Algebraic attacks for the Rank Decoding Problem

Magali Bardet

magali.bardet@univ-rouen.fr

JNCF 2024,
March 4–8, 2024

Laboratoire d'Informatique, du Traitement de l'Information et des Systèmes

Université de Rouen Normandie

NIST CALL FOR PROPOSALS

Post-Quantum Cryptography standardization process, 2017–2022–

- KEM + Signature.
- based on mathematical problems resistant to quantum computer.
- 4 Rounds since 2017.
- first selection for standardization in 07/2022:
  - 1 lattice-based KEM;
  - 2 lattice-based signatures;
  - 1 Hash-based signature.
- 3 code-based KEMs in the 4th Round.

Additional Digital Signature Schemes

- June 1, 2023. First Round ongoing.
- 40 submissions, with:
  - multivariate cryptography (12).
  - code-based cryptography (11).
  - Symmetric-based cryptography (4).
  - Lattice-based cryptography (7).
  - Other (6).

NIST call for Digital Signatures

Additional Digital Signature Schemes

- June 1, 2023. First Round ongoing.
- 40 submissions, with:
  - multivariate cryptography (12).
  - code-based cryptography (11).
  - Symmetric-based cryptography (4).
  - Lattice-based cryptography (7).
  - Other (6).

Algebraic approaches are at the core of security assessment for multivariate and code-based cryptography.

Principle: write a Polynomial System

$$\begin{cases} f_1(x_1, \ldots, x_n) \\ \vdots \\ f_m(x_1, \ldots, x_n) \end{cases}, \quad \deg(f_i) = d_i, f_i \in \mathbb{F}_q[x_1, \ldots, x_n].$$

such that finding the set of solutions gives (part of) the secret:

$$V(f_1, \ldots, f_m) = \left\{ (x_1, \ldots, x_n) \in \overline{\mathbb{F}_q}^n : f_i(x_1, \ldots, x_n) = 0, \forall i \in \{1..m\} \right\}$$

Principle: write a Polynomial System

$$\begin{cases} f_1(x_1,\dots,x_n) \\ \vdots \\ f_m(x_1,\dots,x_n) \end{cases}, \quad \deg(f_i) = d_i, f_i \in \mathbb{F}_q[x_1,\dots,x_n].$$

such that finding the set of solutions gives (part of) the secret:

$$V(f_1,\dots,f_m) = \{(x_1,\dots,x_n) \in \overline{\mathbb{F}_q}^n : f_i(x_1,\dots,x_n) = 0, \forall i \in \{1..m\}\}$$

- Key-recovery attack.
- Message-recovery attack.
- Signature forgery attack.

Ideally: any solution is related to the secret!

Relations between Solutions and Secrets

Ideally: any solution is related to the secret!

- Otherwise, we have to deal with spurious solutions $\rightarrow$ change the modeling!

Relations between Solutions and Secrets

Ideally: any solution is related to the secret!

- Otherwise, we have to deal with spurious solutions $\rightarrow$ change the modeling!
- Only solutions in $\mathbb{F}_q$

Relations between Solutions and Secrets

Ideally: any solution is related to the secret!

- Otherwise, we have to deal with spurious solutions $\rightarrow$ change the modeling!
- Only solutions in $\mathbb{F}_q$
  - **Combinatorial** approach = try "all possible solutions" efficiently (often solve a linear system).

Relations between Solutions and Secrets

Ideally: any solution is related to the secret!

- Otherwise, we have to deal with spurious solutions $\rightarrow$ change the modeling!
- Only solutions in $\mathbb{F}_q$
  - **Combinatorial** approach = try "all possible solutions" efficiently (often solve a linear system).
  - **Algebraic** approach: solve an algebraic system with algebraic constraints $x_i^q - x_i$!

Ideally: any solution is related to the secret!

- Otherwise, we have to deal with spurious solutions $\rightarrow$ change the modeling!
- Only solutions in $\mathbb{F}_q$
  - **Combinatorial** approach = try "all possible solutions" efficiently (often solve a linear system).
  - **Algebraic** approach: solve an algebraic system with algebraic constraints $x_i^q - x_i$!
  - Combinatorial vs Algebraic approaches: $\rightarrow$ **hybrid** approach (better over a Small finite field).

Ideally: any solution is related to the secret!

- Otherwise, we have to deal with spurious solutions → change the modeling!
- Only solutions in $\mathbb{F}_q$
  - **Combinatorial** approach = try "all possible solutions" efficiently (often solve a linear system).
  - **Algebraic** approach: solve an algebraic system with algebraic constraints $x_i^q - x_i$!
  - Combinatorial vs Algebraic approaches: → **hybrid** approach (better over a Small finite field).
  - Large prime field? 😬

Relations between Solutions and Secrets

Ideally: any solution is related to the secret!

- Otherwise, we have to deal with spurious solutions $\rightarrow$ change the modeling!
- Only solutions in $\mathbb{F}_q$
  - **Combinatorial** approach = try “all possible solutions” efficiently (often solve a linear system).
  - **Algebraic** approach: solve an algebraic system with algebraic constraints $x_i^q - x_i$!
  - Combinatorial vs Algebraic approaches: $\rightarrow$ **hybrid** approach (better over a Small finite field).
  - Large prime field? 😬
- Cryptographic applications: always a finite number of solutions (one of them is enough).

Ideally: any solution is related to the secret!

- Otherwise, we have to deal with spurious solutions $\rightarrow$ change the modeling!
- Only solutions in $\mathbb{F}_q$
  - **Combinatorial** approach = try "all possible solutions" efficiently (often solve a linear system).
  - **Algebraic** approach: solve an algebraic system with algebraic constraints $x_i^q - x_i$!
  - Combinatorial vs Algebraic approaches: $\rightarrow$ **hybrid** approach (better over a Small finite field).
  - Large prime field? 😬
- Cryptographic applications: always a finite number of solutions (one of them is enough).
- Often 0 or 1 solution, but sometimes m solutions over $\mathbb{F}_{q^m}$.

Signature forgery (or Message-recovery attack)

- Public key: a polynomial system, indistinguishable from a random system.

$$\begin{cases} f_1(x_1,\ldots,x_n) \\ \vdots \\ f_m(x_1,\ldots,x_n) \end{cases}, \quad \deg(f_i) = 2, \quad f_i \in \mathbb{F}_q[x_1,\ldots,x_n].$$

- $(y_1,\ldots,y_m)$ hash of the message to be signed (or ciphertext).
- signature (or cleartext) = $(x_1,\ldots,x_n)$ such that $(y_1,\ldots,y_m) = (f_1(\boldsymbol{x}),\ldots,f_m(\boldsymbol{x}))$
- Secret key: a trapdoor to solve the system efficiently = Hash and sign.

Signature forgery (or Message-recovery attack)

- Public key: a polynomial system, indistinguishable from a random system.

$$\begin{cases} f_1(x_1,\dots,x_n) \\ \vdots \\ f_m(x_1,\dots,x_n) \end{cases}, \quad \deg(f_i) = 2, \quad f_i \in \mathbb{F}_q[x_1,\dots,x_n].$$

- $(y_1,\dots,y_m)$ hash of the message to be signed (or ciphertext).
- signature (or cleartext) = $(x_1,\dots,x_n)$ such that $(y_1,\dots,y_m) = (f_1(\boldsymbol{x}),\dots,f_m(\boldsymbol{x}))$
- Secret key: a trapdoor to solve the system efficiently = Hash and sign.
- other approach: Zero-knowledge proof of knowledge.

Signature forgery (or Message-recovery attack)

- Public key: a polynomial system, indistinguishable from a random system.

$$\begin{cases} f_1(x_1,\ldots,x_n) \\ \vdots \\ f_m(x_1,\ldots,x_n) \end{cases}, \quad \deg(f_i) = 2, \quad f_i \in \mathbb{F}_q[x_1,\ldots,x_n].$$

- $(y_1,\ldots,y_m)$ hash of the message to be signed (or ciphertext).
- signature (or cleartext) = $(x_1,\ldots,x_n)$ such that $(y_1,\ldots,y_m) = (f_1(\mathbf{x}),\ldots,f_m(\mathbf{x}))$
- Secret key: a trapdoor to solve the system efficiently = Hash and sign.
- other approach: Zero-knowledge proof of knowledge.

How hard is it to solve a random system of algebraic equations?

Multivariate public-key cryptography

Signature forgery (or Message-recovery attack)

- Public key: a polynomial system, indistinguishable from a random system.

$$\begin{cases} f_1(x_1,\dots,x_n) \\ \vdots \\ f_m(x_1,\dots,x_n) \end{cases}, \quad \deg(f_i) = 2, \quad f_i \in \mathbb{F}_q[x_1,\dots,x_n].$$

- $(y_1,\dots,y_m)$ hash of the message to be signed (or ciphertext).
- signature (or cleartext) = $(x_1,\dots,x_n)$ such that $(y_1,\dots,y_m) = (f_1(\mathbf{x}),\dots,f_m(\mathbf{x}))$
- Secret key: a trapdoor to solve the system efficiently = Hash and sign.
- other approach: Zero-knowledge proof of knowledge.

How hard is it to solve a random system of algebraic equations?
How hard is it to solve a trapdoored system of algebraic equations?

Solving the algebraic system using Gröbner bases (object)

- A particular basis of the ideal

$$I = \langle f_1, \ldots, f_m \rangle = \left\{ \sum_{i=1}^{m} g_i f_i : g_i \in \mathbb{F}_q[x_1, \ldots, x_n] \right\}$$

  that solves the ideal-membership problem: $f \overset{?}{\in} I$.
- Depends on the choice of a monomial ordering.

