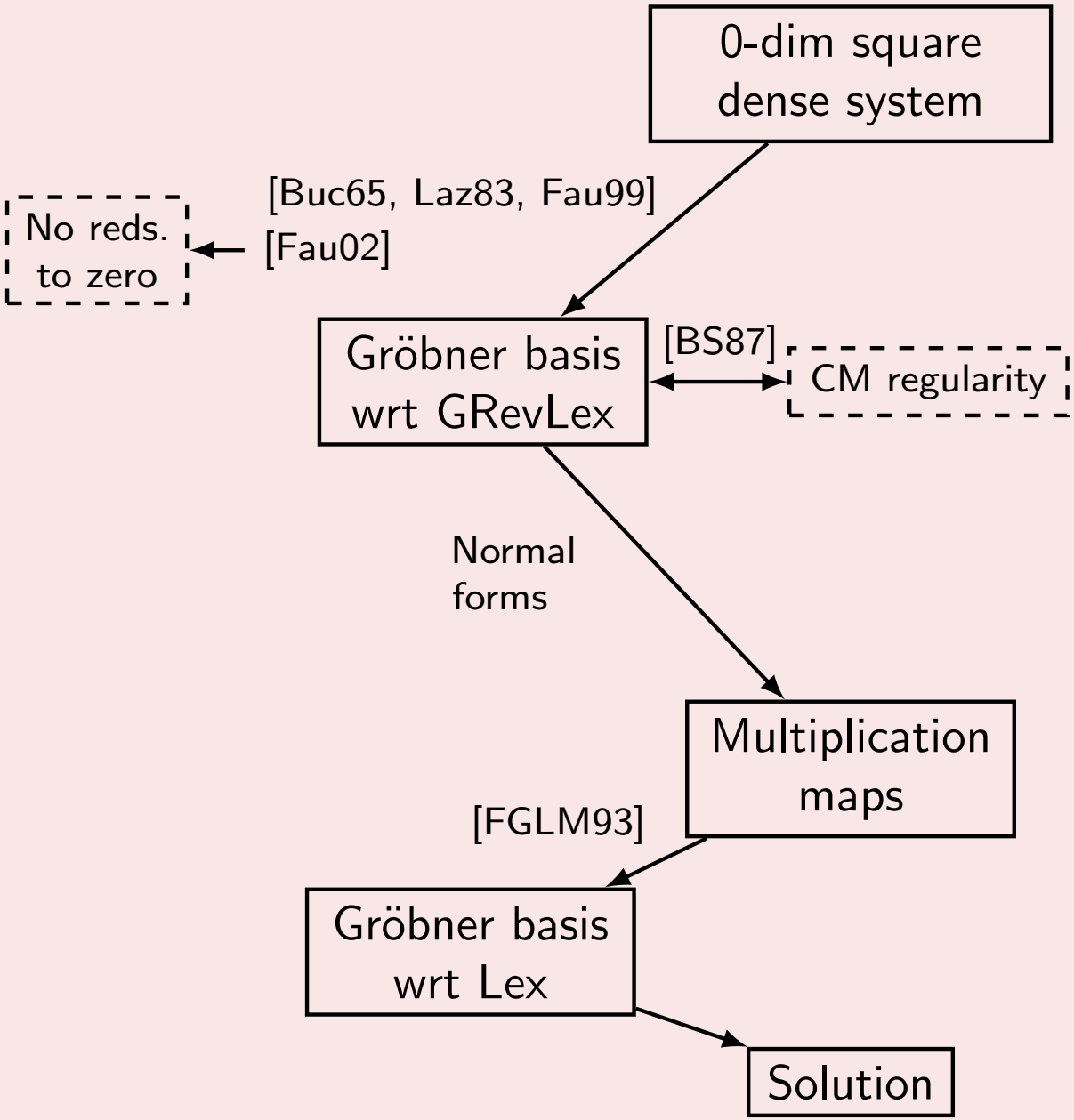
$\rightarrow$ Macaulay bound: CM reg. $= \sum_{i=1}^{n} \deg(f_i) - n + 1$



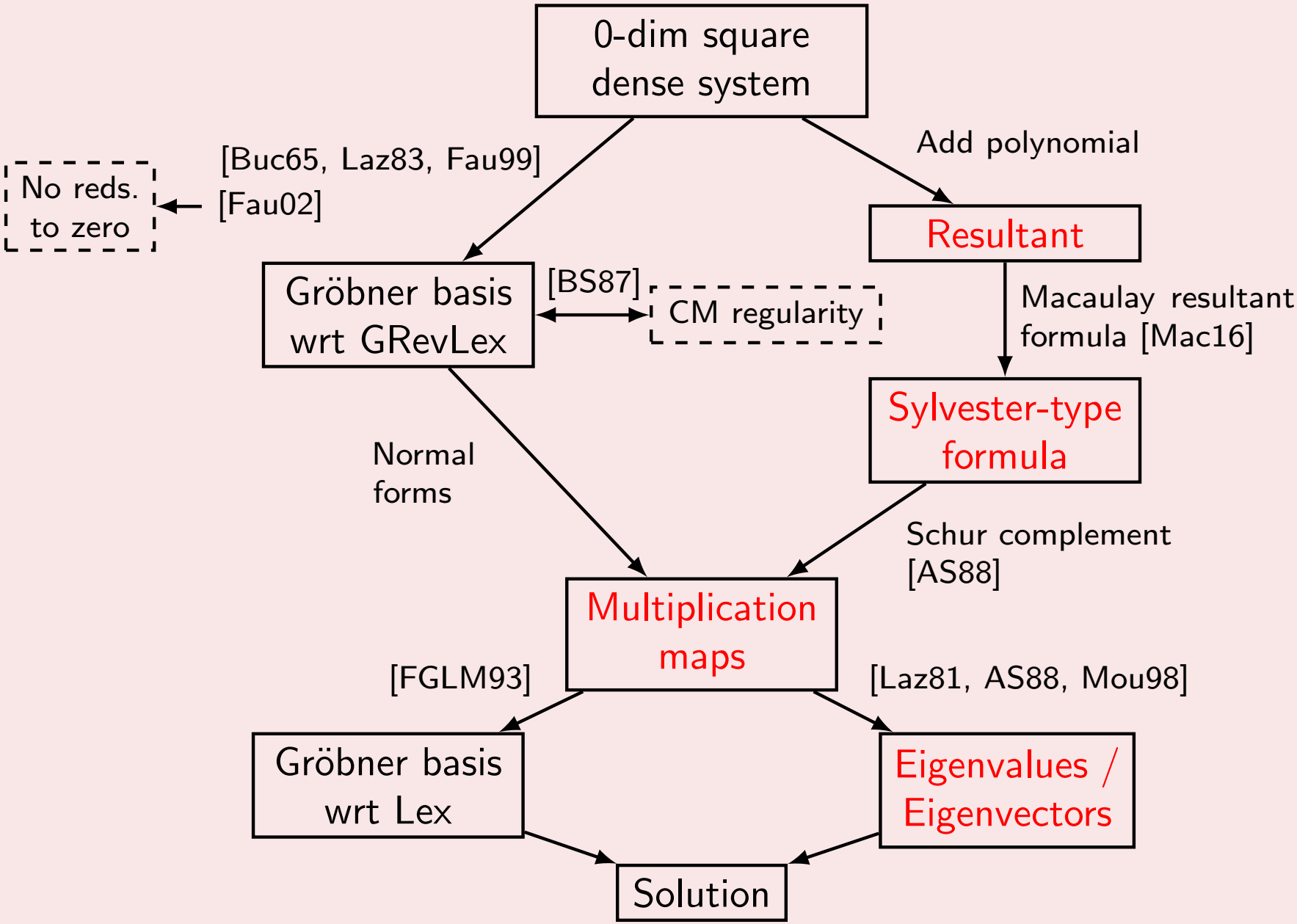
Under suitable assumptions



Under suitable assumptions



Under suitable assumptions



Under suitable assumptions

The resultant and Sylvester-type formulas

Projective resultant

Necessary and sufficient condition for a homogeneous system in $(f_0, \ldots, f_n) \in \mathbb{C}[x_0, \ldots, x_n]^{n+1}$ to have solutions in $\mathbb{P}^n$.

The resultant and Sylvester-type formulas

Projective resultant

Necessary and sufficient condition for a homogeneous system in $(f_0, \ldots, f_n) \in \mathbb{C}[x_0, \ldots, x_n]^{n+1}$ to have solutions in $\mathbb{P}^n$.

Example : Resultant of linear forms = Determinant

$$\begin{cases} a_1\, x + a_2\, y + a_3\, z = 0 \\ b_1\, x + b_2\, y + b_3\, z = 0 \\ c_1\, x + c_2\, y + c_3\, z = 0 \end{cases} \quad \text{has a solution over } \mathbb{P}^2$$

$$\Updownarrow$$

$$\det \begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix} = 0.$$

The resultant and Sylvester-type formulas

Projective resultant

Necessary and sufficient condition for a homogeneous system in $(f_0, \ldots, f_n) \in \mathbb{C}[x_0, \ldots, x_n]^{n+1}$ to have solutions in $\mathbb{P}^n$.

Classical way of computing resultant $\rightarrow$ Sylvester-type formula

$$(g_0, \ldots, g_n) \mapsto \sum_{i=0}^{n} g_i \, f_i$$

The resultant and Sylvester-type formulas

Projective resultant

Necessary and sufficient condition for a homogeneous system in $(f_0, \ldots, f_n) \in \mathbb{C}[x_0, \ldots, x_n]^{n+1}$ to have solutions in $\mathbb{P}^n$.

Classical way of computing resultant $\rightarrow$ Sylvester-type formula

$$(g_0, \ldots, g_n) \mapsto \sum g_i \, f_i$$

Macaulay resultant matrix [Macaulay, 1916]

$$\begin{cases} f_1 := & a_1 x^2 + a_2 x y + a_3 x z + \\ & a_4 y^2 + a_5 y z + a_6 z^2 \\ f_2 := & b_1 x + b_2 y + b_3 z \\ f_3 := & c_1 x + c_2 y + c_3 z \end{cases}$$

	x^2	$x\,y$	$x\,z$	y^2	$y\,z$	z^2
f_1	a_1	a_2	a_3	a_4	a_5	a_6
$x\,f_2$	b_1	b_2	b_3			
$y\,f_2$		b_1		b_2	b_3	
$z\,f_2$			b_1		b_2	b_2
$y\,f_3$		c_1		c_2	c_3	
$z\,f_3$			c_1		c_2	c_3

$$\texttt{Determinant} = \texttt{Resultant} \cdot \texttt{ExtraFactor}.$$

The resultant and Sylvester-type formulas

Projective resultant

Necessary and sufficient condition for a homogeneous system in $(f_0, \ldots, f_n) \in \mathbb{C}[x_0, \ldots, x_n]^{n+1}$ to have solutions in $\mathbb{P}^n$.

Classical way of computing resultant $\rightarrow$ Sylvester-type formula

$$(g_0, \ldots, g_n) \mapsto \sum g_i \, f_i$$

Macaulay resultant matrix [Macaulay, 1916]

$$\begin{cases} f_1 := & a_1 x^2 + a_2 x y + a_3 x z + \\ & a_4 y^2 + a_5 y z + a_6 z^2 \\ f_2 := & b_1 x + b_2 y + b_3 z \\ f_3 := & c_1 x + c_2 y + c_3 z \end{cases}$$

	x^2	$x\,y$	$x\,z$	y^2	$y\,z$	z^2
f_1	a_1	a_2	a_3	a_4	a_5	a_6
$x\,f_2$	b_1	b_2	b_3			
$y\,f_2$		b_1		b_2	b_3	
$z\,f_2$			b_1		b_2	b_2
$y\,f_3$		c_1		c_2	c_3	
$z\,f_3$			c_1		c_2	c_3

$$\texttt{Determinant} = \texttt{Resultant} \cdot \texttt{ExtraFactor}.$$

Determinantal formula $\rightarrow$ `ExtraFactor` is a constant.

Solving via Sylvester-type formulas

- We want to compute the two solutions $\alpha, \beta \in \mathbb{C}^2$ of (f_1, f_2)

$$\begin{cases} f_1 := & 1\,x^2 + -1\,x\,y + 4\,x + -2\,y^2 + -5\,y + 3 \\ f_2 := & 1\,x + -1\,y + -1 \end{cases}.$$

Solving via Sylvester-type formulas

- We want to compute the two solutions $\alpha, \beta \in \mathbb{C}^2$ of (f_1, f_2)

$$\begin{cases} f_1 := & 1\,x^2 + -1\,x\,y + 4\,x + -2\,y^2 + -5\,y + 3 \\ f_2 := & 1\,x + -1\,y + -1 \end{cases}.$$

- Introduce $f_3 := -1\,x + 2\,y + 1$ and consider a Sylvester-type formula.

$$\left(\begin{array}{c|c} M_{1,1} & M_{1,2} \\ \hline M_{2,1} & M_{2,2} \end{array}\right) =$$

	x^2	$x\,y$	x	y^2	y	1
f_1	1	−1	4	−2	−5	3
$x\,f_2$	1	−1	−1			
$y\,f_2$		1		−1	−1	
f_2			1		−1	−1
$y\,f_3$		−1		2	1	
f_3			−1		2	1

Solving via Sylvester-type formulas

- We want to compute the two solutions $\alpha, \beta \in \mathbb{C}^2$ of (f_1, f_2)

$$\begin{cases} f_1 := & 1\,x^2 + -1\,x\,y + 4\,x + -2\,y^2 + -5\,y + 3 \\ f_2 := & 1\,x + -1\,y + -1 \end{cases}.$$

- Introduce $f_3 := -1\,x + 2\,y + 1$ and consider a Sylvester-type formula.

$$\left(\begin{array}{c|c} M_{1,1} & M_{1,2} \\ \hline M_{2,1} & M_{2,2} \end{array}\right) =$$

	x^2	$x\,y$	x	y^2	y	1
f_1	1	−1	4	−2	−5	3
$x\,f_2$	1	−1	−1			
$y\,f_2$		1		−1	−1	
f_2			1		−1	−1
$y\,f_3$		−1		2	1	
f_3			−1		2	1

- Schur complement of $M_{2,2}$ $\leftrightarrow$ Multiplication map of f_3 in $\mathbb{C}[x,y]/\langle f_1, f_2\rangle$

$$\widetilde{M}_{2,2} = M_{2,2} - M_{2,1}\,M_{1,1}^{-1}\,M_{1,2} = \begin{pmatrix} 0 & 4 \\ 1 & 0 \end{pmatrix}$$

Solving via Sylvester-type formulas

- We want to compute the two solutions $\alpha, \beta \in \mathbb{C}^2$ of (f_1, f_2)

$$\begin{cases} f_1 := & 1\,x^2 + -1\,x\,y + 4\,x + -2\,y^2 + -5\,y + 3 \\ f_2 := & 1\,x + -1\,y + -1 \end{cases}.$$

- Introduce $f_3 := -1\,x + 2\,y + 1$ and consider a Sylvester-type formula.

$$\left(\begin{array}{c|c} M_{1,1} & M_{1,2} \\ \hline M_{2,1} & M_{2,2} \end{array}\right) = \begin{array}{r} f_1 \\ x\,f_2 \\ y\,f_2 \\ f_2 \\ \hline y\,f_3 \\ f_3 \end{array} \left(\begin{array}{cccc|cc} 1 & -1 & 4 & -2 & -5 & 3 \\ 1 & -1 & -1 & & & \\ & 1 & & -1 & -1 & \\ & & 1 & & -1 & -1 \\ \hline & -1 & & 2 & 1 & \\ & & -1 & & 2 & 1 \end{array}\right)$$

- Schur complement of $M_{2,2} \leftrightarrow$ Multiplication map of f_3 in $\mathbb{C}[x,y]/\langle f_1, f_2\rangle$

$$\widetilde{M}_{2,2} = M_{2,2} - M_{2,1}\,M_{1,1}^{-1}\,M_{1,2} = \begin{pmatrix} 0 & 4 \\ 1 & 0 \end{pmatrix}$$

Solving via Sylvester-type formulas

- We want to compute the two solutions $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{C}^2$ of (f_1, f_2)

$$\begin{cases} f_1 := & \mathbf{1}\,x^2 + \mathbf{-1}\,x\,y + \mathbf{4}\,x + \mathbf{-2}\,y^2 + \mathbf{-5}\,y + \mathbf{3} \\ f_2 := & \mathbf{1}\,x + \mathbf{-1}\,y + \mathbf{-1} \end{cases}.$$

- Introduce $f_3 := \mathbf{-1}\,x + \mathbf{2}\,y + \mathbf{1}$ and consider a Sylvester-type formula.

$$\left(\begin{array}{c|c} M_{1,1} & M_{1,2} \\ \hline M_{2,1} & M_{2,2} \end{array}\right) = \begin{array}{r} f_1 \\ x\,f_2 \\ y\,f_2 \\ f_2 \\ \hline y\,f_3 \\ f_3 \end{array} \left(\begin{array}{cccc|cc} \mathbf{1} & \mathbf{-1} & \mathbf{4} & \mathbf{-2} & \mathbf{-5} & \mathbf{3} \\ \mathbf{1} & \mathbf{-1} & \mathbf{-1} & & & \\ & \mathbf{1} & & \mathbf{-1} & \mathbf{-1} & \\ & & \mathbf{1} & & \mathbf{-1} & \mathbf{-1} \\ \hline & \mathbf{-1} & & \mathbf{2} & \mathbf{1} & \\ & & \mathbf{-1} & & \mathbf{2} & \mathbf{1} \end{array}\right)$$

- Schur complement of $M_{2,2}$ $\leftrightarrow$ Multiplication map of f_3 in $\mathbb{C}[x,y]/\langle f_1, f_2\rangle$

$$\widetilde{M}_{2,2} = M_{2,2} - M_{2,1}\,M_{1,1}^{-1}\,M_{1,2} = \begin{pmatrix} 0 & 4 \\ 1 & 0 \end{pmatrix}$$

- Eigenvalues of $\widetilde{M}_{2,2}$ [Lazard, 1981]

$$f_3(\boldsymbol{\alpha}) = 2 \qquad \text{and} \qquad f_3(\boldsymbol{\beta}) = -2.$$

Solving via Sylvester-type formulas

- We want to compute the two solutions $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{C}^2$ of (f_1, f_2)

$$\begin{cases} f_1 := & \mathbf{1}\, x^2 + \mathbf{-1}\, x\, y + \mathbf{4}\, x + \mathbf{-2}\, y^2 + \mathbf{-5}\, y + \mathbf{3} \\ f_2 := & \mathbf{1}\, x + \mathbf{-1}\, y + \mathbf{-1} \end{cases}.$$

- Introduce $f_3 := \mathbf{-1}\, x + \mathbf{2}\, y + \mathbf{1}$ and consider a Sylvester-type formula.

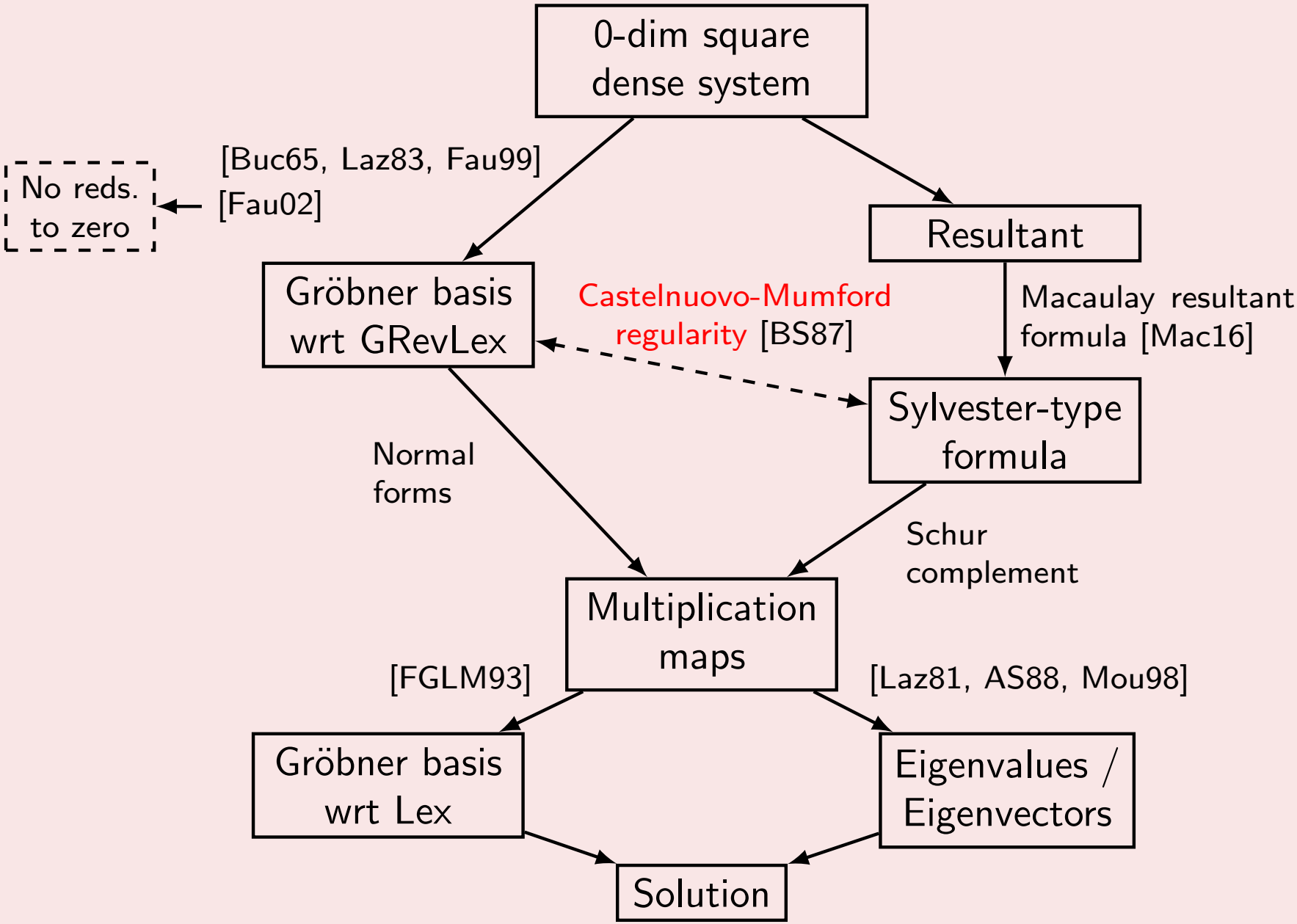
$$\left(\begin{array}{c|c} M_{1,1} & M_{1,2} \\ \hline M_{2,1} & M_{2,2} \end{array}\right) = \begin{array}{c} f_1 \\ x\, f_2 \\ y\, f_2 \\ f_2 \\ \hline y\, f_3 \\ f_3 \end{array} \left(\begin{array}{cccc|cc} \mathbf{1} & \mathbf{-1} & \mathbf{4} & \mathbf{-2} & \mathbf{-5} & \mathbf{3} \\ \mathbf{1} & \mathbf{-1} & \mathbf{-1} & & & \\ & \mathbf{1} & & \mathbf{-1} & \mathbf{-1} & \\ & & \mathbf{1} & & \mathbf{-1} & \mathbf{-1} \\ \hline & \mathbf{-1} & & \mathbf{2} & \mathbf{1} & \\ & & \mathbf{-1} & & \mathbf{2} & \mathbf{1} \end{array}\right)$$

- Schur complement of $M_{2,2} \leftrightarrow$ Multiplication map of f_3 in $\mathbb{C}[x, y]/\langle f_1, f_2 \rangle$

$$\widetilde{M}_{2,2} = M_{2,2} - M_{2,1}\, M_{1,1}^{-1}\, M_{1,2} = \begin{pmatrix} 0 & 4 \\ 1 & 0 \end{pmatrix}$$

- Eigenvalues of $\widetilde{M}_{2,2} \leftrightarrow f_3(\boldsymbol{\alpha}) = 2$ and $f_3(\boldsymbol{\beta}) = -2$. [Lazard, 1981]
- Eigenvectors of $\widetilde{M}_{2,2}$ [Auzinger & Stetter, 1988]

$$\begin{pmatrix} \alpha_y \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \beta_y \\ 1 \end{pmatrix} = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$$

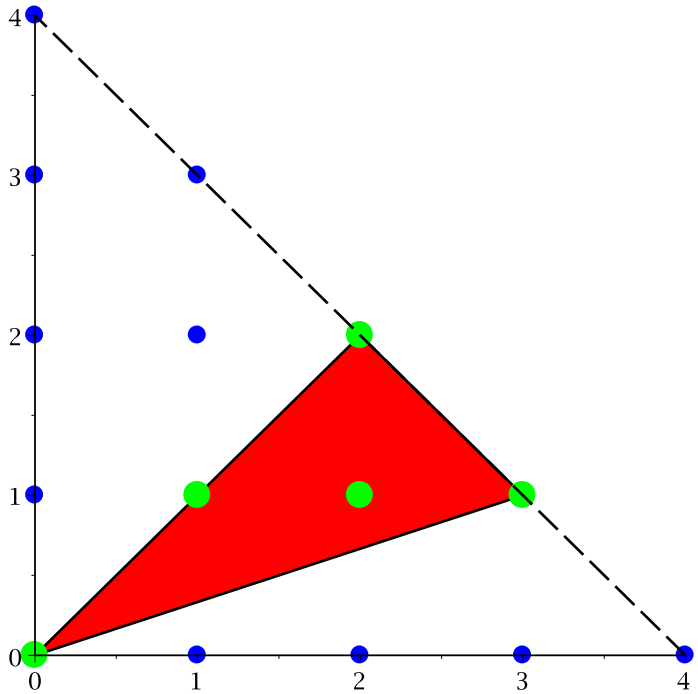

0-dim square dense system
[Buc65, Laz83, Fau99]
No reds. to zero
[Fau02]
Resultant
Gröbner basis wrt GRevLex
Castelnuovo-Mumford regularity [BS87]
Macaulay resultant formula [Mac16]
Sylvester-type formula
Normal forms
Schur complement
Multiplication maps
[FGLM93]
[Laz81, AS88, Mou98]
Gröbner basis wrt Lex
Eigenvalues / Eigenvectors
Solution
Under suitable assumptions

Sparse polynomials / systems

Sparse polynomials / systems

- **Newton polytope** of $f = \sum_\alpha c_\alpha \boldsymbol{x}^\alpha \longrightarrow$ Convex hull of $\{\alpha : c_\alpha \neq 0\}$.
- Sparse polynomial $\longrightarrow$ Its Newton polytope is "small".

$$1 + xy + x^2y + x^2y^2 + x^3y = \mathbf{1} + \mathbf{0} \cdot x + \mathbf{0} \cdot y + \mathbf{0} \cdot x^2 + \boldsymbol{xy} + \mathbf{0} \cdot y^2 + \mathbf{0} \cdot x^3 + \boldsymbol{x^2y} + \mathbf{0} \cdot xy^2 + \mathbf{0} \cdot y^3 + \mathbf{0} \cdot x^4 + \boldsymbol{x^3y} + \boldsymbol{x^2y^2} + \mathbf{0} \cdot xy^3 + \mathbf{0} \cdot y^4$$

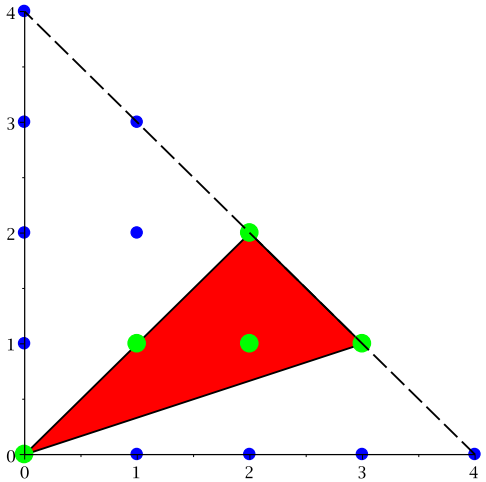


Sparse polynomials / systems

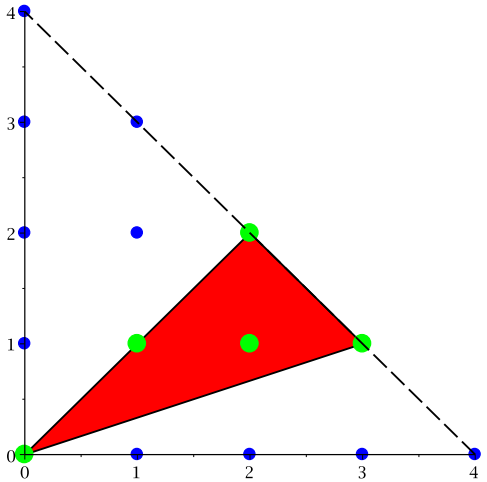
- **Newton polytope** of $f = \sum_\alpha c_\alpha \boldsymbol{x}^\alpha \longrightarrow$ Convex hull of $\{\alpha : c_\alpha \neq 0\}$.
- Sparse polynomial $\longrightarrow$ Its Newton polytope is "small".

- Unmixed sparse system $\longrightarrow$ Polynomials with equal Newton polytope.

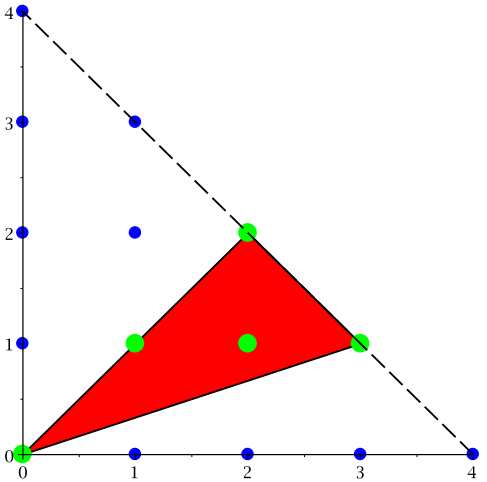
$f_1 := 1 + xy + x^2y + x^2y^2 + x^3y$



$f_2 := 3 - xy + x^2y - x^2y^2 + x^3y$



$f_3 := 2 + xy - x^2y + x^2y^2 + x^3y$

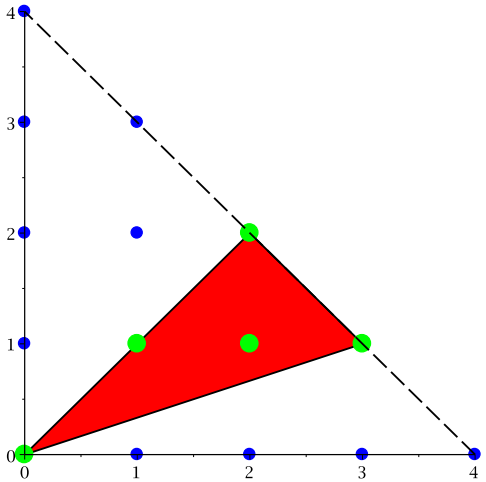


Sparse polynomials / systems

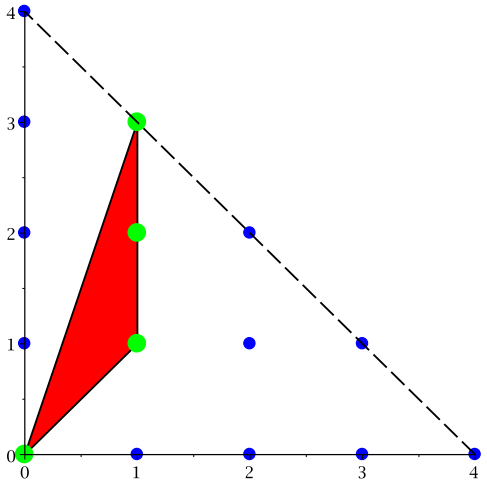
- **Newton polytope** of $f = \sum_\alpha c_\alpha \boldsymbol{x}^\alpha \longrightarrow$ Convex hull of $\{\alpha : c_\alpha \neq 0\}$.
- Sparse polynomial $\longrightarrow$ Its Newton polytope is "small".

- Unmixed sparse system $\longrightarrow$ Polynomials with equal Newton polytope.
- **Mixed sparse system** $\longrightarrow$ Different Newton polytope.

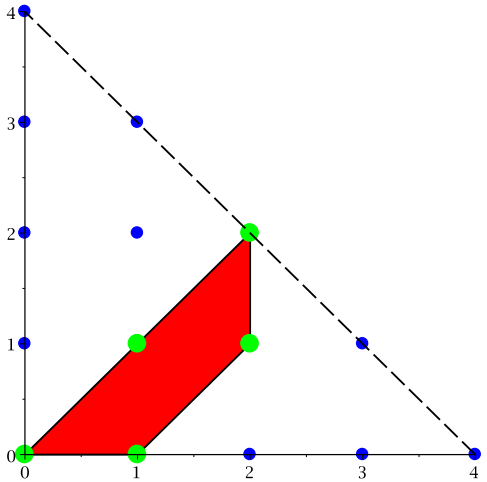
$f_1 := 1 + xy + x^2y + x^2y^2 + x^3y$



$f_2 := 1 + xy + xy^2 + xy^3$



$f_3 := 1 + x + xy + x^2y + x^2y^2$



Exploiting sparsity [Gelfand, Kapranov, & Zelevinsky '90]

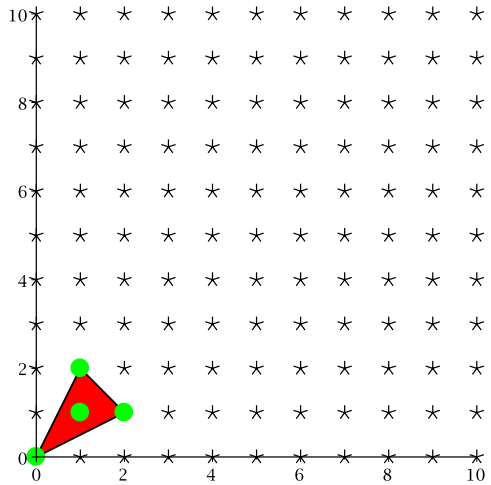
- Sparse resultant $\rightarrow$ Generalization of the resultant for sparse systems.