Monomial ordering examples

$$x_1 \quad x_3 \quad 1 \quad x_3^3 \quad x_1x_3 \quad x_2^2 \quad x_1^2$$

Monomial ordering examples

$$x_1 \quad x_3 \quad 1 \quad x_3^3 \quad x_1x_3 \quad x_2^2 \quad x_1^2$$

Lexicographical ordering $x_1 > \cdots > x_n$

$$x_1^{\alpha_1} \dots x_n^{\alpha_n} > x_1^{\beta_1} \dots x_n^{\beta_n} \text{ iff } \qquad \alpha_j = \beta_j \quad \forall j < i, \text{ and } \alpha_i > \beta_i.$$

Monomial ordering examples

$$x_1 \quad x_3 \quad 1 \quad x_3^3 \quad x_1x_3 \quad x_2^2 \quad x_1^2$$

Lexicographical ordering $x_1 > \cdots > x_n$

$$x_1^{\alpha_1}\dots x_n^{\alpha_n} > x_1^{\beta_1}\dots x_n^{\beta_n} \text{ iff } \qquad \alpha_j = \beta_j \quad \forall j < i, \text{ and } \alpha_i > \beta_i.$$

$$x_1^2 > x_1x_3 > x_1 > x_2^2 > x_3^3 > x_3 > 1$$

Monomial ordering examples

$$x_1 \quad x_3 \quad 1 \quad x_3^3 \quad x_1x_3 \quad x_2^2 \quad x_1^2$$

Lexicographical ordering $x_1 > \cdots > x_n$

$$x_1^{\alpha_1} \ldots x_n^{\alpha_n} > x_1^{\beta_1} \ldots x_n^{\beta_n} \text{ iff } \qquad \alpha_j = \beta_j \quad \forall j < i, \text{ and } \alpha_i > \beta_i.$$

$$x_1^2 > x_1x_3 > x_1 > x_2^2 > x_3^3 > x_3 > 1$$

Graded Reverse Lexicographical ordering $x_1 > \cdots > x_n$

$$x_1^{\alpha_1} \ldots x_n^{\alpha_n} > x_1^{\beta_1} \ldots x_n^{\beta_n} \text{ iff } \begin{cases} \deg(\boldsymbol{x}^{\alpha}) > \deg(\boldsymbol{x}^{\beta}) \\ \text{or } \alpha_j = \beta_j \ \forall j > i, \text{ and } \alpha_i < \beta_i. \end{cases}$$

Monomial ordering examples

$$x_1 \quad x_3 \quad 1 \quad x_3^3 \quad x_1x_3 \quad x_2^2 \quad x_1^2$$

Lexicographical ordering $x_1 > \cdots > x_n$

$$x_1^{\alpha_1} \ldots x_n^{\alpha_n} > x_1^{\beta_1} \ldots x_n^{\beta_n} \text{ iff } \qquad \alpha_j = \beta_j \quad \forall j < i, \text{ and } \alpha_i > \beta_i.$$

$$x_1^2 > x_1x_3 > x_1 > x_2^2 > x_3^3 > x_3 > 1$$

Graded Reverse Lexicographical ordering $x_1 > \cdots > x_n$

$$x_1^{\alpha_1} \ldots x_n^{\alpha_n} > x_1^{\beta_1} \ldots x_n^{\beta_n} \text{ iff } \begin{cases} \deg(\boldsymbol{x}^{\alpha}) > \deg(\boldsymbol{x}^{\beta}) \\ \text{or } \alpha_j = \beta_j \ \forall j > i, \text{ and } \alpha_i < \beta_i. \end{cases}$$

$$x_3^3 > x_1^2 > x_2^2 > x_1x_3 > x_1 > x_3 > 1$$

Different monomial orderings have different properties

- the *lex* order (Lexicographical): in Shape Position, for a zero-dimension ideal, the (reduced) lex basis is

$$\begin{cases} x_1 - & g_1(x_n), \\ x_2 - & g_2(x_n), \\ \vdots & \\ x_{n-1} - & g_{n-1}(x_n), \\ & g_n(x_n), \end{cases}$$

  with $\deg(g_n) = D$ the number of solutions to the system.
- the *grevlex* order (Graded Reverse Lexicographical): usually the best one w.r.t. the complexity.

The (reduced) grevlex and lex bases are the same:

- If the system has no solution:

$$\langle 1 \rangle .$$

Systems with 0 or 1 solution

The (reduced) grevlex and lex bases are the same:

- If the system has no solution:

$$\langle 1 \rangle .$$

- If the system has 1 solution:

$$\begin{cases} x_1 - a_1, \\ \vdots \\ x_n - a_n, \end{cases}$$

where $(a_1, \ldots, a_n) \in \mathbb{F}_q^n$ is the solution.

For zero-dimensional systems:

- The FGLM (J.-C. Faugère, Gianni, Daniel Lazard, and Mora (1993)) Algorithm performs a change of ordering in complexity

$$O(nD^3),$$

n number of variables, $n \to \infty$, D degree of the ideal (number of solutions).

- Complexity for grevlex to lex (Shape position) (J.-C. Faugère, Gaudry, Huot, and Renault (2014)):

$$O(\log_2(D)(D^\omega + n\log_2(D)D)).$$

ω coefficient of linear algebra.

Change of ordering

For zero-dimensional systems:

- The FGLM (J.-C. Faugère, Gianni, Daniel Lazard, and Mora (1993)) Algorithm performs a change of ordering in complexity

$$O(nD^3),$$

  n number of variables, $n \to \infty$, D degree of the ideal (number of solutions).
- Complexity for grevlex to lex (Shape position) (J.-C. Faugère, Gaudry, Huot, and Renault (2014)):

$$O(\log_2(D)(D^\omega + n\log_2(D)D)).$$

ω coefficient of linear algebra.

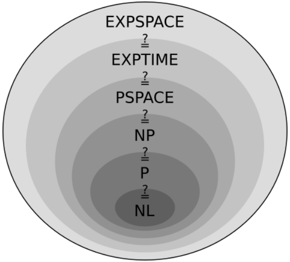
We focus on the grevlex ordering

Complexity classes

A Gröbner basis solves the Ideal Membership problem.

A hard problem

- Ideal Membership testing is EXPSPACE-complete,
- Existence of solutions to a system of polynomial equations over a finite field is NP-complete (Fraenkel and Yesha (1979)),



For cryptographic applications

- We need precise estimates for concrete parameters.
- Asymptotic estimates are also appreciated.
- The security levels are $2^{143}, 2^{207}$ and 2^{272} bits operations.
- Take the best algorithm (combinatorial, algebraic, hybrid, ...).

General algorithms, for any input system:

- Buchberger (1965);
- F4 from J.-C. Faugère (1999);

The algorithms will always terminate and give the Gröbner basis.
But the time is hard to predict for *any* instance.

General algorithms, for any input system:

- Buchberger (1965);
- F4 from J.-C. Faugère (1999);

The algorithms will always terminate and give the Gröbner basis.
But the time is hard to predict for *any* instance.

Specific algorithms, for a particular class of systems:

- The algorithms will terminate in a predictable time.
- The result is not always a Gröbner basis of the system.
- For random instances in the specific class, the result is a Gröbner basis.

$$\text{System} \begin{cases} f_1(x_1,\ldots,x_n) \\ \vdots \\ f_m(x_1,\ldots,x_n) \end{cases}, \quad \deg(f_i) = d_i, f_i \in \mathbb{F}_q[x_1,\ldots,x_n].$$

- Macaulay Matrices Macaulay (1902):

$$\mathscr{M}_d(\{f_1,\ldots,f_m\}) = \begin{array}{c} \vdots \\ (\boldsymbol{x}^\alpha, i) \\ \vdots \end{array} \overset{\boldsymbol{x}^\beta}{\begin{pmatrix} & & \\ & \mathrm{coeff}(\boldsymbol{x}^\alpha f_i, \boldsymbol{x}^\beta) & \\ & & \end{pmatrix}}$$

$\deg(\boldsymbol{x}^\alpha f_i) = d = \deg(\boldsymbol{x}^\beta)$.

Example: 3 quadratic equations in 3 variables, $\mathbb{F}_5$

$$
\begin{cases}
x_1^2 + 3x_1x_2 + x_2^2 + x_1x_3 + 2x_2x_3 + 2x_3^2, & (f_1) \\
x_1^2 + 4x_1x_2 + 3x_2^2 + 4x_1x_3 + 3x_3^2, & (f_2) \\
x_1^2 + 2x_2^2 + 4x_2x_3 + 3x_3^2. & (f_3)
\end{cases}
$$

EXAMPLE: 3 QUADRATIC EQUATIONS IN 3 VARIABLES, $\mathbb{F}_5$

$$
\begin{cases}
x_1^2 + 3x_1x_2 + x_2^2 + x_1x_3 + 2x_2x_3 + 2x_3^2, & (f_1)\\
x_1^2 + 4x_1x_2 + 3x_2^2 + 4x_1x_3 + 3x_3^2, & (f_2)\\
x_1^2 + 2x_2^2 + 4x_2x_3 + 3x_3^2. & (f_3)
\end{cases}
$$

$$
\mathscr{M}_2 = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \end{array}
\begin{array}{cccccc} x_1^2 & x_1x_2 & x_2^2 & x_1x_3 & x_2x_3 & x_3^2 \end{array}
\begin{pmatrix}
{\color{red}1} & 3 & 1 & 1 & 2 & 2\\
{\color{blue}1} & 4 & 3 & 4 & 0 & 3\\
{\color{blue}1} & 0 & 2 & 0 & 4 & 3
\end{pmatrix}
$$

Example: 3 quadratic equations in 3 variables, $\mathbb{F}_5$

$$\begin{cases} x_1^2 + 3x_1x_2 + x_2^2 + x_1x_3 + 2x_2x_3 + 2x_3^2, & (f_1) \\ x_1^2 + 4x_1x_2 + 3x_2^2 + 4x_1x_3 + 3x_3^2, & (f_2) \\ x_1^2 + 2x_2^2 + 4x_2x_3 + 3x_3^2. & (f_3) \end{cases}$$

$$Ech(\mathscr{M}_2) = \begin{array}{c} \\ \tilde{f}_1 \\ \tilde{f}_2 \\ \tilde{f}_3 \end{array} \begin{array}{cccccc} x_1^2 & x_1x_2 & x_2^2 & x_1x_3 & x_2x_3 & x_3^2 \\ \hline 1 & & & 2 & 3 & 4 \\ 0 & 1 & & & 2 & 2 \\ 0 & & 1 & 4 & 3 & 2 \end{array}$$

$$\left\{\begin{array}{ccccccccccc}
x_1^2 & + & 3x_1x_2 & + & x_2^2 & + & x_1x_3 & + & 2x_2x_3 & + & 2x_3^2, \\
x_1^2 & + & 4x_1x_2 & + & 3x_2^2 & + & 4x_1x_3 & & & + & 3x_3^2, \\
x_1^2 & & & + & 2x_2^2 & & & + & 4x_2x_3 & + & 3x_3^2.
\end{array}\right.$$