Exploiting sparsity [Gelfand, Kapranov, & Zelevinsky '90]

- Sparse resultant $\rightarrow$ Generalization of the resultant for sparse systems.
- Geometrically $\rightarrow$ Instead of projective space, use proj. toric variety.

Exploiting sparsity [Gelfand, Kapranov, & Zelevinsky '90]

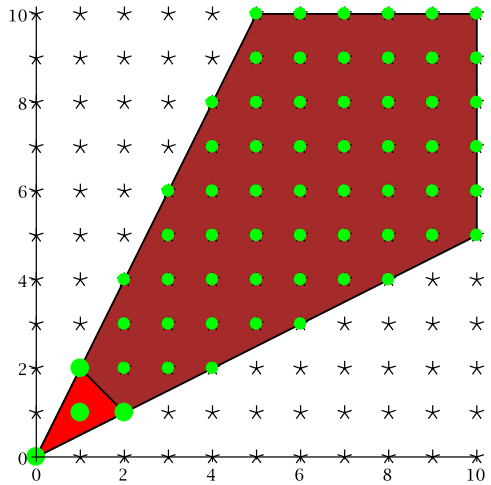
- Sparse resultant $\rightarrow$ Generalization of the resultant for sparse systems.
- Geometrically $\rightarrow$ Instead of projective space, use proj. toric variety.
- Algebraically $\rightarrow$ Replace classical algebra for a semigroup subalgebra.



$$1 + x\,y + x^2\,y + x\,y^2 \in \qquad \mathbb{C}[x, y]$$

Exploiting sparsity [Gelfand, Kapranov, & Zelevinsky '90]

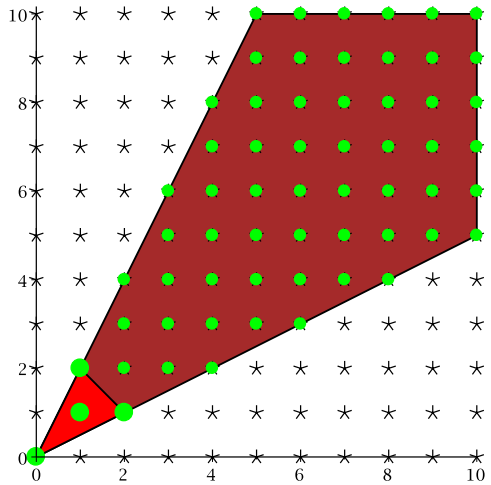
- Sparse resultant $\rightarrow$ Generalization of the resultant for sparse systems.
- Geometrically $\rightarrow$ Instead of projective space, use proj. toric variety.
- Algebraically $\rightarrow$ Replace classical algebra for a semigroup subalgebra.



$$1 + x\,y + x^2\,y + x\,y^2 \in \mathbb{C}[x\,y, x^2\,y, x\,y^2] \subset \mathbb{C}[x, y]$$

Exploiting sparsity [Gelfand, Kapranov, & Zelevinsky '90]

- Sparse resultant $\rightarrow$ Generalization of the resultant for sparse systems.
- Geometrically $\rightarrow$ Instead of projective space, use proj. toric variety.
- Algebraically $\rightarrow$ Replace classical algebra for a semigroup subalgebra.

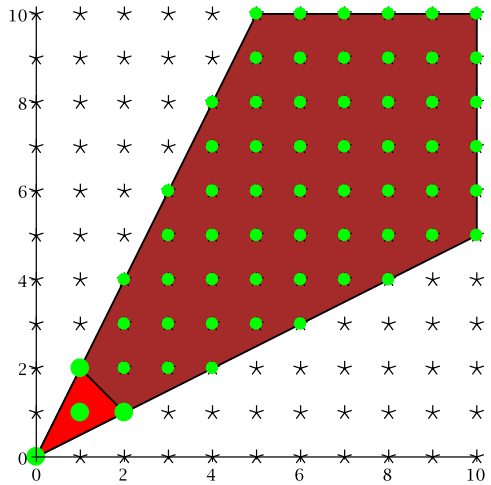


$$1 + x\,y + x^2\,y + x\,y^2 \in \mathbb{C}[x\,y, x^2\,y, x\,y^2] \subset \mathbb{C}[x, y]$$

- Important cases:

Exploiting sparsity [Gelfand, Kapranov, & Zelevinsky '90]

- Sparse resultant $\rightarrow$ Generalization of the resultant for sparse systems.
- Geometrically $\rightarrow$ Instead of projective space, use proj. toric variety.
- Algebraically $\rightarrow$ Replace classical algebra for a semigroup subalgebra.

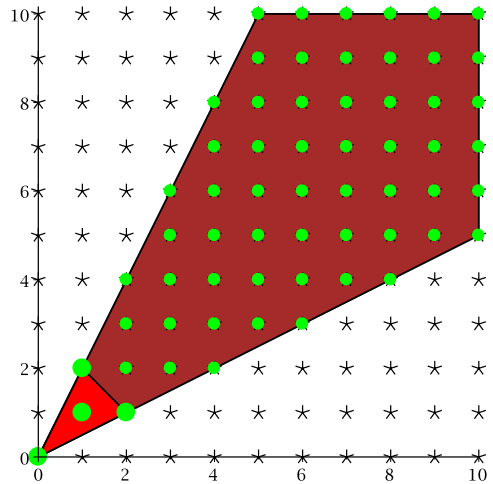


$$1 + x\,y + x^2\,y + x\,y^2 \in \mathbb{C}[x\,y, x^2\,y, x\,y^2] \subset \mathbb{C}[x, y]$$

- Important cases:
  - **Multiprojective space** $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r}$.
    $\rightarrow$ Multihomogeneous polynomials $f_i \in \mathbb{C}[\boldsymbol{x_1}]_{d_{i,1}} \otimes \cdots \otimes \mathbb{C}[\boldsymbol{x_r}]_{d_{i,r}}$.

Exploiting sparsity [Gelfand, Kapranov, & Zelevinsky '90]

- Sparse resultant $\rightarrow$ Generalization of the resultant for sparse systems.
- Geometrically $\rightarrow$ Instead of projective space, use proj. toric variety.
- Algebraically $\rightarrow$ Replace classical algebra for a semigroup subalgebra.



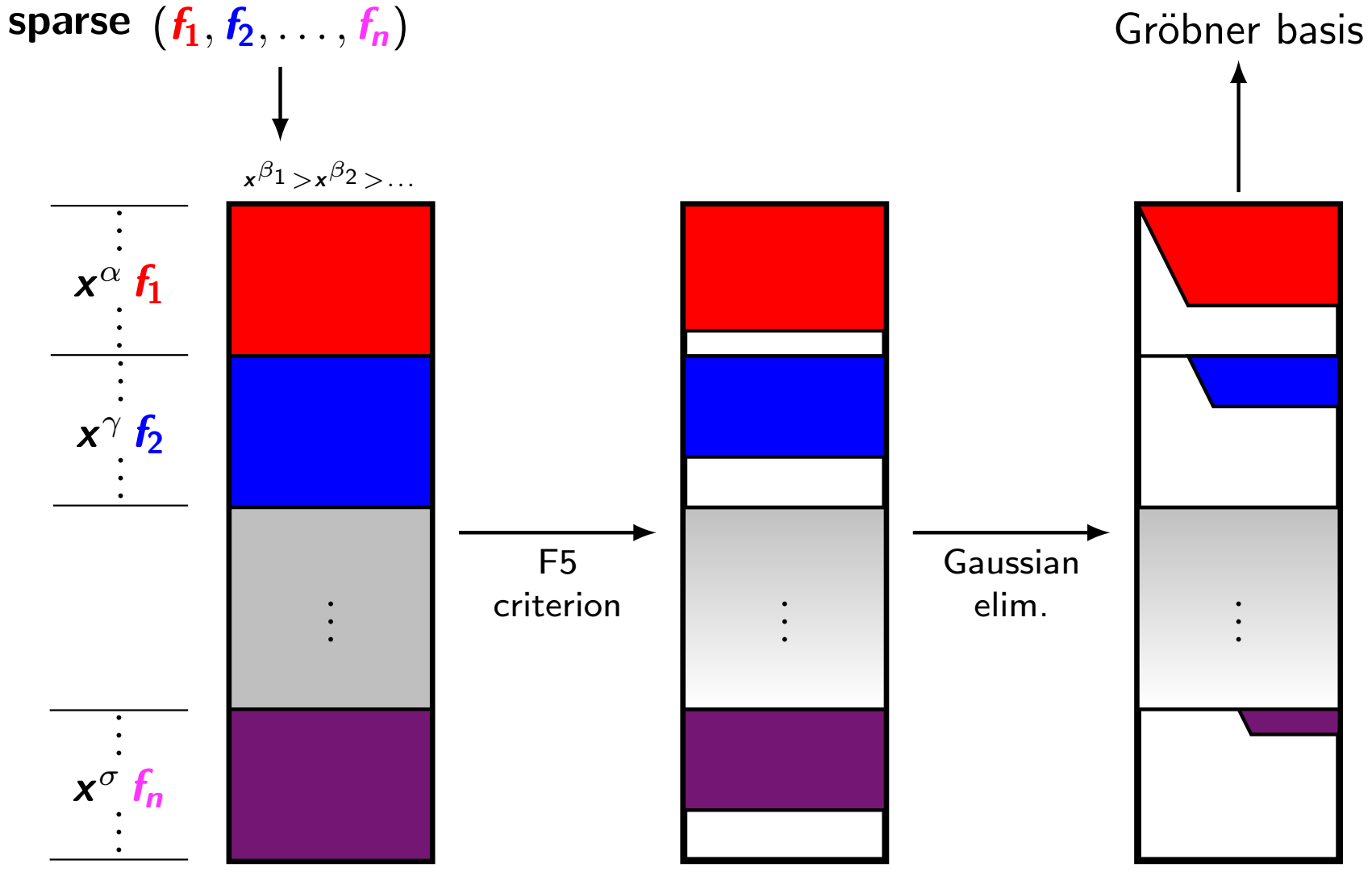
$$1 + x\,y + x^2\,y + x\,y^2 \in \mathbb{C}[x\,y, x^2\,y, x\,y^2] \subset \mathbb{C}[x, y]$$

- Important cases:
  - **Multiprojective space** $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r}$.
    $\rightarrow$ Multihomogeneous polynomials $f_i \in \mathbb{C}[\boldsymbol{x_1}]_{d_{i,1}} \otimes \cdots \otimes \mathbb{C}[\boldsymbol{x_r}]_{d_{i,r}}$.
  - **Weighted projective space** $\mathbb{P}(w_0, \ldots, w_n)$.
    $\rightarrow$ Weighted homogeneous polynomials $\deg(x_i) = w_i$.

Gröbner bases for sparse systems

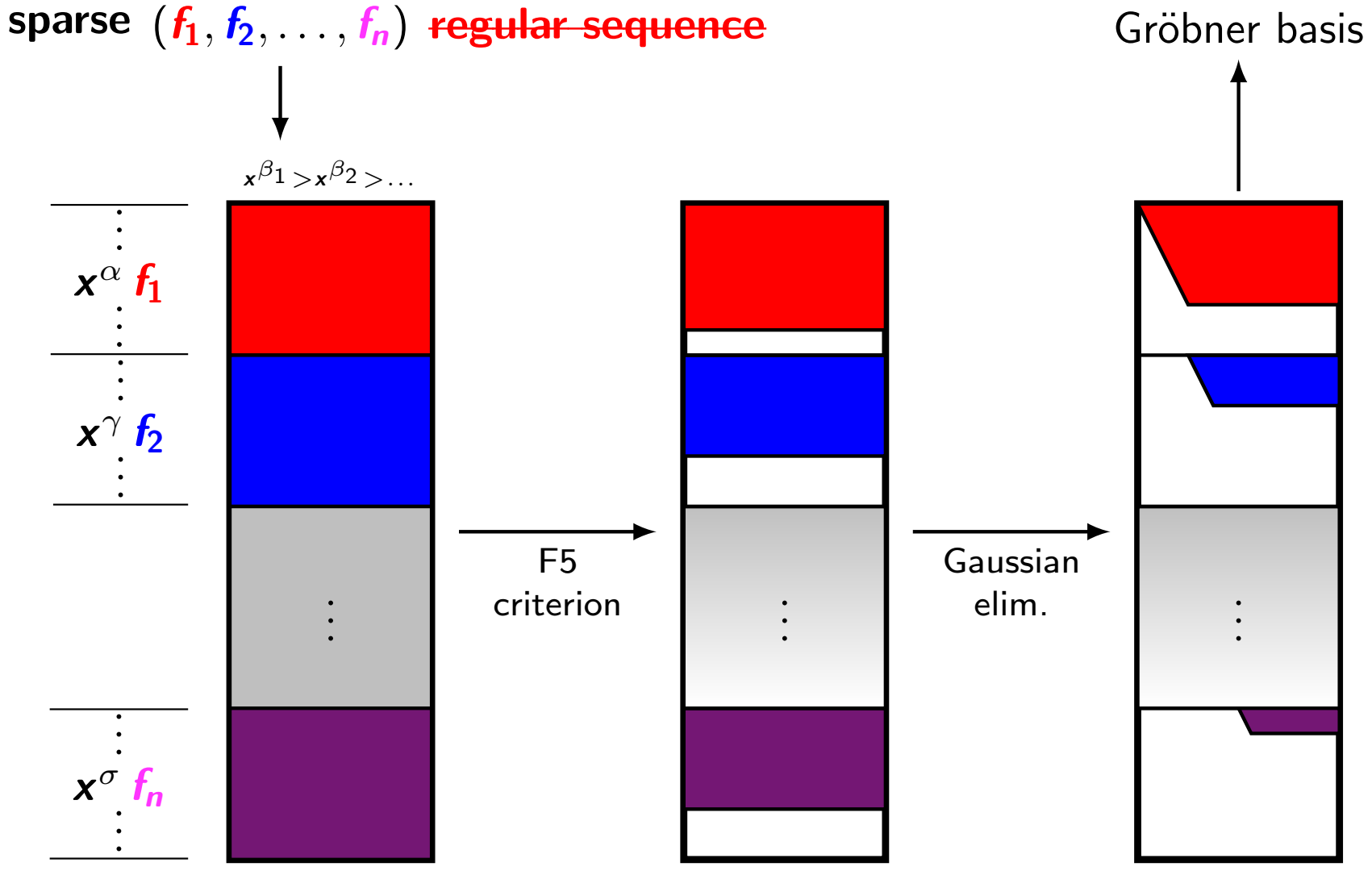
Joint work with Jean-Charles Faugère & Elias Tsigaridas.

Computing Gröbner bases for sparse systems



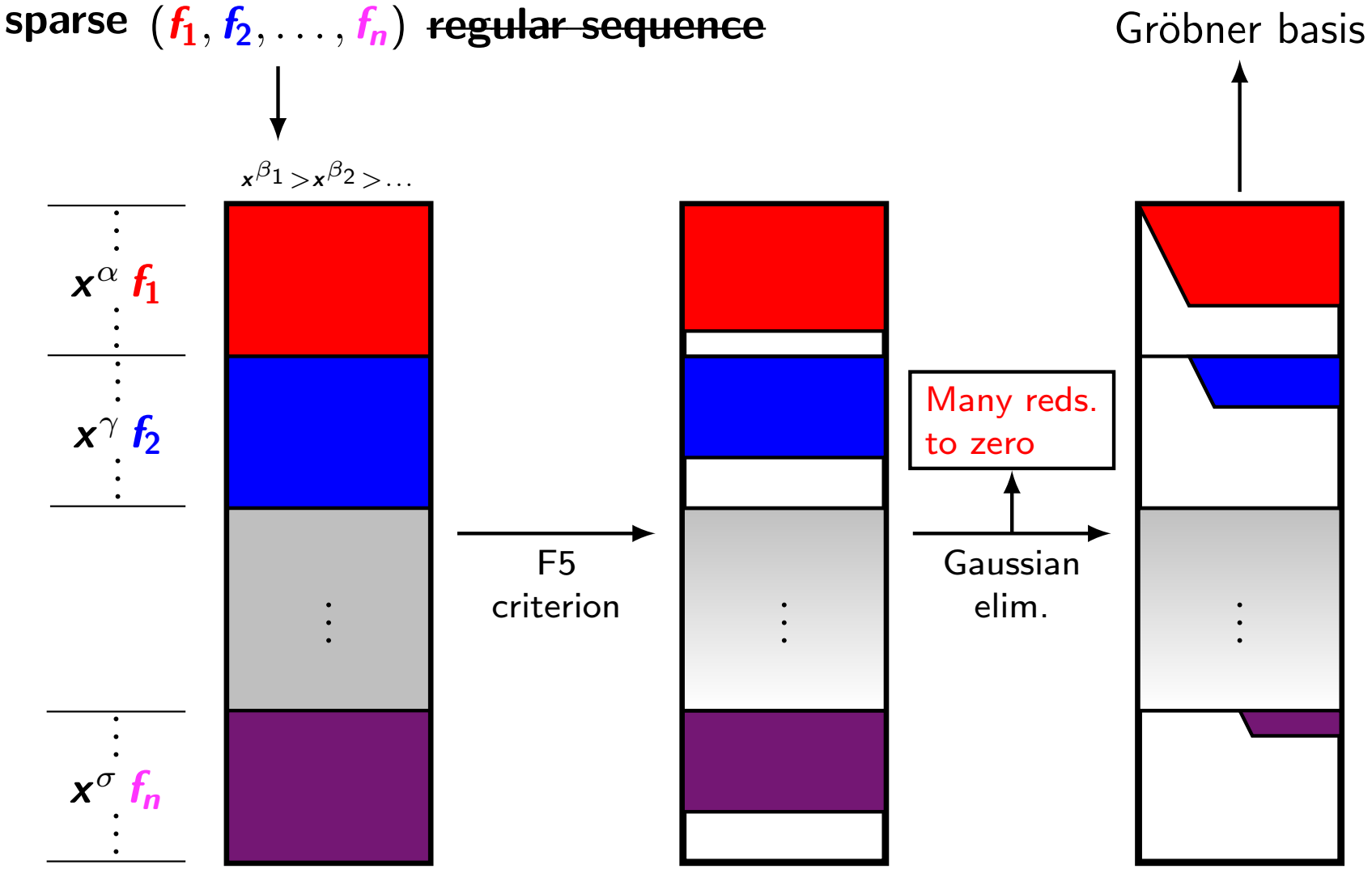
Complexity $\leftrightarrow$ Size of matrix

Computing Gröbner bases for sparse systems



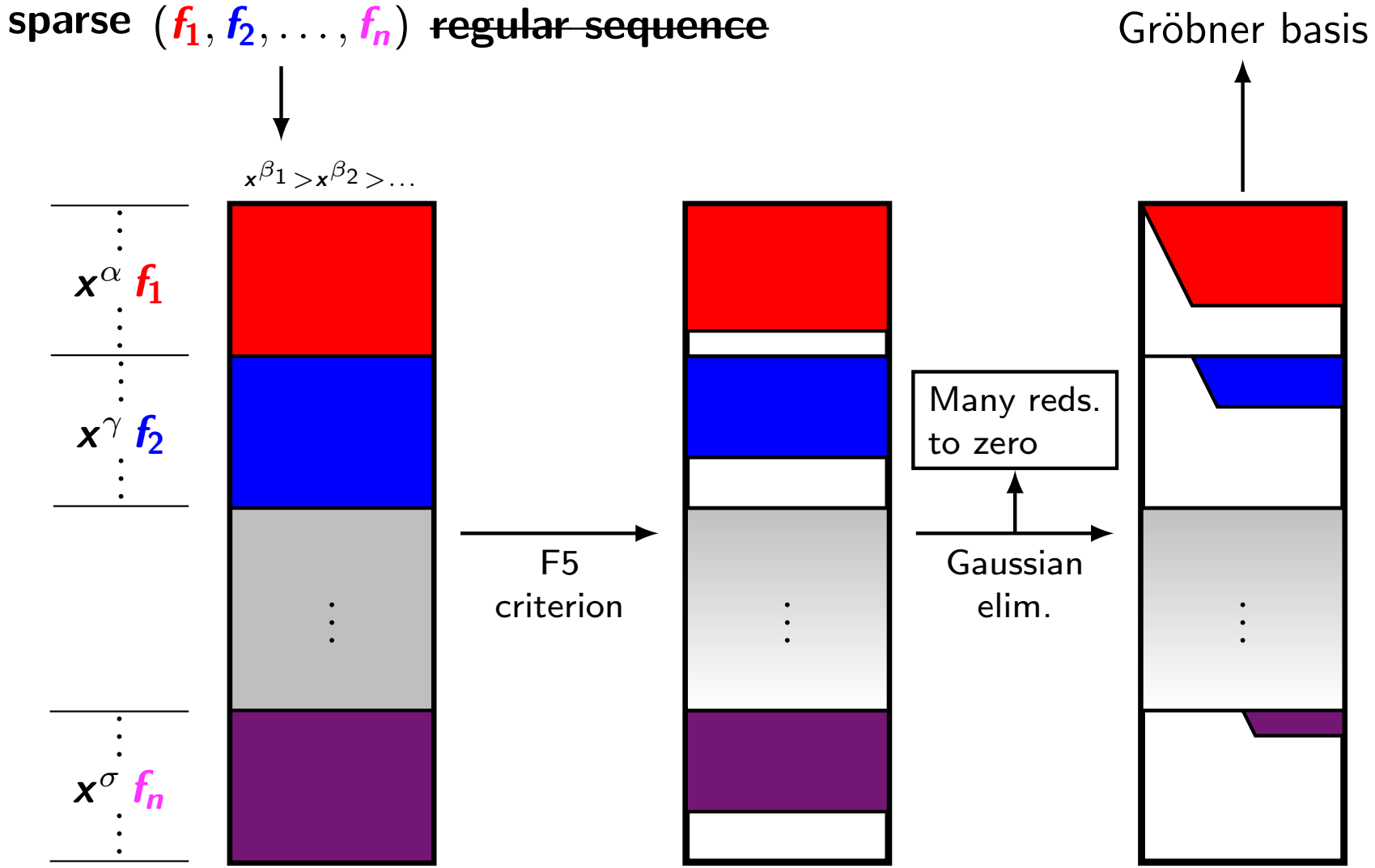
Complexity $\leftrightarrow$ Size of matrix

Computing Gröbner bases for sparse systems



Complexity $\leftrightarrow$ Size of matrix

Computing Gröbner bases for sparse systems



Complexity $\leftrightarrow$ Size of matrix $\rightarrow$ No Macaulay bound for d

Computing Gröbner bases for sparse systems

Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Computing Gröbner bases for sparse systems

Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.

Computing Gröbner bases for sparse systems

Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.

For $n = 2$, consider semigroup algebra $\mathbb{C}[S]$ multigraded by $\mathbb{N}^2$.

$$X^{\left(\alpha,(d_1,d_2)\right)} \in \mathbb{C}[S]_{(d_1,d_2)} \longleftrightarrow \alpha \in (d_1 P_1 + d_2 P_2) \cap \mathbb{Z}^n$$

Computing Gröbner bases for sparse systems

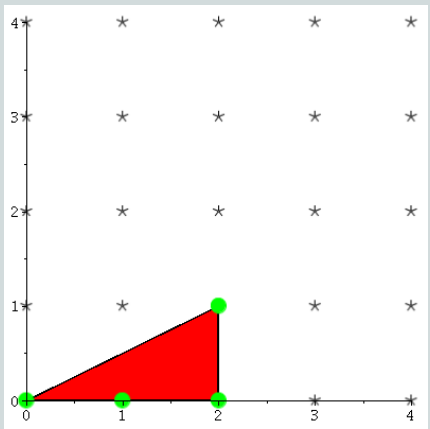
Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.

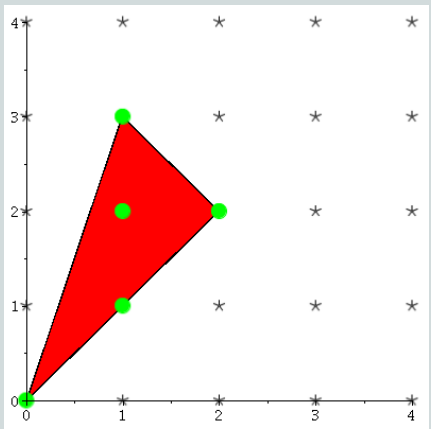
For $n = 2$, consider semigroup algebra $\mathbb{C}[S]$ multigraded by $\mathbb{N}^2$.

$$x^{(\alpha,(d_1,d_2))} \in \mathbb{C}[S]_{(d_1,d_2)} \longleftrightarrow \alpha \in (d_1 P_1 + d_2 P_2) \cap \mathbb{Z}^n$$



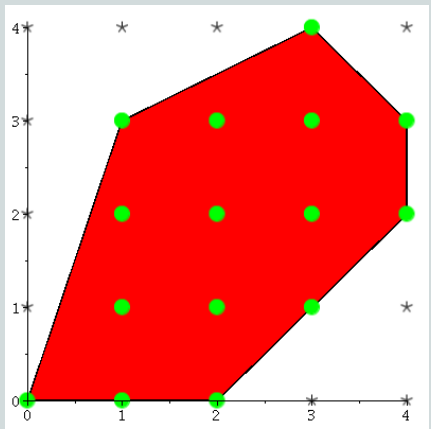
P_1

$+$



P_2

$=$



$P_1 + P_2$

Computing Gröbner bases for sparse systems

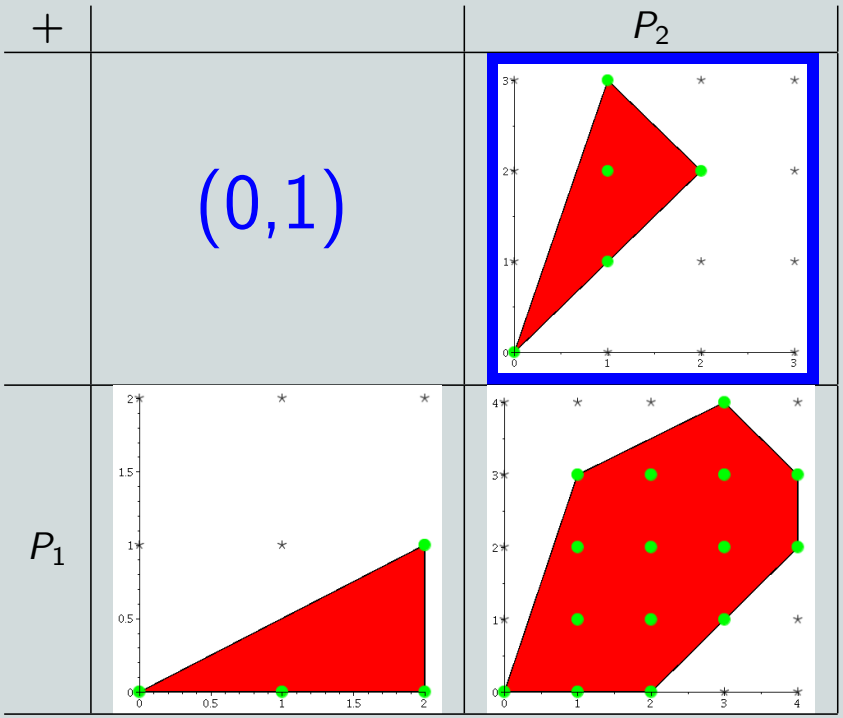
Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.

For $n = 2$, consider semigroup algebra $\mathbb{C}[S]$ multigraded by $\mathbb{N}^2$.

$$x^{(\alpha,(d_1,d_2))} \in \mathbb{C}[S]_{(d_1,d_2)} \longleftrightarrow \alpha \in (d_1 P_1 + d_2 P_2) \cap \mathbb{Z}^n$$



Computing Gröbner bases for sparse systems

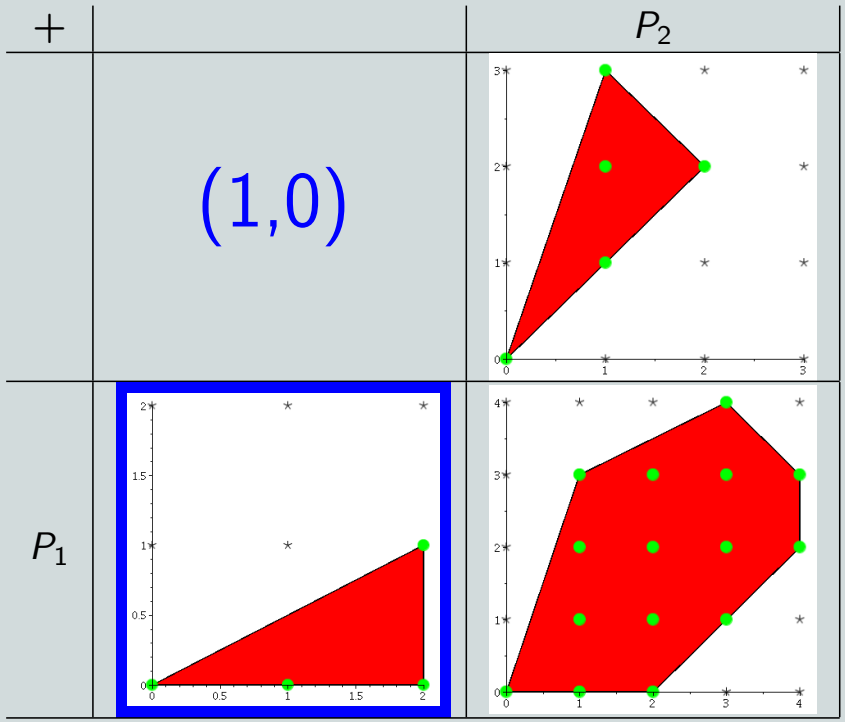
Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.

For $n = 2$, consider semigroup algebra $\mathbb{C}[S]$ multigraded by $\mathbb{N}^2$.

$$x^{(\alpha,(d_1,d_2))} \in \mathbb{C}[S]_{(d_1,d_2)} \longleftrightarrow \alpha \in (d_1 P_1 + d_2 P_2) \cap \mathbb{Z}^n$$



Computing Gröbner bases for sparse systems

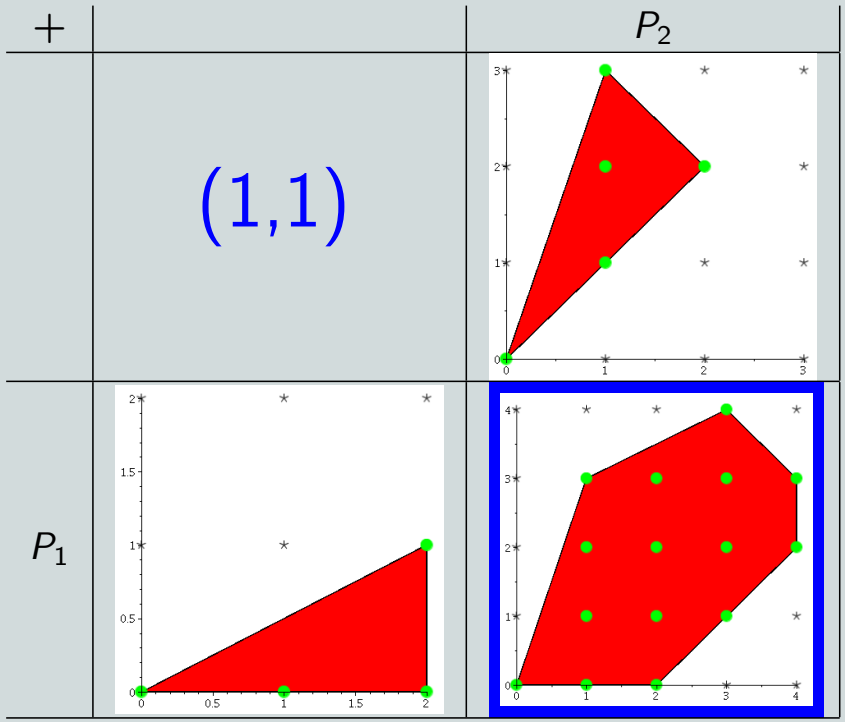
Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.

For $n = 2$, consider semigroup algebra $\mathbb{C}[S]$ multigraded by $\mathbb{N}^2$.

$$x^{(\alpha,(d_1,d_2))} \in \mathbb{C}[S]_{(d_1,d_2)} \longleftrightarrow \alpha \in (d_1 P_1 + d_2 P_2) \cap \mathbb{Z}^n$$



Computing Gröbner bases for sparse systems

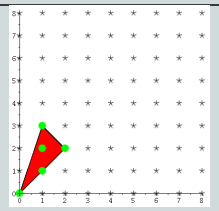
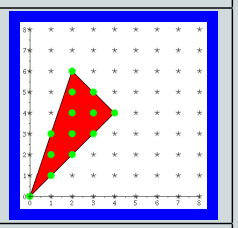
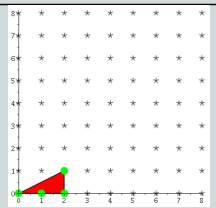
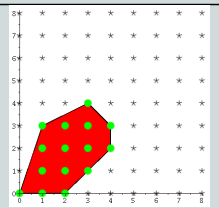
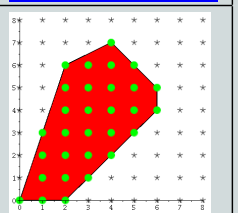
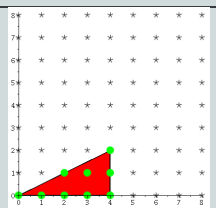
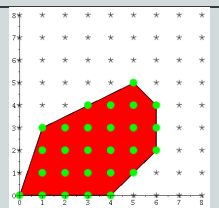
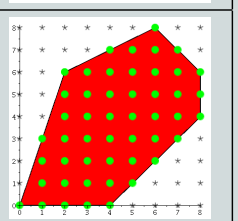
Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.