$\mathscr{M}_3 =$

	x_1^3	$x_1^2x_2$	$x_1x_2^2$	x_2^3	$x_1^2x_3$	$x_1x_2x_3$	$x_2^2x_3$	$x_1x_3^2$	$x_2x_3^2$	x_3^3
x_3f_1					1	3	1	1	2	2
x_2f_1		1	3	1		1	2		2	
x_1f_1	1	3	1		1	2		2		
x_3f_2					1	4	3	4	0	3
x_2f_2		1	4	3		4	0		3	
x_1f_2	1	4	3		4	0		3		
x_3f_3					1	0	2	0	4	3
x_2f_3		1	0	2		0	4		3	
x_1f_3	1	0	2		0	4		3		

$$
\begin{cases}
x_1^2 & & & + & 2x_1x_3 & + & 3x_2x_3 & + & 4x_3^2, \\
& x_1x_2 & & & & + & 2x_2x_3 & + & 2x_3^2, \\
& & x_2^2 & + & 4x_1x_3 & + & 3x_2x_3 & + & 2x_3^2.
\end{cases}
$$

$$
\mathscr{M}_3 = \begin{array}{c} \\ x_3\tilde{f}_1 \\ x_2\tilde{f}_1 \\ x_1\tilde{f}_1 \\ x_3\tilde{f}_2 \\ x_2\tilde{f}_2 \\ x_1\tilde{f}_2 \\ x_3\tilde{f}_3 \\ x_2\tilde{f}_3 \\ x_1\tilde{f}_3 \end{array}
\begin{array}{c}
\begin{array}{cccccccccc} x_1^3 & x_1^2x_2 & x_1x_2^2 & x_2^3 & x_1^2x_3 & x_1x_2x_3 & x_2^2x_3 & x_1x_3^2 & x_2x_3^2 & x_3^3 \end{array} \\
\begin{pmatrix}
 & & & & {\color{red}1} & & & 2 & 3 & 4 \\
 & {\color{red}1} & & & & 2 & 3 & & 4 & \\
{\color{red}1} & & & & 2 & 3 & & 4 & & \\
 & & & & & {\color{red}1} & & & 2 & 2 \\
 & & {\color{red}1} & & & & 2 & & 2 & \\
 & {\color{blue}1} & & & & 2 & & 2 & & \\
 & & & & & & {\color{red}1} & 4 & 3 & 2 \\
 & & & {\color{red}1} & & 4 & 3 & & 2 & \\
 & & {\color{blue}1} & & 4 & 3 & & 2 & &
\end{pmatrix}
\end{array}
$$

$$\left\{\begin{array}{ccccccccc}
x_1^2 & & & + & 2x_1x_3 & + & 3x_2x_3 & + & 4x_3^2, \\
& x_1x_2 & & & & + & 2x_2x_3 & + & 2x_3^2, \\
& & x_2^2 & + & 4x_1x_3 & + & 3x_2x_3 & + & 2x_3^2.
\end{array}\right.$$

$$Ech(\mathscr{M}_3) = \begin{array}{c} \\ x_3\tilde{f}_1 \\ x_2\tilde{f}_1 \\ x_1\tilde{f}_1 \\ x_3\tilde{f}_2 \\ x_2\tilde{f}_2 \\ x_1\tilde{f}_2 \\ x_3\tilde{f}_3 \\ x_2\tilde{f}_3 \\ x_1\tilde{f}_3 \end{array}
\begin{array}{c}
\begin{array}{cccccccccc} x_1^3 & x_1^2x_2 & x_1x_2^2 & x_2^3 & x_1^2x_3 & x_1x_2x_3 & x_2^2x_3 & x_1x_3^2 & x_2x_3^2 & x_3^3 \end{array} \\
\begin{pmatrix}
 & & & & 1 & & & & & 4 \\
 & 1 & & & & & & & & 4 \\
1 & & & & & & & & & 4 \\
 & & & & & 1 & & & & 4 \\
 & & 1 & & & & & & & 4 \\
 & 0 & & & & & & 1 & & 4 \\
 & & & & & & 1 & & & 4 \\
 & & & 1 & & & & & & 4 \\
 & & 0 & & & & & & 1 & 4
\end{pmatrix}
\end{array}$$

Gröbner basis via linear algebra

$$\text{Gröbner Basis} = \begin{cases} x_1x_3^2 + 4x_3^3, & (x_1f_2) \\ x_2x_3^2 + 4x_3^3, & (x_1f_3) \\ x_1^2 + 2x_1x_3 + 3x_2x_3 + 4x_3^2, & (f_1) \\ x_1x_2 + 2x_2x_3 + 2x_3^2, & (f_2) \\ x_2^2 + 4x_1x_3 + 3x_2x_3 + 2x_3^2 & (f_3). \end{cases}$$

One projective solution: $(1, 1, 1)$.

Example: 3 quadratic equations in 3 variables, $\mathbb{F}_5$

$$
\begin{cases}
x_1^2 + 3x_1x_2 + x_2^2 + x_1x_3 + 2x_2x_3 + 2x_3^2, \\
x_1^2 + 4x_1x_2 + 3x_2^2 + 4x_1x_3 + 3x_3^2, \\
x_1^2 + 2x_2^2 + 4x_2x_3 + {\color{red}3}x_3^2.
\end{cases}
$$

Example: 3 quadratic equations in 3 variables, $\mathbb{F}_5$

$$\begin{cases} x_1^2 + 3x_1x_2 + x_2^2 + x_1x_3 + 2x_2x_3 + 2x_3^2, \\ x_1^2 + 4x_1x_2 + 3x_2^2 + 4x_1x_3 + 3x_3^2, \\ x_1^2 + 2x_2^2 + 4x_2x_3 + 1x_3^2. \end{cases}$$

$$\mathcal{M}_2 = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \end{array} \begin{array}{c} \begin{array}{cccccc} x_1^2 & x_1x_2 & x_2^2 & x_1x_3 & x_2x_3 & x_3^2 \end{array} \\ \begin{pmatrix} 1 & 3 & 1 & 1 & 2 & 2 \\ 1 & 4 & 3 & 4 & 0 & 3 \\ 1 & 0 & 2 & 0 & 4 & 1 \end{pmatrix} \end{array}$$

Example: 3 quadratic equations in 3 variables, $\mathbb{F}_5$

$$\begin{cases} x_1^2 + 3x_1x_2 + x_2^2 + x_1x_3 + 2x_2x_3 + 2x_3^2, \\ x_1^2 + 4x_1x_2 + 3x_2^2 + 4x_1x_3 + 3x_3^2, \\ x_1^2 + 2x_2^2 + 4x_2x_3 + 1x_3^2. \end{cases}$$

$$Ech(\mathscr{M}_2) = \begin{array}{c} \\ \tilde{f}_1 \\ \tilde{f}_2 \\ \tilde{f}_3 \end{array} \begin{array}{cccccc} x_1^2 & x_1x_2 & x_2^2 & x_1x_3 & x_2x_3 & x_3^2 \\ \end{array} \begin{pmatrix} 1 & & & 2 & 3 & 4 \\ 0 & 1 & & & 2 & 4 \\ 0 & & 1 & 4 & 3 & 2 \end{pmatrix}$$

Example: 3 quadratic equations in 3 variables, $\mathbb{F}_5$

$Ech(\mathscr{M}_3) =$

	x_1^3	$x_1^2x_2$	$x_1x_2^2$	x_2^3	$x_1^2x_3$	$x_1x_2x_3$	$x_2^2x_3$	$x_1x_3^2$	$x_2x_3^2$	x_3^3
$x_3\tilde{f}_1$					1					1
$x_2\tilde{f}_1$		1								1
$x_1\tilde{f}_1$	1									3
$x_3\tilde{f}_2$						1				1
$x_2\tilde{f}_2$			1							0
$x_1\tilde{f}_2$		0						1		3
$x_3\tilde{f}_3$							1			2
$x_2\tilde{f}_3$				1						1
$x_1\tilde{f}_3$			0						1	4

Example: 3 quadratic equations in 3 variables, $\mathbb{F}_5$

$$
Ech(\mathscr{M}_3) = \begin{array}{c} \\ x_3\tilde{f}_1 \\ x_2\tilde{f}_1 \\ x_1\tilde{f}_1 \\ x_3\tilde{f}_2 \\ x_2\tilde{f}_2 \\ x_1\tilde{f}_2 \\ x_3\tilde{f}_3 \\ x_2\tilde{f}_3 \\ x_1\tilde{f}_3 \end{array}
\begin{array}{c}
\begin{array}{cccccccccc} x_1^3 & x_1^2x_2 & x_1x_2^2 & x_2^3 & x_1^2x_3 & x_1x_2x_3 & x_2^2x_3 & x_1x_3^2 & x_2x_3^2 & x_3^3 \end{array} \\
\begin{pmatrix}
 & & & & 1 & & & & & 1 \\
 & 1 & & & & & & & & 1 \\
1 & & & & & & & & & 3 \\
 & & & & & 1 & & & & 1 \\
 & & 1 & & & & & & & 0 \\
 & 0 & & & & & & 1 & & 3 \\
 & & & & & & 1 & & & 2 \\
 & & & 1 & & & & & & 1 \\
 & & 0 & & & & & & 1 & 4
\end{pmatrix}
\end{array}
$$

x_1f_3 vs x_3f_3: need to go to degree $D = 4$ to get the Gröbner Basis.

At $D = 4$:

- $\binom{6}{4} = 15$ monomials of degree 4,
- $3\binom{4}{2} = 18$ rows tf_i of degree 4,
- $\mathscr{M}_4$ has rank $15 \to 3$ rows reduce to 0 ($x_1^2 f_2, x_1 x_2 f_3, x_1^2 f_3$), 1 new polynomial ($x_1 x_3 f_3$).

Gröbner basis via linear algebra

At $D = 4$:

- $\binom{6}{4} = 15$ monomials of degree 4,
- $3\binom{4}{2} = 18$ rows tf_i of degree 4,
- $\mathscr{M}_4$ has rank $15 \to 3$ rows reduce to 0 $(x_1^2 f_2, x_1x_2f_3, x_1^2 f_3)$, 1 new polynomial $(x_1x_3f_3)$.

$$\text{Gröbner Basis} = \begin{cases} x_3^4, & (x_1x_3f_3) \\ x_1x_3^2 + 3x_3^3, & (x_1f_2) \\ x_2x_3^2 + 4x_3^3, & (x_1f_3) \\ x_1^2 + 2x_1x_3 + 3x_2x_3 + 4x_3^2, & (f_1) \\ x_1x_2 + 2x_2x_3 + 4x_3^2, & (f_2) \\ x_2^2 + 4x_1x_3 + 3x_2x_3 + x_3^2 & (f_3). \end{cases}$$

Gröbner basis via linear algebra

At $D = 4$:

- $\binom{6}{4} = 15$ monomials of degree 4,
- $3\binom{4}{2} = 18$ rows tf_i of degree 4,
- $\mathscr{M}_4$ has rank $15 \to 3$ rows reduce to 0 $(x_1^2 f_2, x_1x_2f_3, x_1^2 f_3)$, 1 new polynomial $(x_1x_3f_3)$.

$$\text{Gröbner Basis} = \begin{cases} x_3^4, & (x_1x_3f_3) \\ x_1x_3^2 + 3x_3^3, & (x_1f_2) \\ x_2x_3^2 + 4x_3^3, & (x_1f_3) \\ x_1^2 + 2x_1x_3 + 3x_2x_3 + 4x_3^2, & (f_1) \\ x_1x_2 + 2x_2x_3 + 4x_3^2, & (f_2) \\ x_2^2 + 4x_1x_3 + 3x_2x_3 + x_3^2 & (f_3). \end{cases}$$

First system:

- $\mathscr{M}_4$ has rank $14 \to 4$ rows reduce to 0, no new polynomial.

Do we need to compute the Gröbner basis?

- easy to recover the value of all variables from the evaluation of all monomials of degree D.
  e.g. from ${x_n}^D = \alpha$ and $x_i {x_n}^{D-1} = \beta$ we get $x_i = \frac{\beta}{\alpha} x_n$ (or $x_n = 0$).

Do we need to compute the Gröbner basis?

- easy to recover the value of all variables from the evaluation of all monomials of degree D.
  e.g. from ${x_n}^D = \alpha$ and $x_i {x_n}^{D-1} = \beta$ we get $x_i = \frac{\beta}{\alpha} x_n$ (or $x_n = 0$).
- evaluation of all monomials of degree D on a solution $\Rightarrow$ a vector $\boldsymbol{t}$ such that $\mathscr{M}_D(\{f_1, \ldots, f_m\})\boldsymbol{t} = 0$

Do we need to compute the Gröbner basis?

- easy to recover the value of all variables from the evaluation of all monomials of degree D.
  e.g. from ${x_n}^D = \alpha$ and $x_i {x_n}^{D-1} = \beta$ we get $x_i = \frac{\beta}{\alpha} x_n$ (or $x_n = 0$).
- evaluation of all monomials of degree D on a solution $\Rightarrow$ a vector $\boldsymbol{t}$ such that $\mathscr{M}_D(\{f_1, \ldots, f_m\})\boldsymbol{t} = 0$
- Homogeneous system with 0 or 1 solution:

$$\mathrm{Rk}_D = \mathrm{Mon}_D \text{ or } \mathrm{Rk}_D = \mathrm{Mon}_D - 1.$$

$\Rightarrow$ only computes the kernel of $\mathscr{M}_D$ (instead of a basis of $\mathscr{M}_{\leq D}$):

Do we need to compute the Gröbner basis?

- easy to recover the value of all variables from the evaluation of all monomials of degree D.
  e.g. from ${x_n}^D = \alpha$ and $x_i {x_n}^{D-1} = \beta$ we get $x_i = \frac{\beta}{\alpha} x_n$ (or $x_n = 0$).
- evaluation of all monomials of degree D on a solution $\Rightarrow$ a vector $\boldsymbol{t}$ such that $\mathscr{M}_D(\{f_1, \ldots, f_m\})\boldsymbol{t} = 0$
- Homogeneous system with 0 or 1 solution:

$$\mathrm{Rk}_D = \mathrm{Mon}_D \text{ or } \mathrm{Rk}_D = \mathrm{Mon}_D - 1.$$

$\Rightarrow$ only computes the kernel of $\mathscr{M}_D$ (instead of a basis of $\mathscr{M}_{\leq D}$):

- no need for RREF!

Bi-homogeneous systems

$$f_i = \sum_{i,j} c_{i,j} x_i y_j \in \mathbb{F}_q[\boldsymbol{x}, \boldsymbol{y}].$$

Macaulay matrix at bi-degree (d_1, d_2) = the vector space $\langle \boldsymbol{x}^\alpha \boldsymbol{y}^\beta f_i \rangle$ with $\deg(\boldsymbol{x}^\alpha) = d_1 - 1$, $\deg(\boldsymbol{y}^\beta) = d_2 - 1$.

Bi-homogeneous systems

$$f_i = \sum_{i,j} c_{i,j} x_i y_j \in \mathbb{F}_q[\boldsymbol{x}, \boldsymbol{y}].$$

Macaulay matrix at bi-degree (d_1, d_2) = the vector space $\langle \boldsymbol{x}^\alpha \boldsymbol{y}^\beta f_i \rangle$ with $\deg(\boldsymbol{x}^\alpha) = d_1 - 1$, $\deg(\boldsymbol{y}^\beta) = d_2 - 1$.

- $\mathscr{M}_D$ is a block diagonal matrix of the $\mathscr{M}_{d_1,d_2}$'s

Bi-homogeneous systems

$$f_i = \sum_{i,j} c_{i,j} x_i y_j \in \mathbb{F}_q[\boldsymbol{x}, \boldsymbol{y}].$$

Macaulay matrix at bi-degree (d_1, d_2) = the vector space $\langle \boldsymbol{x}^\alpha \boldsymbol{y}^\beta f_i \rangle$ with $\deg(\boldsymbol{x}^\alpha) = d_1 - 1$, $\deg(\boldsymbol{y}^\beta) = d_2 - 1$.

- $\mathscr{M}_D$ is a block diagonal matrix of the $\mathscr{M}_{d_1,d_2}$'s
- easy to recover the value of all variables from the evaluation of all monomial of bi-degree (d_1, d_2)
  e.g. from ${x_1}^{d_1}{y_1}^{d_2} = \alpha$ and ${x_1}^{d_1-1} x_i {y_1}^{d_2} = \beta$ we get $x_i = \frac{\beta}{\alpha} x_1$ (or $y_1 = 0$).

Bi-homogeneous systems

$$f_i = \sum_{i,j} c_{i,j} x_i y_j \in \mathbb{F}_q[\boldsymbol{x}, \boldsymbol{y}].$$

Macaulay matrix at bi-degree (d_1, d_2) = the vector space $\langle \boldsymbol{x}^\alpha \boldsymbol{y}^\beta f_i \rangle$ with $\deg(\boldsymbol{x}^\alpha) = d_1 - 1$, $\deg(\boldsymbol{y}^\beta) = d_2 - 1$.

- $\mathscr{M}_D$ is a block diagonal matrix of the $\mathscr{M}_{d_1,d_2}$'s
- easy to recover the value of all variables from the evaluation of all monomial of bi-degree (d_1, d_2)
  e.g. from ${x_1}^{d_1}{y_1}^{d_2} = \alpha$ and ${x_1}^{d_1-1} x_i {y_1}^{d_2} = \beta$ we get $x_i = \frac{\beta}{\alpha} x_1$ (or $y_1 = 0$).
- At bi-degree (d_1, d_2), evaluation of all monomials of bi-degree (d_1, d_2) on a solution $\Rightarrow$ a vector $\boldsymbol{t}$ such that $\mathscr{M}_{d_1,d_2}(\{f_1, \ldots, f_m\})\boldsymbol{t} = 0$

Bi-homogeneous systems

$$f_i = \sum_{i,j} c_{i,j} x_i y_j \in \mathbb{F}_q[\mathbf{x}, \mathbf{y}].$$

Macaulay matrix at bi-degree (d_1, d_2) = the vector space $\langle \mathbf{x}^{\alpha} \mathbf{y}^{\beta} f_i \rangle$ with $\deg(\mathbf{x}^{\alpha}) = d_1 - 1$, $\deg(\mathbf{y}^{\beta}) = d_2 - 1$.

- $\mathscr{M}_D$ is a block diagonal matrix of the $\mathscr{M}_{d_1,d_2}$'s
- easy to recover the value of all variables from the evaluation of all monomial of bi-degree (d_1, d_2)
  e.g. from ${x_1}^{d_1} {y_1}^{d_2} = \alpha$ and ${x_1}^{d_1 - 1} x_i {y_1}^{d_2} = \beta$ we get $x_i = \frac{\beta}{\alpha} x_1$ (or $y_1 = 0$).
- At bi-degree (d_1, d_2), evaluation of all monomials of bi-degree (d_1, d_2) on a solution $\Rightarrow$ a vector $\mathbf{t}$ such that $\mathscr{M}_{d_1,d_2}(\{f_1, \ldots, f_m\})\mathbf{t} = 0$
- 0 or 1 solution: the kernel of $\mathscr{M}_{d_1,d_2}$ for $D = d_1 + d_2$ such that:

$$\mathrm{Rk}_{d_1,d_2} = \mathrm{Mon}_{d_1,d_2} \text{ or } \mathrm{Rk}_{d_1,d_2} = \mathrm{Mon}_{d_1,d_2} - 1.$$

Gröbner basis via linear algebra

Rows of Macaulay matrices:

- Describes the vector space $\langle tf_i : \deg(tf_i) = d \rangle_{\mathbb{F}_q}$.
- D. Lazard (1983); Giusti (1984): linear algebra on the Macaulay matrices up to degree $D \rightarrow$ Gröbner basis.
- "Linearization"! with an exponential number of rows/columns.