For $n = 2$, consider semigroup algebra $\mathbb{C}[S]$ multigraded by $\mathbb{N}^2$.

$$x^{(\alpha,(d_1,d_2))} \in \mathbb{C}[S]_{(d_1,d_2)} \longleftrightarrow \alpha \in (d_1P_1 + d_2P_2) \cap \mathbb{Z}^n$$

$+$		P_2	$2P_2$
	(0,2)		
P_1			
$2P_1$			

Computing Gröbner bases for sparse systems

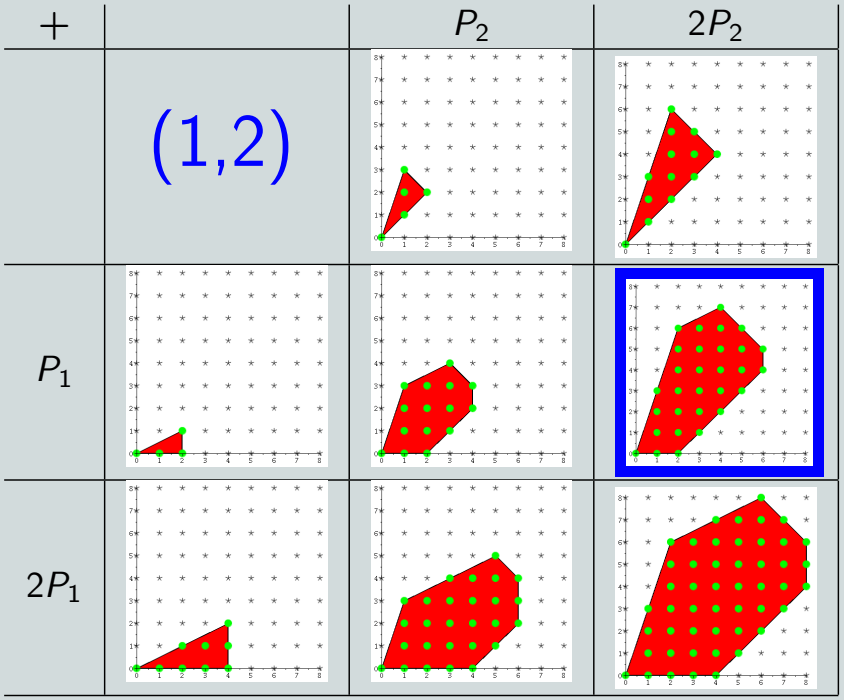
Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.

For $n = 2$, consider semigroup algebra $\mathbb{C}[S]$ multigraded by $\mathbb{N}^2$.

$$x^{(\alpha,(d_1,d_2))} \in \mathbb{C}[S]_{(d_1,d_2)} \longleftrightarrow \alpha \in (d_1 P_1 + d_2 P_2) \cap \mathbb{Z}^n$$



Computing Gröbner bases for sparse systems

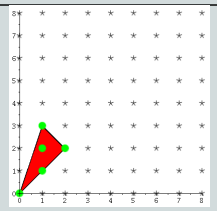
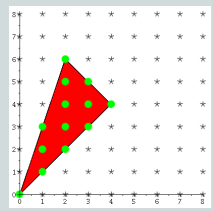
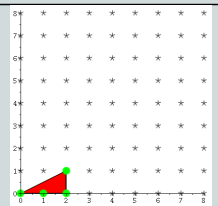
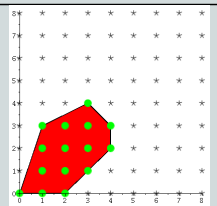
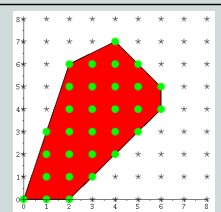
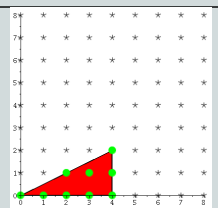
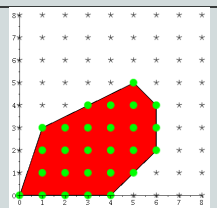
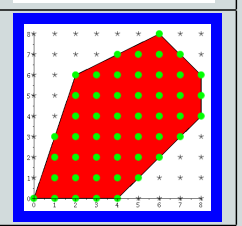
Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.

For $n = 2$, consider semigroup algebra $\mathbb{C}[S]$ multigraded by $\mathbb{N}^2$.

$$x^{(\alpha,(d_1,d_2))} \in \mathbb{C}[S]_{(d_1,d_2)} \longleftrightarrow \alpha \in (d_1P_1 + d_2P_2) \cap \mathbb{Z}^n$$

$+$		P_2	$2P_2$
	(2,2)		
P_1			
$2P_1$			

Computing Gröbner bases for sparse systems

Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.

In unmixed case (all pols same Newton polytope, $P = P_i$),
$\rightarrow$ approach considered in [Faugère, Spaenlehauer, Svartz '14].

Computing Gröbner bases for sparse systems

Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [**B**FT18], [**B**FT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.

Computing Gröbner bases for sparse systems

Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.
- New Koszul-F5 criterion $\rightarrow$ Under mild assumptions, no reds. to zero.

Computing Gröbner bases for sparse systems

Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [**B**FT18], [**B**FT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.
- New Koszul-F5 criterion $\rightarrow$ Under mild assumptions, no reds. to zero.

- Careful $\rightarrow$ polynomials are NOT regular sequence
- But Koszul complex exact at "big enough" degree $\rightarrow$ predict syzygies.

Computing Gröbner bases for sparse systems

Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.
- New Koszul-F5 criterion $\rightarrow$ Under mild assumptions, no reds. to zero.

Computing Gröbner bases for sparse systems

Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.
- New Koszul-F5 criterion $\rightarrow$ Under mild assumptions, no reds. to zero.
- Complexity bounds for square systems $\rightarrow$ Complexity of solving sparse sys.

Computing Gröbner bases for sparse systems

Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [**B**FT18], [**B**FT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.
- New Koszul-F5 criterion $\rightarrow$ Under mild assumptions, no reds. to zero.
- Complexity bounds for square systems $\rightarrow$ Complexity of solving sparse sys.

Biggest matrix has size $\sum_i P_i \cap \mathbb{Z}^n$.

Computing Gröbner bases for sparse systems

Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.
- New Koszul-F5 criterion $\rightarrow$ Under mild assumptions, no reds. to zero.
- Complexity bounds for square systems $\rightarrow$ Complexity of solving sparse sys.

Biggest matrix has size $\sum_i P_i \cap \mathbb{Z}^n$. $\sum \deg(f_i)$

Computing Gröbner bases for sparse systems

Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.
- New Koszul-F5 criterion $\rightarrow$ Under mild assumptions, no reds. to zero.
- Complexity bounds for square systems $\rightarrow$ Complexity of solving sparse sys.

  Biggest matrix has size $\sum_i P_i \cap \mathbb{Z}^n$. $\sum \deg(f_i)$
- Improvements for special cases $\longrightarrow$ Generalization of the Macaulay bound

Computing Gröbner bases for sparse systems

Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.
- New Koszul-F5 criterion $\rightarrow$ Under mild assumptions, no reds. to zero.
- Complexity bounds for square systems $\rightarrow$ Complexity of solving sparse sys.

  Biggest matrix has size $\sum_i P_i \cap \mathbb{Z}^n$. $\sum \deg(f_i)$
- Improvements for special cases $\longrightarrow$ Generalization of the Macaulay bound
  - Multihomogeneous systems $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r}$.

    $\sum \texttt{multideg}(f_i) - (n_1, \ldots, n_r) + (1, \ldots, 1)$

Computing Gröbner bases for sparse systems

Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.
- New Koszul-F5 criterion $\rightarrow$ Under mild assumptions, no reds. to zero.
- Complexity bounds for square systems $\rightarrow$ Complexity of solving sparse sys.

  Biggest matrix has size $\sum_i P_i \cap \mathbb{Z}^n$. $\sum \deg(f_i)$
- Improvements for special cases $\longrightarrow$ Generalization of the Macaulay bound
  - Multihomogeneous systems $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r}$.

    $\sum \texttt{multideg}(f_i) - (n_1, \ldots, n_r) + (1, \ldots, 1)$
  - Unmixed systems (all pols same Newton polytope, $P = P_i$).

    $\sum \deg(f_i) - \texttt{codegree}(P) + 1$

Complexity for *variant* of [Faugère, Spaenlehauer, Svartz '14].

Computing Gröbner bases for sparse systems

Given $f_1, \ldots, f_n \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Our algorithm [BFT18], [BFT19]

- Exploit sparsity $\rightarrow$ compute over multigraded semigroup algebras.
- New Koszul-F5 criterion $\rightarrow$ Under mild assumptions, no reds. to zero.
- Complexity bounds for square systems $\rightarrow$ Complexity of solving sparse sys.

  Biggest matrix has size $\sum_i P_i \cap \mathbb{Z}^n$. $\sum \deg(f_i)$
- Improvements for special cases $\longrightarrow$ Generalization of the Macaulay bound
  - Multihomogeneous systems $\mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_r}$.

    $\sum \texttt{multideg}(f_i) - (n_1, \ldots, n_r) + (1, \ldots, 1)$
  - Unmixed systems (all pols same Newton polytope, $P = P_i$).

    $\sum \deg(f_i) - \texttt{codegree}(P) + 1$
  - Not all variables in all P_i.

Solving numerically sparse systems

Joint work with Simon Telen.

Symbolic-numerical approaches to solve sparse systems

In applications $\rightarrow$ sparse polynomials with noisy coefficients, and we need to approximate solutions.

Symbolic-numerical approaches to solve sparse systems

In applications $\rightarrow$ sparse polynomials with noisy coefficients, and we need to approximate solutions.

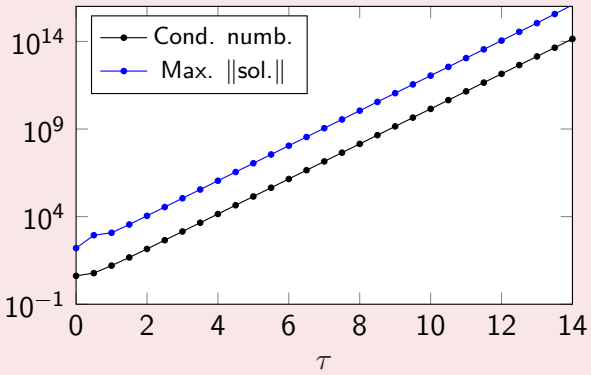
Symbolic-numerical approaches to solve sparse systems

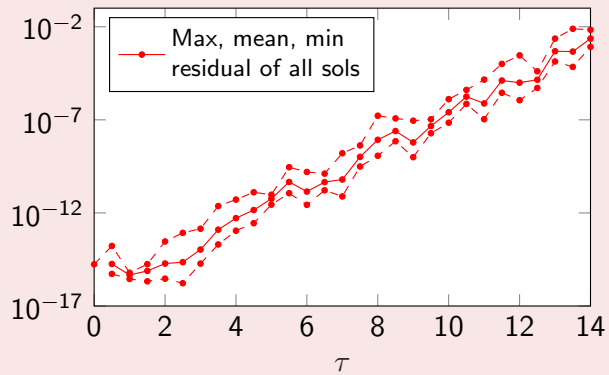
In applications $\rightarrow$ sparse polynomials with noisy coefficients, and we need to approximate solutions.

In degenerate situations (e.g. 'sols. going to ∞') $\rightarrow$ Sparse GBs or resultants $\rightarrow$ large numerical errors (for every solution!).

Example

$$\begin{cases} f_1 := -1 + x + x^2 + y + x\,y, \\ f_2 := -1 + x + (\frac{5}{2} - 10^{-\tau})\,x^2 + 2\,y + \frac{5}{2}\,x\,y. \end{cases}$$





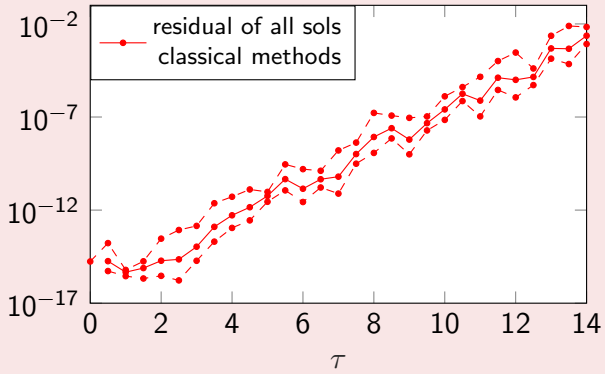
Symbolic-numerical approaches to solve sparse systems

In applications $\rightarrow$ sparse polynomials with noisy coefficients, and we need to approximate solutions.

Example

$$f_1 := 1 - x - x^2 - y - xy,$$

$$f_2 := 1 - x - (5/2 - 10^{-\tau})\,x^2 - 2y - 5/2\,xy$$



Symbolic-numerical approaches to solve sparse systems

In applications $\rightarrow$ sparse polynomials with noisy coefficients, and we need to approximate solutions.

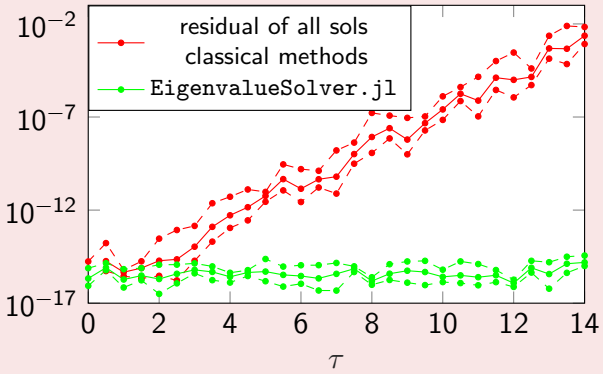
Our algorithm [BT20],[BT21]

- Homogenize polynomials over Cox ring (homogeneous coord. ring of toric variety).

Example

$$f_1 := 1 - x - x^2 - y - xy,$$

$$f_2 := 1 - x - (5/2 - 10^{-\tau})\,x^2 - 2y - 5/2\,xy$$



Symbolic-numerical approaches to solve sparse systems

In applications $\rightarrow$ sparse polynomials with noisy coefficients, and we need to approximate solutions.

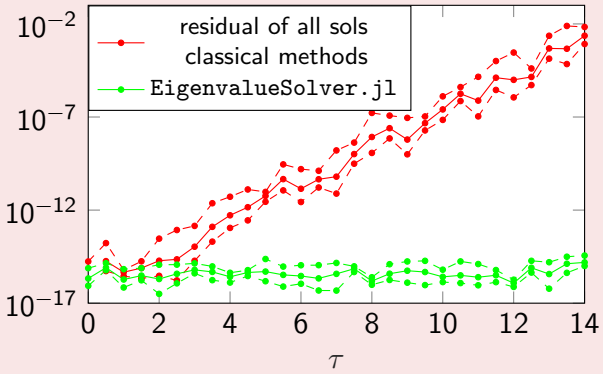
Our algorithm [BT20],[BT21]

- Homogenize polynomials over Cox ring (homogeneous coord. ring of toric variety).
- Consider Sylvester map ($\boldsymbol{g} \mapsto \sum f_i\, g_i$) at "big enough" (known) degree.

Example

$$f_1 := 1 - x - x^2 - y - x\,y,$$

$$f_2 := 1 - x - (5/2 - 10^{-\tau})\,x^2 - 2\,y - 5/2\,x\,y$$



Symbolic-numerical approaches to solve sparse systems

In applications $\rightarrow$ sparse polynomials with noisy coefficients, and we need to approximate solutions.

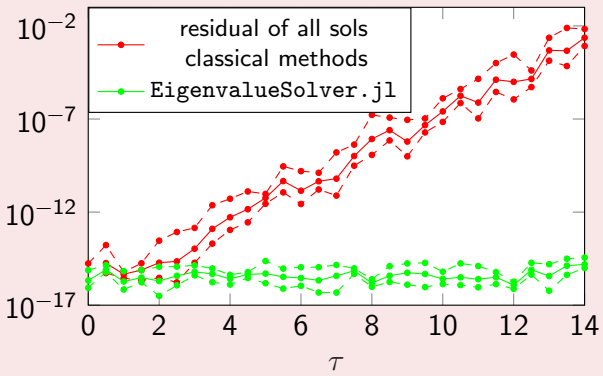
Our algorithm [BT20],[BT21]

- Homogenize polynomials over Cox ring (homogeneous coord. ring of toric variety).
- Consider Sylvester map ($\boldsymbol{g} \mapsto \sum f_i\, g_i$) at "big enough" (known) degree.
- Using numerical linear algebra tools (QR with optimal pivoting, SVD) $\rightarrow$ reduce to eigenvalue comput. (mult. maps).