Rows of Macaulay matrices:

- Describes the vector space $\langle tf_i : \deg(tf_i) = d \rangle_{\mathbb{F}_q}$.
- D. Lazard (1983); Giusti (1984): linear algebra on the Macaulay matrices up to degree $D \to$ Gröbner basis.
- "Linearization"! with an exponential number of rows/columns.

Main challenges to get complexity estimates for Gröbner Basis computations

- Estimate D.
- Estimate the cost of linear algebra.

$\mathbb{C}$ of linear algebra. Jeannerod, Pernet, and Storjohann (2013)

Matrix $\mathscr{M}$ with N rows, Mon columns, rank Rk, and δ non-zero elements per row. Echelon Form can be computed in:

$$C_\omega \times \mathrm{N} \times \mathrm{Mon} \times \mathrm{Rk}^{\omega-2} + o(\mathrm{N}\,\mathrm{Mon}\,\mathrm{Rk}^{\omega-2}), \quad \mathrm{N}, \mathrm{Mon}, \mathrm{Rk} \to \infty,$$

For instance:

- $(\omega, C_\omega) = (3, 1)$ for Gaussian Elimination;
- $(\omega, C_\omega) = (\log_2(7), 4.4)$ for the Strassen Algorithm;

$\mathbb{C}$ of linear algebra. Jeannerod, Pernet, and Storjohann (2013)

Matrix $\mathscr{M}$ with N rows, Mon columns, rank Rk, and δ non-zero elements per row. Echelon Form can be computed in:

$$C_\omega \times \mathrm{N} \times \mathrm{Mon} \times \mathrm{Rk}^{\omega-2} + o(\mathrm{N}\,\mathrm{Mon}\,\mathrm{Rk}^{\omega-2}), \quad \mathrm{N}, \mathrm{Mon}, \mathrm{Rk} \to \infty,$$

For instance:

- $(\omega, C_\omega) = (3, 1)$ for Gaussian Elimination;
- $(\omega, C_\omega) = (\log_2(7), 4.4)$ for the Strassen Algorithm;

Probabilistic Wiedemann (1986) algorithm:

$$3\delta \times \mathrm{N} \times \mathrm{Mon} + o(\delta\,\mathrm{N}\,\mathrm{Mon}), \quad \mathrm{N}, \mathrm{Mon} \to \infty.$$

$\mathbb{C}$ of linear algebra. Jeannerod, Pernet, and Storjohann (2013)

Matrix $\mathscr{M}$ with N rows, Mon columns, rank Rk, and δ non-zero elements per row. Echelon Form can be computed in:

$$C_\omega \times \mathrm{N} \times \mathrm{Mon} \times \mathrm{Rk}^{\omega-2} + o(\mathrm{N}\,\mathrm{Mon}\,\mathrm{Rk}^{\omega-2}), \quad \mathrm{N}, \mathrm{Mon}, \mathrm{Rk} \to \infty,$$

For instance:

- $(\omega, C_\omega) = (3, 1)$ for Gaussian Elimination;
- $(\omega, C_\omega) = (\log_2(7), 4.4)$ for the Strassen Algorithm;

Probabilistic Wiedemann (1986) algorithm:

$$3\delta \times \mathrm{N} \times \mathrm{Mon} + o(\delta\,\mathrm{N}\,\mathrm{Mon}), \quad \mathrm{N}, \mathrm{Mon} \to \infty.$$

These are upper bounds.

Linear dependencies between rows

- the rows of $\mathscr{M}_{\leq D}$ are not linearly independent: e.g.

$$f_k f_\ell - f_\ell f_k = 0.$$

Linear dependencies between rows

- the rows of $\mathscr{M}_{\leq D}$ are not linearly independent: e.g.

$$f_k f_\ell - f_\ell f_k = 0.$$

- relations between the rows are called syzygies of the system.

Linear dependencies between rows

- the rows of $\mathscr{M}_{\leq D}$ are not linearly independent: e.g.

$$f_k f_\ell - f_\ell f_k = 0.$$

- relations between the rows are called syzygies of the system.
- a system has trivial syzygies, and may have other: a system is regular if it has only trivial syzygies.

Linear dependencies between rows

- the rows of $\mathscr{M}_{\leq D}$ are not linearly independent: e.g.

$$f_k f_\ell - f_\ell f_k = 0.$$

- relations between the rows are called syzygies of the system.
- a system has trivial syzygies, and may have other: a system is regular if it has only trivial syzygies.
- F5 criterion J.-C. Faugère (2002) = a criterion to detect syzygies. Can detect all trivial syzygies.

Linear dependencies between rows

- the rows of $\mathscr{M}_{\leq D}$ are not linearly independent: e.g.

$$f_k f_\ell - f_\ell f_k = 0.$$

- relations between the rows are called syzygies of the system.
- a system has trivial syzygies, and may have other: a system is regular if it has only trivial syzygies.
- F5 criterion J.-C. Faugère (2002) = a criterion to detect syzygies. Can detect all trivial syzygies.
- $\rightarrow$ construct a matrix with only Rk_D rows for regular sequences.

Linear dependencies between rows

- the rows of $\mathscr{M}_{\leq D}$ are not linearly independent: e.g.

$$f_k f_\ell - f_\ell f_k = 0.$$

- relations between the rows are called syzygies of the system.
- a system has trivial syzygies, and may have other: a system is regular if it has only trivial syzygies.
- F5 criterion J.-C. Faugère (2002) = a criterion to detect syzygies. Can detect all trivial syzygies.
- $\rightarrow$ construct a matrix with only Rk_D rows for regular sequences.

Linear dependencies between rows

- the rows of $\mathscr{M}_{\leq D}$ are not linearly independent: e.g.

$$f_k f_\ell - f_\ell f_k = 0.$$

- relations between the rows are called syzygies of the system.
- a system has trivial syzygies, and may have other: a system is regular if it has only trivial syzygies.
- F5 criterion J.-C. Faugère (2002) = a criterion to detect syzygies. Can detect all trivial syzygies.
- $\rightarrow$ construct a matrix with only Rk_D rows for regular sequences.

⚠ we cannot remove rows at random ⚠

Estimation of D

For regular systems:

- we can count the number of trivial syzygies, hence estimate theoretically Rk_d for any d.

Estimation of D

For regular systems:

- we can count the number of trivial syzygies, hence estimate theoretically Rk_d for any d.

If the system has 1 (resp. 0) (projective) solution:

- then D is bounded by the smallest value such that

$$\mathrm{Rk}_d = \mathrm{Mon}_d - 1 \qquad \text{(resp. } \mathrm{Rk}_d = \mathrm{Mon}_d\text{)}.$$

Hilbert Series (homogeneous system)

$I \subset R = \mathbb{F}_q[x_1, \ldots, x_n], \quad R = \oplus_d R_d, \quad I_d = R_d \cap I.$

$$HS_{R/I}(z) \stackrel{\text{def}}{=} \sum_{d \in \mathbb{N}} \dim(R_d/I_d) z^d.$$

Hilbert Series (homogeneous system)

$I \subset R = \mathbb{F}_q[x_1, \ldots, x_n], \quad R = \oplus_d R_d, \quad I_d = R_d \cap I.$

$$HS_{R/I}(z) \stackrel{\text{def}}{=} \sum_{d \in \mathbb{N}} \dim(R_d/I_d) z^d.$$

- $\dim(R_d/I_d)$ is the co-rank of the Macaulay matrix $\mathscr{M}_d = \mathrm{Mon}_d - \mathrm{Rk}_d$.

Hilbert Series (homogeneous system)

$I \subset R = \mathbb{F}_q[x_1,\ldots,x_n], \quad R = \oplus_d R_d, \quad I_d = R_d \cap I.$

$$HS_{R/I}(z) \stackrel{\text{def}}{=} \sum_{d\in\mathbb{N}} \dim(R_d/I_d) z^d.$$

- $\dim(R_d/I_d)$ is the co-rank of the Macaulay matrix $\mathscr{M}_d$ = $\mathrm{Mon}_d - \mathrm{Rk}_d$.
- Knowing all the parameters for the Macaulay matrices = knowing the Hilbert series.

Hilbert Series (homogeneous system)

$I \subset R = \mathbb{F}_q[x_1, \ldots, x_n], \quad R = \oplus_d R_d, \quad I_d = R_d \cap I.$

$$HS_{R/I}(z) \stackrel{\text{def}}{=} \sum_{d \in \mathbb{N}} \dim(R_d/I_d) z^d.$$

- $\dim(R_d/I_d)$ is the co-rank of the Macaulay matrix $\mathscr{M}_d$ = $\text{Mon}_d - \text{Rk}_d$.
- Knowing all the parameters for the Macaulay matrices = knowing the Hilbert series.
- No projective solution: $\dim(R_d/I_d) = 0$ for all $d \geq D$ ($D = \deg(HS) + 1$).

Hilbert Series (homogeneous system)

$I \subset R = \mathbb{F}_q[x_1, \ldots, x_n], \quad R = \oplus_d R_d, \quad I_d = R_d \cap I.$

$$HS_{R/I}(z) \stackrel{\text{def}}{=} \sum_{d \in \mathbb{N}} \dim(R_d/I_d) z^d.$$

- $\dim(R_d/I_d)$ is the co-rank of the Macaulay matrix $\mathscr{M}_d$ = $\text{Mon}_d - \text{Rk}_d$.
- Knowing all the parameters for the Macaulay matrices = knowing the Hilbert series.
- No projective solution: $\dim(R_d/I_d) = 0$ for all $d \geq D$ ($D = \deg(HS) + 1$).
- One projective solution: $\dim(R_d/I_d) = 1$ for all $d \geq D$.

Known classes of "regular" systems

- **regular** systems; Macaulay (1994),

(not exhaustive)

Known classes of "regular" systems

- **regular** systems; Macaulay (1994),
- **semi-regular** systems; Bardet, J.-C. Faugère, and Salvy (2004),

(not exhaustive)

- **regular** systems; Macaulay (1994),
- **semi-regular** systems; Bardet, J.-C. Faugère, and Salvy (2004),
- solutions in $\mathbb{F}_2$: **boolean semi-regular** systems; Bardet, J.-C. Faugère, Salvy, and Yang (2005),

(not exhaustive)

Known classes of "regular" systems

- **regular** systems; Macaulay (1994),
- **semi-regular** systems; Bardet, J.-C. Faugère, and Salvy (2004),
- solutions in $\mathbb{F}_2$: **boolean semi-regular** systems; Bardet, J.-C. Faugère, Salvy, and Yang (2005),
- **bi-regular bilinear** systems; J.-C. Faugère, Safey El Din, and P.-J. Spaenlehauer (2011).

(not exhaustive)

Known classes of "regular" systems

- **regular** systems; Macaulay (1994),
- **semi-regular** systems; Bardet, J.-C. Faugère, and Salvy (2004),
- solutions in $\mathbb{F}_2$: **boolean semi-regular** systems; Bardet, J.-C. Faugère, Salvy, and Yang (2005),
- **bi-regular bilinear** systems; J.-C. Faugère, Safey El Din, and P.-J. Spaenlehauer (2011).
- **determinantal** systems; Conca and Herzog (1994),

(not exhaustive)

Generic Complexity analysis

- Over an infinite field: Zariski topology, non-empty open sets are dense.

- Over an infinite field: Zariski topology, non-empty open sets are dense.
- The set of non-”regular” systems = a closed set for the Zariski topology.

Generic Complexity analysis

- Over an infinite field: Zariski topology, non-empty open sets are dense.
- The set of non-”regular” systems = a closed set for the Zariski topology.
- The set of “regular” systems = an open Zariski set.

Generic Complexity analysis

- Over an infinite field: Zariski topology, non-empty open sets are dense.
- The set of non-"regular" systems = a closed set for the Zariski topology.
- The set of "regular" systems = an open Zariski set.
- Conjecture: the open set is not empty.

Generic Complexity analysis

- Over an infinite field: Zariski topology, non-empty open sets are dense.
- The set of non-"regular" systems = a closed set for the Zariski topology.
- The set of "regular" systems = an open Zariski set.
- Conjecture: the open set is not empty.
- In practice: we take the coefficients in a finite field.

Generic Complexity analysis

- Over an infinite field: Zariski topology, non-empty open sets are dense.
- The set of non-”regular” systems = a closed set for the Zariski topology.
- The set of “regular” systems = an open Zariski set.
- Conjecture: the open set is not empty.
- In practice: we take the coefficients in a finite field.
- Conjecture: the proportion of “regular” systems is large.

Generic Complexity analysis

- Over an infinite field: Zariski topology, non-empty open sets are dense.
- The set of non-"regular" systems = a closed set for the Zariski topology.
- The set of "regular" systems = an open Zariski set.
- Conjecture: the open set is not empty.
- In practice: we take the coefficients in a finite field.
- Conjecture: the proportion of "regular" systems is large.

Generic Complexity analysis

- Over an infinite field: Zariski topology, non-empty open sets are dense.
- The set of non-"regular" systems = a closed set for the Zariski topology.
- The set of "regular" systems = an open Zariski set.
- Conjecture: the open set is not empty.
- In practice: we take the coefficients in a finite field.
- Conjecture: the proportion of "regular" systems is large.

c-ex: there is no boolean semi-regular quadratic system of 1 polynomial in $n > 6$ variables. Hodges, Molina, and Schlather (2017).
More generally, if $n \gg m$ there is no boolean semi-regular sequence of m polynomials of degree $d_1, \ldots, d_m \geq 2$.

- $m = n$ regular system: $D \leq n+1$, $\mathrm{Mon}_D = \binom{n+D-1}{D}$

Quadratic systems in different classes

- $m = n$ regular system: $D \leq n + 1$, $\mathrm{Mon}_D = \binom{n+D-1}{D}$
- $m = n + 1$ semi-regular system: $D \leq \lceil \frac{n+2}{2} \rceil$, $\rightarrow$ hybrid approach 🙂

Quadratic systems in different classes

- $m = n$ regular system: $D \leq n+1$, $\mathrm{Mon}_D = \binom{n+D-1}{D}$
- $m = n+1$ semi-regular system: $D \leq \lceil \frac{n+2}{2} \rceil$, $\rightarrow$ hybrid approach 🙂
- $m = n$ regular bilinear system with $\lfloor \frac{n}{2} \rfloor$ variables x and $\lceil \frac{n}{2} \rceil$ variables y: $D \leq \lfloor \frac{n}{2} \rfloor + 2$.

Quadratic systems in different classes

- $m = n$ regular system: $D \leq n+1$, $\mathrm{Mon}_D = \binom{n+D-1}{D}$
- $m = n+1$ semi-regular system: $D \leq \lceil \frac{n+2}{2} \rceil$, $\rightarrow$ hybrid approach 🙂
- $m = n$ regular bilinear system with $\lfloor \frac{n}{2} \rfloor$ variables x and $\lceil \frac{n}{2} \rceil$ variables y: $D \leq \lfloor \frac{n}{2} \rfloor + 2$.
- $m = 2n$ semi-regular system: $D \leq 0.0858n + o(n^{1/3})$

Quadratic systems in different classes

- $m = n$ regular system: $D \leq n+1$, $\text{Mon}_D = \binom{n+D-1}{D}$
- $m = n+1$ semi-regular system: $D \leq \lceil \frac{n+2}{2} \rceil$, $\rightarrow$ hybrid approach 🙂
- $m = n$ regular bilinear system with $\lfloor \frac{n}{2} \rfloor$ variables x and $\lceil \frac{n}{2} \rceil$ variables y: $D \leq \lfloor \frac{n}{2} \rfloor + 2$.
- $m = 2n$ semi-regular system: $D \leq 0.0858n + o(n^{1/3})$
- $m = n$ regular over $\mathbb{F}_2$: $D \leq 0.0900n + o(n^{1/3})$, but $\text{Mon}_D = \binom{n}{D}$.

Affine systems

- Apply previous results: homogenize the system ! (new variable h)

Affine systems

- Apply previous results: homogenize the system ! (new variable h)
  - it may give overestimated D^h

Affine systems

- Apply previous results: homogenize the system ! (new variable h)
  - it may give overestimated D^h
- No spurious solution at infinity ($h = 0$) if $\mathscr{F}^{top} = (f_1^{top}, \dots, f_m^{top})$ is zero-dimensional.

- Apply previous results: homogenize the system ! (new variable h)
  - it may give overestimated D^h
- No spurious solution at infinity ($h = 0$) if $\mathscr{F}^{top} = (f_1^{top}, \dots, f_m^{top})$ is zero-dimensional.
- If $\mathscr{F}^{top}$ is not regular, there are some degree drop $\rightarrow$ harder to estimate the complexity, not to compute the Gröbner basis!

Affine systems

- Apply previous results: homogenize the system ! (new variable h)
  - it may give overestimated D^h
- No spurious solution at infinity ($h = 0$) if $\mathscr{F}^{top} = (f_1^{top}, \ldots, f_m^{top})$ is zero-dimensional.
- If $\mathscr{F}^{top}$ is not regular, there are some degree drop $\rightarrow$ harder to estimate the complexity, not to compute the Gröbner basis!
- If $\mathscr{F}^{top}$ is regular: D^{top}

Affine systems

- Apply previous results: homogenize the system ! (new variable h)
  - it may give overestimated D^h
- No spurious solution at infinity ($h = 0$) if $\mathscr{F}^{top} = (f_1^{top}, \dots, f_m^{top})$ is zero-dimensional.
- If $\mathscr{F}^{top}$ is not regular, there are some degree drop $\rightarrow$ harder to estimate the complexity, not to compute the Gröbner basis!
- If $\mathscr{F}^{top}$ is regular: D^{top}
  - may need $D^{top} + 1$

Affine systems

- Apply previous results: homogenize the system ! (new variable h)
  - it may give overestimated D^h
- No spurious solution at infinity ($h = 0$) if $\mathscr{F}^{top} = (f_1^{top}, \ldots, f_m^{top})$ is zero-dimensional.
- If $\mathscr{F}^{top}$ is not regular, there are some degree drop $\rightarrow$ harder to estimate the complexity, not to compute the Gröbner basis!
- If $\mathscr{F}^{top}$ is regular: D^{top}
  - may need $D^{top} + 1$
  - may need several echelon form at degree $D^{top} + 1 \rightarrow$ complexity estimate?

Affine systems

- Apply previous results: homogenize the system ! (new variable h)
  - it may give overestimated D^h
- No spurious solution at infinity ($h = 0$) if $\mathscr{F}^{top} = (f_1^{top}, \ldots, f_m^{top})$ is zero-dimensional.
- If $\mathscr{F}^{top}$ is not regular, there are some degree drop $\to$ harder to estimate the complexity, not to compute the Gröbner basis!
- If $\mathscr{F}^{top}$ is regular: D^{top}
  - may need $D^{top} + 1$
  - may need several echelon form at degree $D^{top} + 1 \to$ complexity estimate?
- If you have degree drops: take that into account? estimate the new complexity?

Affine systems

- Apply previous results: homogenize the system ! (new variable h)
  - it may give overestimated D^h
- No spurious solution at infinity ($h = 0$) if $\mathscr{F}^{top} = (f_1^{top}, \ldots, f_m^{top})$ is zero-dimensional.
- If $\mathscr{F}^{top}$ is not regular, there are some degree drop $\to$ harder to estimate the complexity, not to compute the Gröbner basis!
- If $\mathscr{F}^{top}$ is regular: D^{top}
  - may need $D^{top} + 1$
  - may need several echelon form at degree $D^{top} + 1 \to$ complexity estimate?
- If you have degree drops: take that into account? estimate the new complexity?
  - The complexity can be smaller or larger 🤔 ⁉

3 AFFINE QUADRATIC EQUATIONS IN 2 VARIABLES, $\mathbb{F}_5$

$$\left\{\begin{array}{lllllllll} x_1^2 & & & + & 2x_1 & + & 3x_2 & + & 4, & (f_1) \\ & x_1x_2 & & & & + & 2x_2 & + & 2 (\text{or } 4), & (f_2) \\ & & x_2^2 & + & 4x_1 & + & 3x_2 & + & 2 (\text{or } 1). & (f_3) \end{array}\right.$$

- $D^{top} = 2$, not enough to get linear equations.