Example

$$f_1 := 1 - x - x^2 - y - x\,y,$$

$$f_2 := 1 - x - (5/2 - 10^{-\tau})\,x^2 - 2\,y - 5/2\,x\,y$$



Symbolic-numerical approaches to solve sparse systems

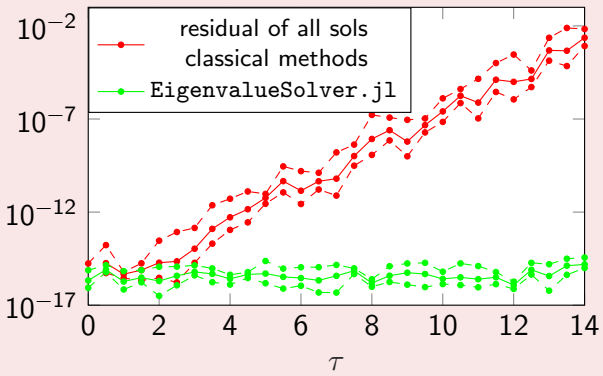
In applications $\rightarrow$ sparse polynomials with noisy coefficients, and we need to approximate solutions.

Our algorithm [BT20],[BT21]

- Homogenize polynomials over Cox ring (homogeneous coord. ring of toric variety).
- Consider Sylvester map ($\boldsymbol{g} \mapsto \sum f_i\, g_i$) at "big enough" (known) degree.
- Using numerical linear algebra tools (QR with optimal pivoting, SVD) $\rightarrow$ reduce to eigenvalue comput. (mult. maps).
- In contrast to homotopy continuation $\rightarrow$ works even better with overdetermined sys.

Example

$$f_1 := 1 - x - x^2 - y - x\,y,$$
$$f_2 := 1 - x - (5/2 - 10^{-\tau})\,x^2 - 2\,y - 5/2\,x\,y$$



Symbolic-numerical approaches to solve sparse systems

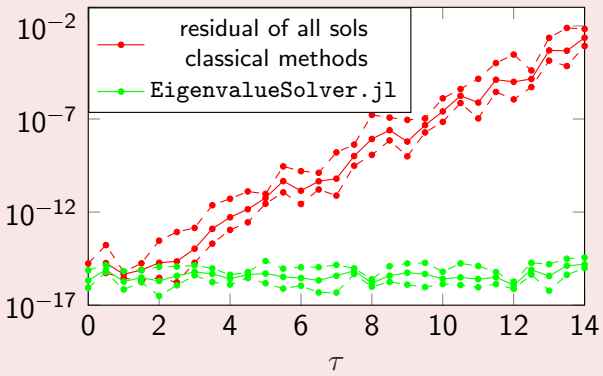
In applications $\rightarrow$ sparse polynomials with noisy coefficients, and we need to approximate solutions.

Our algorithm [BT20],[BT21]

- Homogenize polynomials over Cox ring (homogeneous coord. ring of toric variety).
- Consider Sylvester map ($\boldsymbol{g} \mapsto \sum f_i\, g_i$) at "big enough" (known) degree.
- Using numerical linear algebra tools (QR with optimal pivoting, SVD) $\rightarrow$ reduce to eigenvalue comput. (mult. maps).
- In contrast to homotopy continuation $\rightarrow$ works even better with overdetermined sys.
- Toy implem. in Julia: `EigenvalueSolver.jl`

Example

$$f_1 := 1 - x - x^2 - y - x\,y,$$
$$f_2 := 1 - x - (5/2 - 10^{-\tau})\,x^2 - 2\,y - 5/2\,x\,y$$



How to improve efficiency?

How to improve efficiency?

- Extremely important efficient linear algebra.

How to improve efficiency?

- Extremely important efficient linear algebra.

  $\rightarrow$ However, CM reg. is theoretical limit.

How to improve efficiency?

- Extremely important efficient linear algebra.
  $\rightarrow$ However, CM reg. is theoretical limit.
- Work with smaller matrices:

How to improve efficiency?

- Extremely important efficient linear algebra.
  $\rightarrow$ However, CM reg. is theoretical limit.
- Work with smaller matrices:
  - Homogenize "better" $\rightarrow$ Consider algebras with smaller CM reg.

How to improve efficiency?

- Extremely important efficient linear algebra.
  $\rightarrow$ However, CM reg. is theoretical limit.
- Work with smaller matrices:
  - Homogenize "better" $\rightarrow$ Consider algebras with smaller CM reg.
  - We use a broader class of smaller matrices to solve.

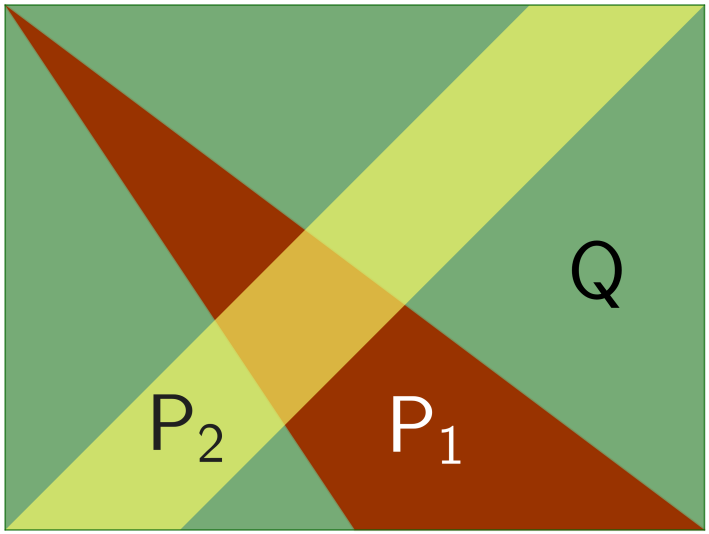
Complete intersections over toric varieties

Joint work with Pierre-Jean Spaenlehauer.

Complete intersections over toric varieties

Given $f_1, \ldots, f_r \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Assume $P_i \subset Q$.



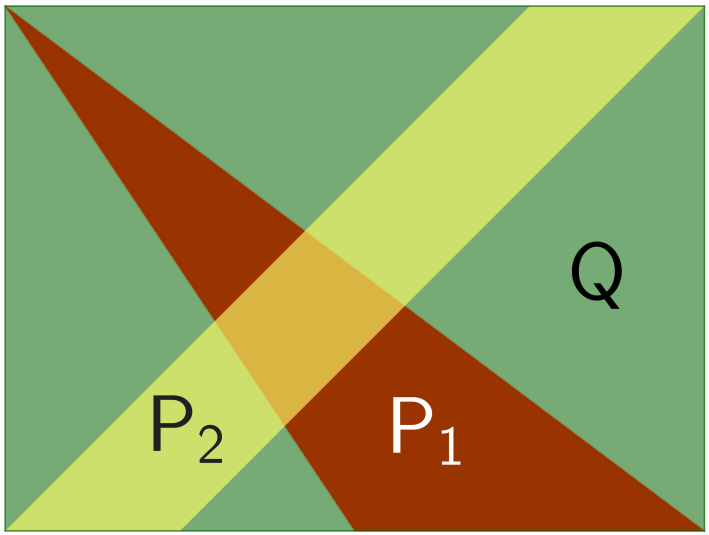
Complete intersections over toric varieties

Given $f_1, \ldots, f_r \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Assume $P_i \subset Q$. The CM regularity bounded by

Mixed case wrt $P_1, \ldots, P_r$

$$\sum \deg(f_i)$$



Complete intersections over toric varieties

Given $f_1, \ldots, f_r \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

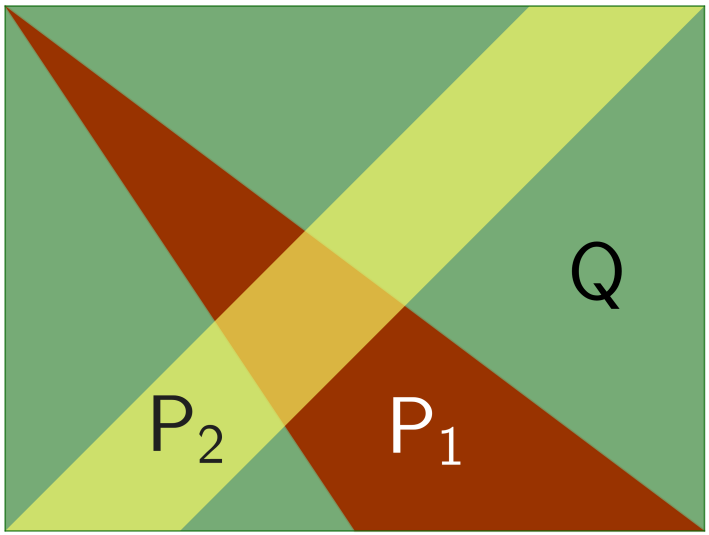
Assume $P_i \subset Q$. The CM regularity bounded by

Mixed case wrt $P_1, \ldots, P_r$

$$\sum \texttt{deg}(f_i)$$

Unmixed case wrt Q (*)

$$\sum \texttt{deg}(f_i) - \texttt{codegree}(Q) + 1$$



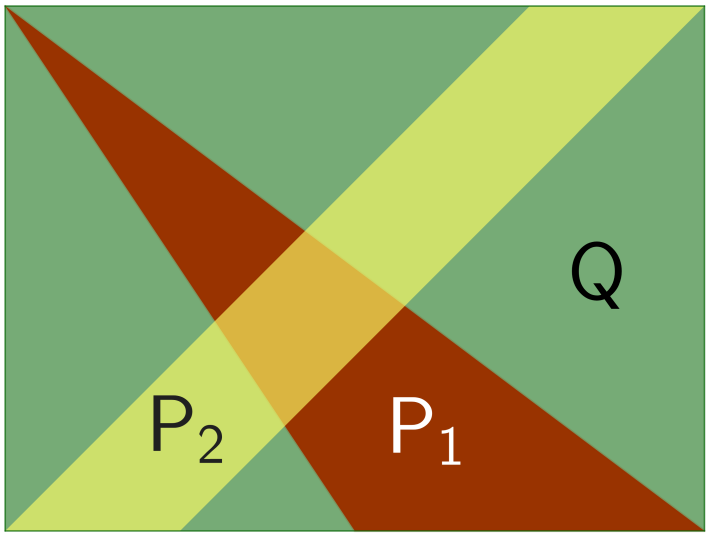
Complete intersections over toric varieties

Given $f_1, \ldots, f_r \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

Assume $P_i \subset Q$. The CM regularity bounded by

Mixed case wrt $P_1, \ldots, P_r$ | Unmixed case wrt Q (*)

$$\sum \deg(f_i) \qquad \sum \deg(f_i) - \texttt{codegree}(Q) + 1$$



(*) Requires regularity assumptions!

$\dim(\text{homogenized}(f_1), \ldots, \text{homogenized}(f_i)) = n - i$, for every i.

Complete intersections over toric varieties

Given $f_1, \ldots, f_r \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

When can homogenize? i.e $\dim(\hom(f_1), \ldots, \hom(f_i)) = n - i$, for every i.

Complete intersections over toric varieties

Given $f_1, \ldots, f_r \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

When can homogenize? i.e $\dim(\hom(f_1), \ldots, \hom(f_i)) = n - i$, for every i.

Necessary and sufficient cond. for homogenization [**B**S22+]

Given a n-dim compact toric variety X,

Complete intersections over toric varieties

Given $f_1, \dots, f_r \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

When can homogenize? i.e $\dim(\hom(f_1), \dots, \hom(f_i)) = n - i$, for every i.

Necessary and sufficient cond. for homogenization [**B**S22+]

Given a n-dim compact toric variety X, monomials sets $A_1, \dots, A_r \subset \texttt{Cox}(X)$,

Complete intersections over toric varieties

Given $f_1, \dots, f_r \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

When can homogenize? i.e $\dim(\hom(f_1), \dots, \hom(f_i)) = n - i$, for every i.

Necessary and sufficient cond. for homogenization [**B**S22+]

Given a n-dim compact toric variety X, monomials sets $A_1, \dots, A_r \subset \texttt{Cox}(X)$,
For generic system $f_1^h, \dots, f_r^h$ s.t. f_i^h only contains monomials in A_i,

Complete intersections over toric varieties

Given $f_1, \dots, f_r \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

When can homogenize? i.e $\dim(\hom(f_1), \dots, \hom(f_i)) = n - i$, for every i.

Necessary and sufficient cond. for homogenization [BS22+]

Given a n-dim compact toric variety X, monomials sets $A_1, \dots, A_r \subset \texttt{Cox}(X)$,
For generic system $f_1^h, \dots, f_r^h$ s.t. f_i^h only contains monomials in A_i,

- Combinatorial criterion to determine $\texttt{dim}(\{f_1^h = \cdots = f_r^h = 0\})$ in X.

Complete intersections over toric varieties

Given $f_1, \dots, f_r \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

When can homogenize? i.e $\dim(\hom(f_1), \dots, \hom(f_i)) = n - i$, for every i.

Necessary and sufficient cond. for homogenization [**B**S22+]

Given a n-dim compact toric variety X, monomials sets $A_1, \dots, A_r \subset \texttt{Cox}(X)$,
For generic system $f_1^h, \dots, f_r^h$ s.t. f_i^h only contains monomials in A_i,

- Combinatorial criterion to determine $\texttt{dim}(\{f_1^h = \cdots = f_r^h = 0\})$ in X.

In classical projective setting $\mathbb{P}^n$

Fix monomial sets $A_1, \dots, A_r$, s.t. $\deg(x^\alpha) = \deg(x^\beta)$, for every $x^\alpha, x^\beta \in A_i$.
Choose generic system $f_1, \dots, f_r$ such that

$$f_i = \sum_{x^\alpha \in A_i} c_{i,\alpha}\, x^\alpha, \text{ for generic } c_{i,\alpha} \in \mathbb{C}.$$

- We determine $\dim(V_{\mathbb{P}^n}(f_1, \dots, f_r))$.

Complete intersections over toric varieties

Given $f_1, \dots, f_r \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

When can homogenize? i.e $\dim(\hom(f_1), \dots, \hom(f_i)) = n - i$, for every i.

Necessary and sufficient cond. for homogenization [**B**S22+]

Given a n-dim compact toric variety X, monomials sets $A_1, \dots, A_r \subset \texttt{Cox}(X)$,
For generic system $f_1^h, \dots, f_r^h$ s.t. f_i^h only contains monomials in A_i,

- Combinatorial criterion to determine $\texttt{dim}(\{f_1^h = \cdots = f_r^h = 0\})$ in X.

Generalizes sufficient cond. ($r = n$) by [Rojas'94],[Bihan&Soprunov'19],[Chen'19].

Complete intersections over toric varieties

Given $f_1, \dots, f_r \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

When can homogenize? i.e $\dim(\hom(f_1), \dots, \hom(f_i)) = n - i$, for every i.

Necessary and sufficient cond. for homogenization [**B**S22+]

Given a n-dim compact toric variety X, monomials sets $A_1, \dots, A_r \subset \texttt{Cox}(X)$,
For generic system $f_1^h, \dots, f_r^h$ s.t. f_i^h only contains monomials in A_i,

- Combinatorial criterion to determine $\texttt{dim}(\{f_1^h = \cdots = f_r^h = 0\})$ in X.

Complete intersections over toric varieties

Given $f_1, \dots, f_r \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

When can homogenize? i.e $\dim(\hom(f_1), \dots, \hom(f_i)) = n - i$, for every i.

Necessary and sufficient cond. for homogenization [**B**S22+]

Given a n-dim compact toric variety X, monomials sets $A_1, \dots, A_r \subset \texttt{Cox}(X)$,
For generic system $f_1^h, \dots, f_r^h$ s.t. f_i^h only contains monomials in A_i,

- Combinatorial criterion to determine $\texttt{dim}(\{f_1^h = \cdots = f_r^h = 0\})$ in X.

Our approach extends to rational polytopes.

Complete intersections over toric varieties

Given $f_1, \dots, f_r \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

When can homogenize? i.e $\dim(\hom(f_1), \dots, \hom(f_i)) = n - i$, for every i.

Necessary and sufficient cond. for homogenization [BS22+]

Given a n-dim compact toric variety X, monomials sets $A_1, \dots, A_r \subset \texttt{Cox}(X)$,
For generic system $f_1^h, \dots, f_r^h$ s.t. f_i^h only contains monomials in A_i,

- Combinatorial criterion to determine $\texttt{dim}(\{f_1^h = \cdots = f_r^h = 0\})$ in X.

Our approach extends to rational polytopes.

Regular sequences for weighted projective spaces [BS22+]

For $\mathbb{C}[\boldsymbol{x}]$ such that $deg(x_i) = w_i$. A generic system with degrees $d_1, \dots, d_r$ is a regular sequence, if and only if, for all $S \subset \{0, \dots, n\}$,

$$|\{i : d_i \in \sum_{j \in S} w_j \, \mathbb{Z}_{\geq 0}\}| \geq |S| + r - n - 1.$$

Complete intersections over toric varieties

Given $f_1, \dots, f_r \in \mathbb{C}[\boldsymbol{x}]$ and **different polytopes** $P_i = \texttt{NewtonPolytope}(f_i)$.

When can homogenize? i.e $\dim(\hom(f_1), \dots, \hom(f_i)) = n - i$, for every i.

Necessary and sufficient cond. for homogenization [BS22+]

Given a n-dim compact toric variety X, monomials sets $A_1, \dots, A_r \subset \texttt{Cox}(X)$,
For generic system $f_1^h, \dots, f_r^h$ s.t. f_i^h only contains monomials in A_i,

- Combinatorial criterion to determine $\texttt{dim}(\{f_1^h = \cdots = f_r^h = 0\})$ in X.