3 AFFINE QUADRATIC EQUATIONS IN 2 VARIABLES, $\mathbb{F}_5$

$$\begin{cases} x_1^2 + 2x_1 + 3x_2 + 4, & (f_1) \\ x_1x_2 + 2x_2 + 2 \text{(or } 4), & (f_2) \\ x_2^2 + 4x_1 + 3x_2 + 2 \text{(or } 1). & (f_3) \end{cases}$$

- $D^{top} = 2$, not enough to get linear equations.
- $D^h = 3$ (or 4)

$$\begin{cases} x_1^2 & & & + & 2x_1 & + & 3x_2 & + & 4, & (f_1) \\ & x_1x_2 & & & & + & 2x_2 & + & 2 (\text{or } 4), & (f_2) \\ & & x_2^2 & + & 4x_1 & + & 3x_2 & + & 2 (\text{or } 1). & (f_3) \end{cases}$$

- $D^{top} = 2$, not enough to get linear equations.
- $D^h = 3$ (or 4)
- $D = 3$ gives

$$\begin{cases} \vdots \\ x_2^2 + 4, \\ x_1 + 4, \\ x_2 + 4. \end{cases} \quad \text{or} \quad \begin{cases} \vdots \\ x_2^2 + 2, \\ x_1 + 3, \\ x_2 + 4. \end{cases}$$

3 AFFINE QUADRATIC EQUATIONS IN 2 VARIABLES, $\mathbb{F}_5$

$$\begin{cases} x_1^2 & & & + 2x_1 & + 3x_2 & + 4, & (f_1) \\ & x_1x_2 & & & + 2x_2 & + 2 \text{ (or } 4), & (f_2) \\ & & x_2^2 & + 4x_1 & + 3x_2 & + 2 \text{ (or } 1). & (f_3) \end{cases}$$

- $D^{top} = 2$, not enough to get linear equations.
- $D^h = 3$ (or 4)
- $D = 3$ gives

$$\begin{cases} \vdots \\ x_2^2 + 4, \\ x_1 + 4, \\ x_2 + 4. \end{cases} \quad \text{or} \quad \begin{cases} \vdots \\ x_2^2 + 2, \\ x_1 + 3, \\ x_2 + 4. \end{cases}$$

- second case: need another $D = 2$ matrix to get $I = \langle 1 \rangle$.

For a class of system coming from an algebraic modeling

- determine the generic relations between rows in the Macaulay matrices = syzygies,

For a class of system coming from an algebraic modeling

- determine the generic relations between rows in the Macaulay matrices = syzygies,
- compute the rank of the Macaulay matrices for generic systems,

For a class of system coming from an algebraic modeling

- determine the generic relations between rows in the Macaulay matrices = syzygies,
- compute the rank of the Macaulay matrices for generic systems,
- deduce the maximal degree $D \rightarrow$ complexity estimates,

For a class of system coming from an algebraic modeling

- determine the generic relations between rows in the Macaulay matrices = syzygies,
- compute the rank of the Macaulay matrices for generic systems,
- deduce the maximal degree $D \rightarrow$ complexity estimates,
- design a specific Gb algorithm that is more efficient.

LET'S PLAY A GAME

Some important parameters to estimate the complexity of solving a polynomial system:

- the number of variables,
- the number of equations,
- the degree of the equations,
- the degree of the intermediate computations

Let's play a game

Some important parameters to estimate the complexity of solving a polynomial system:

- the number of variables,
- the number of equations,
- the degree of the equations,
- the degree of the intermediate computations

But not sufficient!

LET'S PLAY A GAME

Some important parameters to estimate the complexity of solving a polynomial system:

- the number of variables,
- the number of equations,
- the degree of the equations,
- the degree of the intermediate computations

But not sufficient!
Given a polynomial system of equations, what can you say "a priori" about its complexity?

Complexity of solving a system

$$\begin{cases} x_1 + 2x_5 + 2x_6 + 1, \\ x_1 + x_5 + x_6 + 2, \\ x_1 + 2x_2 + 2x_3 + 2x_4 + x_6 + 1, \\ x_1 + x_2 + x_4 + 2x_5 + x_6 + 1, \\ x_1 + x_2 + 2x_3 + x_4 + x_5 + x_6, \\ 2x_1 + 2x_2 + x_3 + x_4 + x_5 + 1 \end{cases}$$

Complexity of solving a system

$$\begin{cases} x_1 + 2x_5 + 2x_6 + 1, \\ x_1 + x_5 + x_6 + 2, \\ x_1 + 2x_2 + 2x_3 + 2x_4 + x_6 + 1, \\ x_1 + x_2 + x_4 + 2x_5 + x_6 + 1, \\ x_1 + x_2 + 2x_3 + x_4 + x_5 + x_6, \\ 2x_1 + 2x_2 + x_3 + x_4 + x_5 + 1 \end{cases}$$

Linear system, polynomial time complexity.

Complexity of solving a system

$$\begin{cases} x_1 + 2x_5 + 2x_6 + 1, \\ x_1 + x_5 + x_6 + 2, \\ x_1 + 2x_2 + 2x_3 + 2x_4 + x_6 + 1, \\ x_1 + x_2 + x_4 + 2x_5 + x_6 + 1, \\ x_1 + x_2 + 2x_3 + x_4 + x_5 + x_6, \\ 2x_1 + 2x_2 + x_3 + x_4 + x_5 + 1 \end{cases}$$

Linear system, polynomial time complexity.
Number of solutions? ($\mathbb{F}_3$)

Example (Bayer-Stillman 1988)

$$\mathscr{S}_{ex} = \begin{cases} f_0 c_{0,\ell} b_{0,\ell}^2 + s_0 c_{0,\ell}, & \\ s_i c_{i,1} + s_{i+1}, & \\ s_i c_{i,4} + f_{i+1}, & \\ f_i c_{i,1} + s_i c_{i,2}, & i \in \{0..2\} \\ s_i c_{i,3} + f_i c_{i,4}, & \ell \in \{1..4\} \\ f_i c_{i,2} b_{i,1} + f_i c_{i,3} b_{i,4}, & \\ s_i c_{i,2} + s_i c_{i,3}, & \\ f_i c_{i,2} b_{i,3} c_{i+1,\ell} b_{i+1,\ell} + f_i c_{i,\ell} c_{i,2} b_{i,2}, & \end{cases}$$

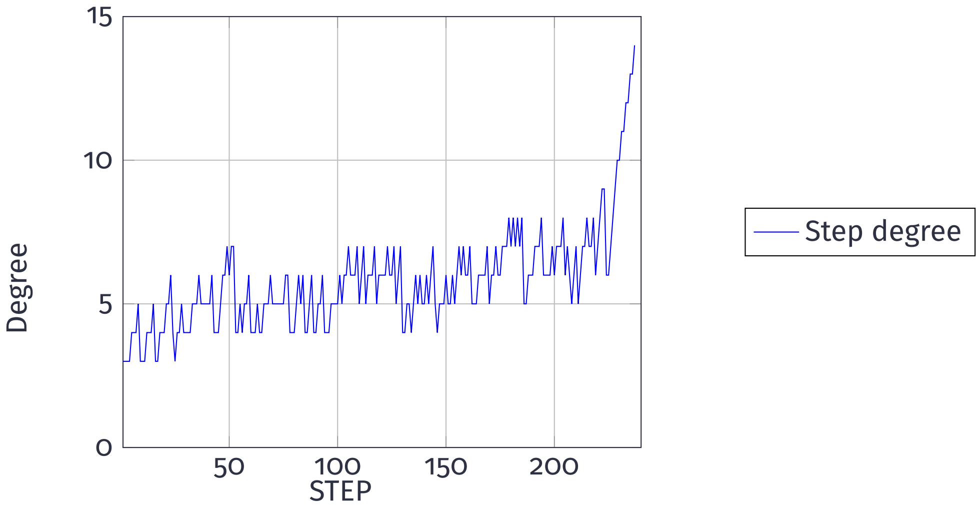
$\mathscr{S}_{ex} \in \mathbb{F}_2[f_i, s_i, c_{i,\ell}, b_{i,\ell}]$ for $i \in \{0..3\}, \ell \in \{1..4\}$.
40 variables, 34 polynomials of degrees `2:15, 3:3, 4:4, 5:12`.

Example (Bayer-Stillman 1988)

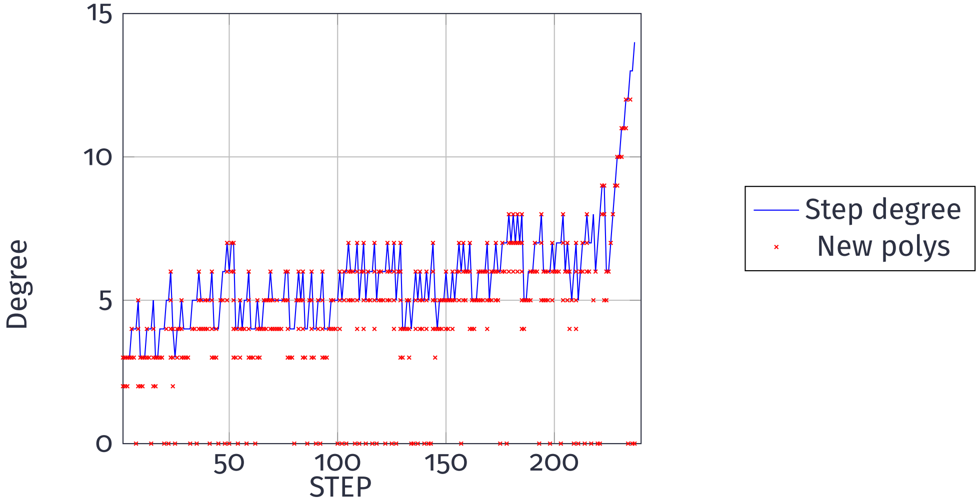
$$\mathscr{S}_{ex} = \begin{cases} f_0 c_{0,\ell} b_{0,\ell}^2 + s_0 c_{0,\ell}, & \\ s_i c_{i,1} + s_{i+1}, & \\ s_i c_{i,4} + f_{i+1}, & \\ f_i c_{i,1} + s_i c_{i,2}, & i \in \{0..2\} \\ s_i c_{i,3} + f_i c_{i,4}, & \ell \in \{1..4\} \\ f_i c_{i,2} b_{i,1} + f_i c_{i,3} b_{i,4}, & \\ s_i c_{i,2} + s_i c_{i,3}, & \\ f_i c_{i,2} b_{i,3} c_{i+1,\ell} b_{i+1,\ell} + f_i c_{i,\ell} c_{i,2} b_{i,2}, & \end{cases}$$