Our approach extends to rational polytopes.

Regular sequences for weighted projective spaces [BS22+]

For $\mathbb{C}[\boldsymbol{x}]$ such that $deg(x_i) = w_i$. A generic system with degrees $d_1, \dots, d_r$ is a regular sequence, if and only if, for all $S \subset \{0, \dots, n\}$,

$$|\{i : d_i \in \sum_{j \in S} w_j \, \mathbb{Z}_{\geq 0}\}| \geq |S| + r - n - 1.$$

Necessary for specialized GB algorithms [Faugère,Safey El Din&Verron'16]

Determinantal formulas for mixed multilinear systems

Joint work with J.-C. Faugère, A. Mantzaflaris & E. Tsigaridas.

Solving using resultant formulas

- Until now, solve via Sylvester map ($\boldsymbol{g} \mapsto \sum f_i\, g_i$).

Solving using resultant formulas

- Until now, solve via Sylvester map ($\boldsymbol{g} \mapsto \sum f_i\, g_i$).
- To solve using smaller matrices $\rightarrow$ more general maps.

Solving using resultant formulas

- Until now, solve via Sylvester map ($\boldsymbol{g} \mapsto \sum f_i\, g_i$).
- To solve using smaller matrices $\rightarrow$ more general maps.
- Compute resultant of $\boldsymbol{f} \rightarrow$ formula as factor in det. of matrix $M(\boldsymbol{f})$.

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f}) \cdot \texttt{ExtraFactor}(\boldsymbol{f}).$$

Solving using resultant formulas

- Until now, solve via Sylvester map ($\boldsymbol{g} \mapsto \sum f_i\, g_i$).
- To solve using smaller matrices $\rightarrow$ more general maps.
- Compute resultant of $\boldsymbol{f} \rightarrow$ formula as factor in det. of matrix $M(\boldsymbol{f})$.

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f}) \cdot \texttt{ExtraFactor}(\boldsymbol{f}).$$

Generalization of solving strategy for Sylvester maps [**B**FMT18]

- Solve using general class of resultant formulas.

Solving using resultant formulas

- Until now, solve via Sylvester map ($\boldsymbol{g} \mapsto \sum f_i\, g_i$).
- To solve using smaller matrices $\rightarrow$ more general maps.
- Compute resultant of $\boldsymbol{f}$ $\rightarrow$ formula as factor in det. of matrix $M(\boldsymbol{f})$.

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f}) \cdot \texttt{ExtraFactor}(\boldsymbol{f}).$$

Generalization of solving strategy for Sylvester maps [**B**FMT18]

- Solve using general class of resultant formulas.
- Do not compute the resultant $\rightarrow$ only linear algebra.

Solving using resultant formulas

- Until now, solve via Sylvester map ($\boldsymbol{g} \mapsto \sum f_i\, g_i$).
- To solve using smaller matrices $\rightarrow$ more general maps.
- Compute resultant of $\boldsymbol{f}$ $\rightarrow$ formula as factor in det. of matrix $M(\boldsymbol{f})$.

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f}) \cdot \texttt{ExtraFactor}(\boldsymbol{f}).$$

Generalization of solving strategy for Sylvester maps [**B**FMT18]

- Solve using general class of resultant formulas.
- Do not compute the resultant $\rightarrow$ only linear algebra.
- Solve by computing eigenvalues. (Matrices are not mult. maps!)

Example : Solving using Koszul-type formula

$$\left\{\begin{array}{l} f_1 := 7\,x_0y_0 + -8\,x_0y_1 + -1\,x_1y_0 + 2\,x_1y_1 \\ f_2 := -5\,x_0y_0 + 7\,x_0y_1 + -1\,x_1y_0 + -1\,x_1y_1 \\ f_3 := -6\,x_0z_0 + 9\,x_0z_1 + -1\,x_1z_0 + -2\,x_1z_1 \end{array}\right\} \begin{array}{l} \in \mathbb{C}[\boldsymbol{X}, \boldsymbol{Y}] \\ \in \mathbb{C}[\boldsymbol{X}, \boldsymbol{Z}] \end{array}$$

Example : Solving using Koszul-type formula

$$\left\{\begin{array}{rl} f_0 := & 3\,x_0y_0z_0 + -1x_0y_0z_1 + -4\,x_0y_1z_0 + 2\,x_0y_1z_1 \\ & +1x_1y_0z_0 + 2\,x_1y_0z_1 + 2\,x_1y_1z_0 + -2\,x_1y_1z_1 \\ f_1 := & 7\,x_0y_0 + -8\,x_0y_1 + -1\,x_1y_0 + 2\,x_1y_1 \\ f_2 := & -5\,x_0y_0 + 7\,x_0y_1 + -1\,x_1y_0 + -1\,x_1y_1 \\ f_3 := & -6\,x_0z_0 + 9\,x_0z_1 + -1\,x_1z_0 + -2\,x_1z_1 \end{array}\right. \begin{array}{l} \left.\begin{array}{l} \\ \\ \end{array}\right\} \in \mathbb{C}[\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}] \\ \left.\begin{array}{l} \\ \end{array}\right\} \in \mathbb{C}[\boldsymbol{X},\boldsymbol{Y}] \\ \in \mathbb{C}[\boldsymbol{X},\boldsymbol{Z}] \end{array}$$

Example : Solving using Koszul-type formula

$$\left\{\begin{array}{l} f_0 := 3\,x_0y_0z_0 + -1x_0y_0z_1 + -4\,x_0y_1z_0 + 2\,x_0y_1z_1 \\ \qquad\quad +1x_1y_0z_0 + 2\,x_1y_0z_1 + 2\,x_1y_1z_0 + -2\,x_1y_1z_1 \\ f_1 := 7\,x_0y_0 + -8\,x_0y_1 + -1\,x_1y_0 + 2\,x_1y_1 \\ f_2 := -5\,x_0y_0 + 7\,x_0y_1 + -1\,x_1y_0 + -1\,x_1y_1 \\ f_3 := -6\,x_0z_0 + 9\,x_0z_1 + -1\,x_1z_0 + -2\,x_1z_1 \end{array}\right. \begin{array}{l} \left.\begin{array}{l}\\\\\end{array}\right\} \in \mathbb{C}[\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}] \\ \left.\begin{array}{l}\\\end{array}\right\} \in \mathbb{C}[\boldsymbol{X},\boldsymbol{Y}] \\ \in \mathbb{C}[\boldsymbol{X},\boldsymbol{Z}] \end{array}$$

$$M = \begin{bmatrix} & & & 5 & -7 & 1 & 1 & & & \\ & & & 7 & -8 & -1 & 2 & & & \\ & -1 & & & & & & -1 & -5 & 7 \\ 7 & & -1 & & & & & -1 & & -5 \\ & 1 & & & & & & -2 & -7 & 8 \\ 8 & & -2 & & & & & 1 & & -7 \\ & 2 & & 9 & & -2 & & -2 & -1 & 2 \\ 2 & & -2 & & 9 & & -2 & 2 & & -1 \\ & 1 & & -6 & & -1 & & 2 & 3 & -4 \\ -4 & & 2 & & -6 & & -1 & 1 & & 3 \end{bmatrix}$$

Example : Solving using Koszul-type formula

$$\begin{cases} f_0 := & \boxed{3}\, x_0y_0z_0 + -1x_0y_0z_1 + -4\,x_0y_1z_0 + 2\,x_0y_1z_1 \\ & +1x_1y_0z_0 + 2\,x_1y_0z_1 + 2\,x_1y_1z_0 + -2\,x_1y_1z_1 \\ f_1 := & 7\,x_0y_0 + -8\,x_0y_1 + -1\,x_1y_0 + 2\,x_1y_1 \\ f_2 := & -5\,x_0y_0 + 7\,x_0y_1 + -1\,x_1y_0 + -1\,x_1y_1 \\ f_3 := & -6\,x_0z_0 + 9\,x_0z_1 + -1\,x_1z_0 + -2\,x_1z_1 \end{cases}$$

$$\left[\begin{array}{c|c} M_{1,1} & M_{1,2} \\ \hline M_{2,1} & \boldsymbol{M_{2,2}} \end{array}\right] = \left[\begin{array}{cccccccc|cc} & & & 5 & -7 & 1 & 1 & & & \\ & & & 7 & -8 & -1 & 2 & & & \\ & -1 & & & & & & -1 & -5 & 7 \\ 7 & & -1 & & & & & -1 & & -5 \\ & 1 & & & & & & -2 & -7 & 8 \\ 8 & & -2 & & & & & 1 & & -7 \\ & 2 & & 9 & & -2 & & -2 & -1 & 2 \\ 2 & & -2 & & 9 & & -2 & 2 & & -1 \\ \hline & 1 & & -6 & & -1 & & 2 & \boxed{3} & -4 \\ -4 & & 2 & & -6 & & -1 & 1 & & \boxed{3} \end{array}\right]$$

Example : Solving using Koszul-type formula

$$\begin{cases} f_0 := & \boxed{3}\, x_0y_0z_0 + -1x_0y_0z_1 + -4\,x_0y_1z_0 + 2\,x_0y_1z_1 \\ & +1x_1y_0z_0 + 2\,x_1y_0z_1 + 2\,x_1y_1z_0 + -2\,x_1y_1z_1 \\ f_1 := & 7\,x_0y_0 + -8\,x_0y_1 + -1\,x_1y_0 + 2\,x_1y_1 \\ f_2 := & -5\,x_0y_0 + 7\,x_0y_1 + -1\,x_1y_0 + -1\,x_1y_1 \\ f_3 := & -6\,x_0z_0 + 9\,x_0z_1 + -1\,x_1z_0 + -2\,x_1z_1 \end{cases}$$

$$\left[\begin{array}{c|c} M_{1,1} & M_{1,2} \\ \hline M_{2,1} & \boldsymbol{M_{2,2}} \end{array}\right] = \left[\begin{array}{cccccccc|cc} & & & 5 & -7 & 1 & 1 & & & \\ & & & 7 & -8 & -1 & 2 & & & \\ & -1 & & & & & & -1 & -5 & 7 \\ 7 & & -1 & & & & & -1 & & -5 \\ & 1 & & & & & & -2 & -7 & 8 \\ 8 & & -2 & & & & & 1 & & -7 \\ & 2 & & 9 & & -2 & & -2 & -1 & 2 \\ 2 & & -2 & & 9 & & -2 & 2 & & -1 \\ \hline & 1 & & -6 & & -1 & & 2 & \boxed{3} & -4 \\ -4 & & 2 & & -6 & & -1 & 1 & & \boxed{3} \end{array}\right]$$

$$\widetilde{M}_{2,2} := \left(M_{2,2} - M_{2,1} \cdot M_{1,1}^{-1} \cdot M_{1,2}\right) = \begin{bmatrix} 5 & -2 \\ 4 & -1 \end{bmatrix}$$

Example : Solving using Koszul-type formula

$$\begin{cases} f_0 := & \boxed{3}\, x_0y_0z_0 + -1x_0y_0z_1 + -4\,x_0y_1z_0 + 2\,x_0y_1z_1 \\ & +1x_1y_0z_0 + 2\,x_1y_0z_1 + 2\,x_1y_1z_0 + -2\,x_1y_1z_1 \\ f_1 := & 7\,x_0y_0 + -8\,x_0y_1 + -1\,x_1y_0 + 2\,x_1y_1 \\ f_2 := & -5\,x_0y_0 + 7\,x_0y_1 + -1\,x_1y_0 + -1\,x_1y_1 \\ f_3 := & -6\,x_0z_0 + 9\,x_0z_1 + -1\,x_1z_0 + -2\,x_1z_1 \end{cases}$$

$$\left[\begin{array}{c|c} M_{1,1} & M_{1,2} \\ \hline M_{2,1} & \boldsymbol{M_{2,2}} \end{array}\right] = \left[\begin{array}{cccccccc|cc} & & & 5 & -7 & 1 & 1 & & & \\ & & & 7 & -8 & -1 & 2 & & & \\ & -1 & & & & & & -1 & -5 & 7 \\ 7 & & -1 & & & & & -1 & & -5 \\ & 1 & & & & & & -2 & -7 & 8 \\ 8 & & -2 & & & & & 1 & & -7 \\ & 2 & & 9 & & -2 & & -2 & -1 & 2 \\ 2 & & -2 & & 9 & & -2 & 2 & & -1 \\ \hline & 1 & & -6 & & -1 & & 2 & \boxed{3} & -4 \\ -4 & & 2 & & -6 & & -1 & 1 & & \boxed{3} \end{array}\right]$$

$$\widetilde{M}_{2,2} := \left(M_{2,2} - M_{2,1} \cdot M_{1,1}^{-1} \cdot M_{1,2}\right) = \begin{bmatrix} 5 & -2 \\ 4 & -1 \end{bmatrix}$$

(f_1, f_2, f_3) has 2 solutions

$$\left.\begin{array}{l} (1{:}1\ ;\ 1{:}1\ ;\ 1{:}1) \\ (1{:}3\ ;\ 1{:}2\ ;\ 1{:}3) \end{array}\right\} \in \mathbb{P}^1 \times \mathbb{P}^1 \times \mathbb{P}^1$$

Eigenvalues of $\widetilde{M}_{2,2}$

$$\frac{f_0}{x_0y_0z_0}\left((1{:}1\ ;\ 1{:}1\ ;\ 1{:}1)\right) = 3$$
$$\frac{f_0}{x_0y_0z_0}\left((1{:}3\ ;\ 1{:}2\ ;\ 1{:}3)\right) = 1$$

Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [BFMT18]

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f}) \cdot \texttt{ExtraFactor}(\boldsymbol{f}).$$

Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [BFMT18]

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f}) \cdot \texttt{ExtraFactor}(\boldsymbol{f}).$$

- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [BFMT18]
- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [BFMT18]
- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

- Determinantal formulas do not exist for general systems.

Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [**B**FMT18]
- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

- Determinantal formulas do not exist for general systems.

Determinantal formulas for mixed multilinear sys. [**B**FMT21]

- First det. formulas for mixed sparse systems (different Newton polytopes).

Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [**B**FMT18]
- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

- Determinantal formulas do not exist for general systems.

Determinantal formulas for mixed multilinear sys. [**B**FMT21]

- First det. formulas for mixed sparse systems (different Newton polytopes).
- Special mixed multilinear sys: $\{f_0, \ldots, f_n\} \subset \mathbb{C}[\boldsymbol{X}_1] \otimes \ldots \otimes \mathbb{C}[\boldsymbol{X}_A] \otimes \mathbb{C}[\boldsymbol{Y}_1] \otimes \ldots \otimes \mathbb{C}[\boldsymbol{Y}_B]$.

Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [**B**FMT18]
- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

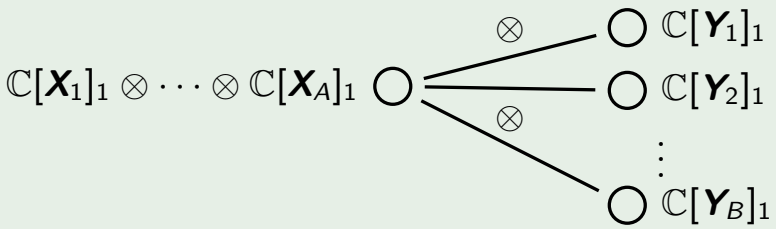
- Determinantal formulas do not exist for general systems.

Determinantal formulas for mixed multilinear sys. [**B**FMT21]

- First det. formulas for mixed sparse systems (different Newton polytopes).
- Special mixed multilinear sys: $\{f_0, \dots, f_n\} \subset \mathbb{C}[\boldsymbol{X}_1] \otimes \dots \otimes \mathbb{C}[\boldsymbol{X}_A] \otimes \mathbb{C}[\boldsymbol{Y}_1] \otimes \dots \otimes \mathbb{C}[\boldsymbol{Y}_B]$.

Star multilinear system:

$$f_k \in \mathbb{C}[\boldsymbol{X}_1]_1 \otimes \dots \mathbb{C}[\boldsymbol{X}_A]_1 \otimes \mathbb{C}[\boldsymbol{Y}_{j_k}]_1.$$



Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [**B**FMT18]
- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

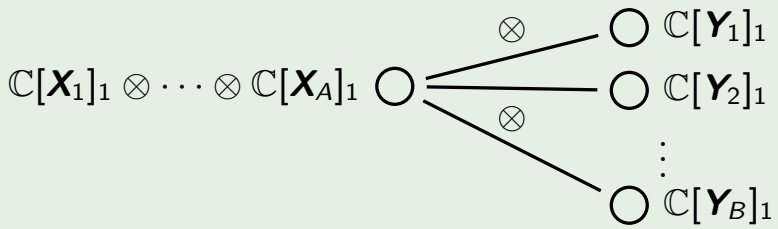
- Determinantal formulas do not exist for general systems.

Determinantal formulas for mixed multilinear sys. [**B**FMT21]

- First det. formulas for mixed sparse systems (different Newton polytopes).
- Special mixed multilinear sys: $\{f_0, \dots, f_n\} \subset \mathbb{C}[\boldsymbol{X}_1] \otimes \dots \otimes \mathbb{C}[\boldsymbol{X}_A] \otimes \mathbb{C}[\boldsymbol{Y}_1] \otimes \dots \otimes \mathbb{C}[\boldsymbol{Y}_B]$.

Star multilinear system:

$$f_k \in \mathbb{C}[\boldsymbol{X}_1]_1 \otimes \dots \mathbb{C}[\boldsymbol{X}_A]_1 \otimes \mathbb{C}[\boldsymbol{Y}_{j_k}]_1.$$



Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [**B**FMT18]
- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

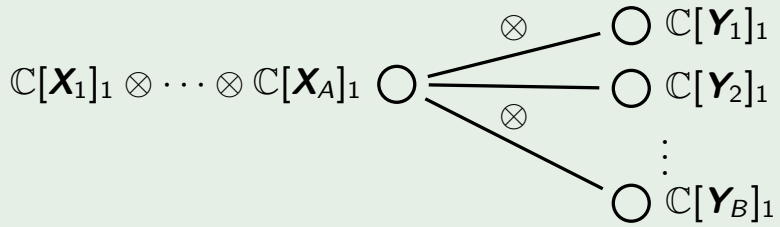
- Determinantal formulas do not exist for general systems.

Determinantal formulas for mixed multilinear sys. [**B**FMT21]

- First det. formulas for mixed sparse systems (different Newton polytopes).
- Special mixed multilinear sys: $\{f_0, \dots, f_n\} \subset \mathbb{C}[\boldsymbol{X}_1] \otimes \dots \otimes \mathbb{C}[\boldsymbol{X}_A] \otimes \mathbb{C}[\boldsymbol{Y}_1] \otimes \dots \otimes \mathbb{C}[\boldsymbol{Y}_B]$.

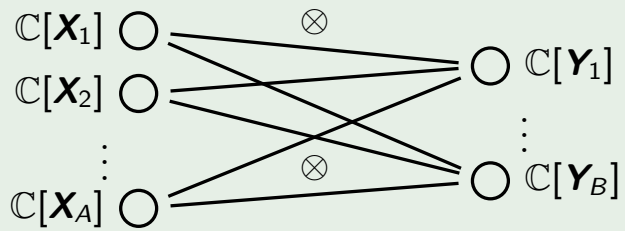
Star multilinear system:

$$f_k \in \mathbb{C}[\boldsymbol{X}_1]_1 \otimes \dots \mathbb{C}[\boldsymbol{X}_A]_1 \otimes \mathbb{C}[\boldsymbol{Y}_{j_k}]_1.$$



Bipartite bilinear system:

$$f_k \in \mathbb{C}[\boldsymbol{X}_{i_k}]_1 \otimes \mathbb{C}[\boldsymbol{Y}_{j_k}]_1.$$



Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [**B**FMT18]
- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

- Determinantal formulas do not exist for general systems.

Determinantal formulas for mixed multilinear sys. [**B**FMT21]

- First det. formulas for mixed sparse systems (different Newton polytopes).
- Special families of mixed multilinear systems.

Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [**B**FMT18]
- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

- Determinantal formulas do not exist for general systems.

Determinantal formulas for mixed multilinear sys. [**B**FMT21]

- First det. formulas for mixed sparse systems (different Newton polytopes).
- Special families of mixed multilinear systems.
- These systems arise in applications $\rightarrow$ Multiparameter eigenvalue problem.

Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [**B**FMT18]
- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

- Determinantal formulas do not exist for general systems.

Determinantal formulas for mixed multilinear sys. [**B**FMT21]

- First det. formulas for mixed sparse systems (different Newton polytopes).
- Special families of mixed multilinear systems.
- These systems arise in applications $\rightarrow$ Multiparameter eigenvalue problem.

Generalized Eigenvalue Problem

$$\left(\begin{bmatrix} -7 & -3 \\ -8 & -2 \end{bmatrix} - \lambda \begin{bmatrix} 12 & 2 \\ 13 & 1 \end{bmatrix} \right) \begin{bmatrix} v_0 \\ v_1 \end{bmatrix} = 0 \quad .$$

Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [**B**FMT18]
- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

- Determinantal formulas do not exist for general systems.

Determinantal formulas for mixed multilinear sys. [**B**FMT21]

- First det. formulas for mixed sparse systems (different Newton polytopes).
- Special families of mixed multilinear systems.
- These systems arise in applications $\rightarrow$ Multiparameter eigenvalue problem.
  - Generalization of the Generalized Eigenvalue Problem

$$\left(\lambda_0 \begin{bmatrix} -7 & -3 \\ -8 & -2 \end{bmatrix} + \lambda_1 \begin{bmatrix} 12 & 2 \\ 13 & 1 \end{bmatrix} + \lambda_2 \begin{bmatrix} -7 & -1 \\ -7 & -1 \end{bmatrix}\right) \begin{bmatrix} v_0 \\ v_1 \end{bmatrix} = 0$$

$$\left(\lambda_0 \begin{bmatrix} -11 & -3 \\ 4 & 1 \end{bmatrix} + \lambda_1 \begin{bmatrix} 7 & -1 \\ 1 & 2 \end{bmatrix} + \lambda_2 \begin{bmatrix} -4 & 0 \\ -1 & -1 \end{bmatrix}\right) \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = 0 \,.$$

Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [**B**FMT18]
- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

- Determinantal formulas do not exist for general systems.

Determinantal formulas for mixed multilinear sys. [**B**FMT21]

- First det. formulas for mixed sparse systems (different Newton polytopes).
- Special families of mixed multilinear systems.
- These systems arise in applications $\rightarrow$ Multiparameter eigenvalue problem.
  - Generalization of the Generalized Eigenvalue Problem
  - Applications in physics (Sturm-Liouville theory)

$$\left(\lambda_0 \begin{bmatrix} -7 & -3 \\ -8 & -2 \end{bmatrix} + \lambda_1 \begin{bmatrix} 12 & 2 \\ 13 & 1 \end{bmatrix} + \lambda_2 \begin{bmatrix} -7 & -1 \\ -7 & -1 \end{bmatrix}\right) \begin{bmatrix} v_0 \\ v_1 \end{bmatrix} = 0$$

$$\left(\lambda_0 \begin{bmatrix} -11 & -3 \\ 4 & 1 \end{bmatrix} + \lambda_1 \begin{bmatrix} 7 & -1 \\ 1 & 2 \end{bmatrix} + \lambda_2 \begin{bmatrix} -4 & 0 \\ -1 & -1 \end{bmatrix}\right) \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = 0$$

Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [**B**FMT18]
- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

- Determinantal formulas do not exist for general systems.

Determinantal formulas for mixed multilinear sys. [**B**FMT21]

- First det. formulas for mixed sparse systems (different Newton polytopes).
- Special families of mixed multilinear systems.
- These systems arise in applications $\rightarrow$ Multiparameter eigenvalue problem.
  - Generalization of the Generalized Eigenvalue Problem
  - Applications in physics (Sturm-Liouville theory)

$$\begin{bmatrix} (-7\,\lambda_0 + 12\,\lambda_1 - 7\,\lambda_2) & (-3\,\lambda_0 + 2\,\lambda_1 - \lambda_2) \\ (-8\,\lambda_0 + 13\,\lambda_1 - 7\,\lambda_2) & (-2\,\lambda_0 + \lambda_1 - \lambda_2) \end{bmatrix} \cdot \begin{bmatrix} v_0 \\ v_1 \end{bmatrix} = 0$$

$$\begin{bmatrix} (-11\,\lambda_0 + 7\,\lambda_1 - 4\,\lambda_2) & (-3\,\lambda_0 - \lambda_1) \\ (4\,\lambda_0 + \lambda_1 - \lambda_2) & (\lambda_0 + 2\,\lambda_1 - \lambda_2) \end{bmatrix} \cdot \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = 0$$

Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [**B**FMT18]
- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

- Determinantal formulas do not exist for general systems.

Determinantal formulas for mixed multilinear sys. [**B**FMT21]

- First det. formulas for mixed sparse systems (different Newton polytopes).
- Special families of mixed multilinear systems.
- These systems arise in applications $\rightarrow$ Multiparameter eigenvalue problem.
  - Generalization of the Generalized Eigenvalue Problem
  - Applications in physics (Sturm-Liouville theory)

$$\begin{cases} (-7\,\lambda_0 + 12\,\lambda_1 - 7\,\lambda_2)\; v_0 + (-3\,\lambda_0 + 2\,\lambda_1 - \lambda_2)\, v_1 = 0 \\ (-8\,\lambda_0 + 13\,\lambda_1 - 7\,\lambda_2)\; v_0 + (-2\,\lambda_0 + \lambda_1 - \lambda_2)\, v_1 = 0 \\ (-11\,\lambda_0 + 7\,\lambda_1 - 4\,\lambda_2)\; w_0 + (-3\,\lambda_0 - \lambda_1)\; w_1 = 0 \\ (4\,\lambda_0 + \lambda_1 - \lambda_2)\; w_0 + (\lambda_0 + 2\,\lambda_1 - \lambda_2)\; w_1 = 0 \end{cases}.$$

Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [**B**FMT18]
- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

- Determinantal formulas do not exist for general systems.

Determinantal formulas for mixed multilinear sys. [**B**FMT21]

- First det. formulas for mixed sparse systems (different Newton polytopes).
- Special families of mixed multilinear systems.
- These systems arise in applications $\rightarrow$ Multiparameter eigenvalue problem.
  - Generalization of the Generalized Eigenvalue Problem
  - Applications in physics (Sturm-Liouville theory)

$$\begin{cases} f_1 := (-7\,\lambda_0 + 12\,\lambda_1 - 7\,\lambda_2)\; v_0 + (-3\,\lambda_0 + 2\,\lambda_1 - \lambda_2)\; v_1 \\ f_2 := (-8\,\lambda_0 + 13\,\lambda_1 - 7\,\lambda_2)\; v_0 + (-2\,\lambda_0 + \lambda_1 - \lambda_2)\; v_1 \\ f_3 := (-11\,\lambda_0 + 7\,\lambda_1 - 4\,\lambda_2)\; w_0 + (-3\,\lambda_0 - \lambda_1)\; w_1 \\ f_4 := (4\,\lambda_0 + \lambda_1 - \lambda_2)\; w_0 + (\lambda_0 + 2\,\lambda_1 - \lambda_2)\; w_1 \end{cases}.$$

Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [**B**FMT18]
- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

- Determinantal formulas do not exist for general systems.

Determinantal formulas for mixed multilinear sys. [**B**FMT21]

- First det. formulas for mixed sparse systems (different Newton polytopes).
- Special families of mixed multilinear systems.
- These systems arise in applications $\rightarrow$ Multiparameter eigenvalue problem.
  - Generalization of the Generalized Eigenvalue Problem
  - Applications in physics (Sturm-Liouville theory)

$$\begin{cases} f_1 := (-7\,\lambda_0 + 12\,\lambda_1 - 7\,\lambda_2)\; v_0 + (-3\,\lambda_0 + 2\,\lambda_1 - \lambda_2)\; v_1 \\ f_2 := (-8\,\lambda_0 + 13\,\lambda_1 - 7\,\lambda_2)\; v_0 + (-2\,\lambda_0 + \lambda_1 - \lambda_2)\; v_1 \\ f_3 := (-11\,\lambda_0 + 7\,\lambda_1 - 4\,\lambda_2)\; w_0 + (-3\,\lambda_0 - \lambda_1)\; w_1 \\ f_4 := (4\,\lambda_0 + \lambda_1 - \lambda_2)\; w_0 + (\lambda_0 + 2\,\lambda_1 - \lambda_2)\; w_1 \end{cases}.$$

Determinantal formulas for mixed multilinear systems

- We can solve using matrix $M(\boldsymbol{f})$ (in certain class) [**B**FMT18]
- Smallest possible matrix $\leftrightarrow$ no $\texttt{ExtraFactor}(\boldsymbol{f})$ (Determinantal Formula)

$$\texttt{Det}(M(\boldsymbol{f})) = \texttt{Resultant}(\boldsymbol{f})$$

- Determinantal formulas do not exist for general systems.

Determinantal formulas for mixed multilinear sys. [**B**FMT21]

- First det. formulas for mixed sparse systems (different Newton polytopes).
- Special families of mixed multilinear systems.
- These systems arise in applications $\rightarrow$ Multiparameter eigenvalue problem.
  - Generalization of the Generalized Eigenvalue Problem
  - Applications in physics (Sturm-Liouville theory)

$$\begin{cases} f_1 := (-7\,\lambda_0 + 12\,\lambda_1 - 7\,\lambda_2)\; v_0 + (-3\,\lambda_0 + 2\,\lambda_1 - \lambda_2)\; v_1 \\ f_2 := (-8\,\lambda_0 + 13\,\lambda_1 - 7\,\lambda_2)\; v_0 + (-2\,\lambda_0 + \lambda_1 - \lambda_2)\; v_1 \\ f_3 := (-11\,\lambda_0 + 7\,\lambda_1 - 4\,\lambda_2)\; w_0 + (-3\,\lambda_0 - \lambda_1)\; w_1 \\ f_4 := (4\,\lambda_0 + \lambda_1 - \lambda_2)\; w_0 + (\lambda_0 + 2\,\lambda_1 - \lambda_2)\; w_1 \end{cases}.$$

Binary form decomposition

Joint work with J.-C. Faugère, L. Perret & E. Tsigaridas.

Binary form decomposition

Binary form of degree $D \longleftrightarrow$ Symmetric tensor of dimension $2 \times \cdots \times 2$ and order D.

Problem

Given a binary form f of degree D, $a_i \in \mathbb{K} \subset \mathbb{C}$,

$$f(x,y) := \sum_{i=0}^{D} a_i \, x^i y^{D-i} \in \mathbb{C}[x,y]_D$$

compute $\lambda_1, \ldots, \lambda_r$, $\alpha_1, \ldots, \alpha_r$, $\beta_1, \ldots, \beta_r \in \bar{\mathbb{K}}$ and r (**rank**) minimal such that,

$$f(x,y) = \sum_{j=1}^{r} \lambda_j \, (\alpha_j \, x + \beta_j \, y)^D$$

Results

[BFPT16],[BFPT21]

(ACM's SIGSAM Distinguished Student Author Award at ISSAC 2016)

- Quasi-optimal algorithm to decompose binary forms in $\widetilde{O}(D)$ ops.
- Tight bounds for the algebraic degree of the decomposition.

Other applications

Topological data analysis [B., Gäfvert, Lesnick '22+]

- Multiparameter persistent homology (MPH) generalizes persistent homology.

Other applications

Topological data analysis [B., Gäfvert, Lesnick '22+]

- Multiparameter persistent homology (MPH) generalizes persistent homology.
- MPH can be computed using GB. [Carlsson, Singh & Zomorodian '09]

Other applications

Topological data analysis [B., Gäfvert, Lesnick '22+]

- Multiparameter persistent homology (MPH) generalizes persistent homology.
- MPH can be computed using GB. [Carlsson, Singh & Zomorodian '09]
- In practice, available software too slow for interesting inputs.

Other applications

Topological data analysis [B., Gäfvert, Lesnick '22+]

- Multiparameter persistent homology (MPH) generalizes persistent homology.
- MPH can be computed using GB. [Carlsson, Singh & Zomorodian '09]
- In practice, available software too slow for interesting inputs.
- We exploit the structure to optimize the GB computations.

Other applications

Topological data analysis [B., Gäfvert, Lesnick '22+]

- Multiparameter persistent homology (MPH) generalizes persistent homology.
- MPH can be computed using GB. [Carlsson, Singh & Zomorodian '09]
- In practice, available software too slow for interesting inputs.
- We exploit the structure to optimize the GB computations.
- We derive complexity bounds for computing MPH.

Other applications

Topological data analysis [B., Gäfvert, Lesnick '22+]

- Multiparameter persistent homology (MPH) generalizes persistent homology.
- MPH can be computed using GB. [Carlsson, Singh & Zomorodian '09]
- In practice, available software too slow for interesting inputs.
- We exploit the structure to optimize the GB computations.
- We derive complexity bounds for computing MPH.
- Soon available: `Muphasa` (Implemented in C++).

Other applications

Topological data analysis [B., Gäfvert, Lesnick '22+]

- Multiparameter persistent homology (MPH) generalizes persistent homology.
- MPH can be computed using GB. [Carlsson, Singh & Zomorodian '09]
- In practice, available software too slow for interesting inputs.
- We exploit the structure to optimize the GB computations.
- We derive complexity bounds for computing MPH.
- Soon available: `Muphasa` (Implemented in C++).

Other applications

Topological data analysis [B., Gäfvert, Lesnick '22+]

- Multiparameter persistent homology (MPH) generalizes persistent homology.
- MPH can be computed using GB. [Carlsson, Singh & Zomorodian '09]
- In practice, available software too slow for interesting inputs.
- We exploit the structure to optimize the GB computations.
- We derive complexity bounds for computing MPH.
- Soon available: `Muphasa` (Implemented in C++).

Computational biology [Schwieger, B., Siebert & Haase '20]

- Gröbner-basis-based approach to construct classifiers for Boolean networks.