$\mathscr{S}_{ex} \in \mathbb{F}_2[f_i, s_i, c_{i,\ell}, b_{i,\ell}]$ for $i \in \{0..3\}, \ell \in \{1..4\}$.
40 variables, 34 polynomials of degrees `2:15, 3:3, 4:4, 5:12`.
$D = 82$ for regular systems

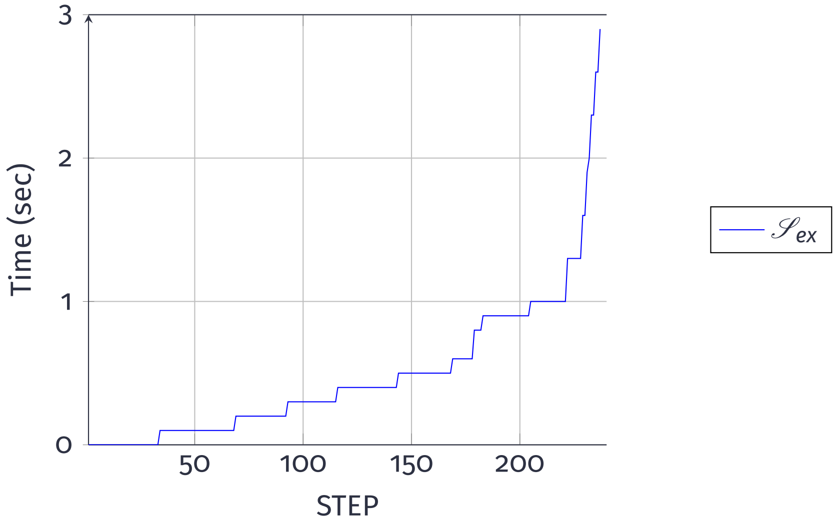
Step Degrees during the grevlex computation for $\mathscr{S}_{ex}$ (`magma V2.28-2`)



Step Degrees during the grevlex computation for $\mathscr{S}_{ex}$ (`magma V2.28-2`)



Time of the computation (in sec) for $\mathscr{S}_{ex}$ (`magma V2.28-2`)



Example (Bayer-Stillman 1988)

$$\mathscr{S}_{ex} = \begin{cases} f_0 c_{0,\ell} b_{0,\ell}^2 + s_0 c_{0,\ell}, & \\ s_i c_{i,1} + s_{i+1}, & \\ s_i c_{i,4} + f_{i+1}, & \\ f_i c_{i,1} + s_i c_{i,2}, & i \in \{0..2\} \\ s_i c_{i,3} + f_i c_{i,4}, & \ell \in \{1..4\} \\ f_i c_{i,2} b_{i,1} + f_i c_{i,3} b_{i,4}, & \\ s_i c_{i,2} + s_i c_{i,3}, & \\ f_i c_{i,2} b_{i,3} c_{i+1,\ell} b_{i+1,\ell} + f_i c_{i \quad ,\ell} c_{i,2} b_{i,2}, & \end{cases}$$

$\mathscr{S}_{ex} \in \mathbb{F}_2[f_i, s_i, c_{i,\ell}, b_{i,\ell}]$ for $i \in \{0..3\}, \ell \in \{1..4\}$.

40 variables, 34 polynomials of degrees `2:15, 3:3, 4:4, 5:12`.

$\mathscr{S}_{ex}$ solved in 3.3 seconds.

Example (Bayer-Stillman 1988)

$$\mathscr{S}_{ex} = \begin{cases} f_0 c_{0,\ell} b_{0,\ell}^2 + s_0 c_{0,\ell}, & \\ s_i c_{i,1} + s_{i+1}, & \\ s_i c_{i,4} + f_{i+1}, & \\ f_i c_{i,1} + s_i c_{i,2}, & i \in \{0..2\} \\ s_i c_{i,3} + f_i c_{i,4}, & \ell \in \{1..4\} \\ f_i c_{i,2} b_{i,1} + f_i c_{i,3} b_{i,4}, & \\ s_i c_{i,2} + s_i c_{i,3}, & \\ f_i c_{i,2} b_{i,3} c_{i+1,\ell} b_{i+1,\ell} + f_i c_{i+1,\ell} c_{i,2} b_{i,2}, & \end{cases}$$

$\mathscr{S}_{ex} \in \mathbb{F}_2[f_i, s_i, c_{i,\ell}, b_{i,\ell}]$ for $i \in \{0..3\}, \ell \in \{1..4\}$.
40 variables, 34 polynomials of degrees `2:15, 3:3, 4:4, 5:12`.
$\mathscr{S}_{ex}$ solved in seconds.

Example (Bayer-Stillman 1988)

$$\mathscr{S}_{bs} = \begin{cases} f_0 c_{0,\ell} b_{0,\ell}^2 + s_0 c_{0,\ell}, & \\ s_i c_{i,1} + s_{i+1}, & \\ s_i c_{i,4} + f_{i+1}, & \\ f_i c_{i,1} + s_i c_{i,2}, & i \in \{0..2\} \\ s_i c_{i,3} + f_i c_{i,4}, & \ell \in \{1..4\} \\ f_i c_{i,2} b_{i,1} + f_i c_{i,3} b_{i,4}, & \\ s_i c_{i,2} + s_i c_{i,3}, & \\ f_i c_{i,2} b_{i,3} c_{i+1,\ell} b_{i+1,\ell} + f_i c_{i+1,\ell} c_{i,2} b_{i,2}, & \end{cases}$$

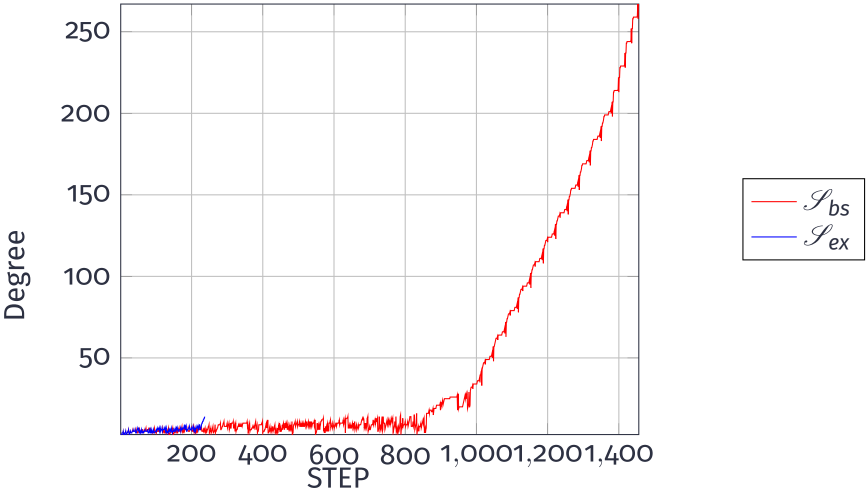
$\mathscr{S}_{bs} \in \mathbb{F}_2[f_i, s_i, c_{i,\ell}, b_{i,\ell}]$ for $i \in \{0..3\}, \ell \in \{1..4\}$.
40 variables, 34 polynomials of degrees `2:15, 3:3, 4:4, 5:12`. $D = 82$? 🤔
$\mathscr{S}_{bs}$ solved in seconds.

Example (Bayer-Stillman 1988)

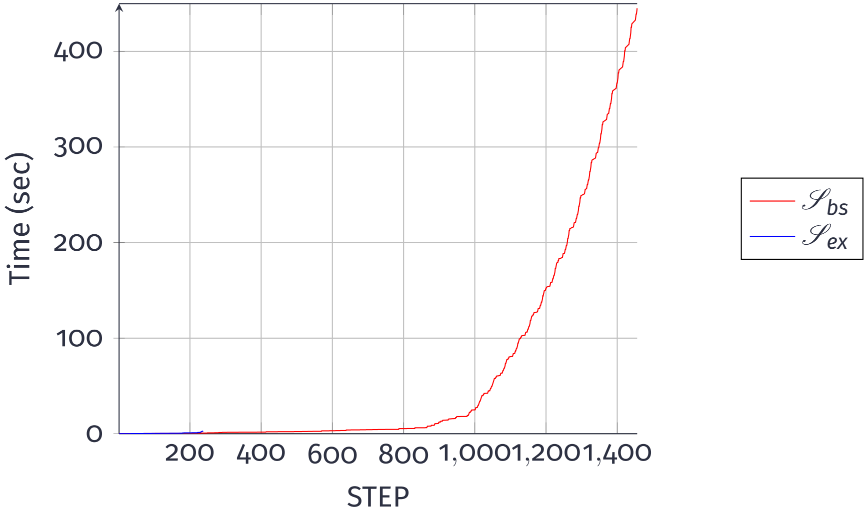
$$\mathscr{S}_{bs} = \begin{cases} f_0 c_{0,\ell} b_{0,\ell}^2 + s_0 c_{0,\ell}, \\ s_i c_{i,1} + s_{i+1}, \\ s_i c_{i,4} + f_{i+1}, \\ f_i c_{i,1} + s_i c_{i,2}, & i \in \{0..2\} \\ s_i c_{i,3} + f_i c_{i,4}, & \ell \in \{1..4\} \\ f_i c_{i,2} b_{i,1} + f_i c_{i,3} b_{i,4}, \\ s_i c_{i,2} + s_i c_{i,3}, \\ f_i c_{i,2} b_{i,3} c_{i+1,\ell} b_{i+1,\ell} + f_i c_{i+1,\ell} c_{i,2} b_{i,2}, \end{cases}$$

$\mathscr{S}_{bs} \in \mathbb{F}_2[f_i, s_i, c_{i,\ell}, b_{i,\ell}]$ for $i \in \{0..3\}, \ell \in \{1..4\}$.
40 variables, 34 polynomials of degrees `2:15, 3:3, 4:4, 5:12`. $D = 82$? 🤔
$\mathscr{S}_{bs}$ solved in 448.5 seconds.

Step Degrees during the computation for $\mathscr{S}_{bs}$ and $\mathscr{S}_{ex}$ (`magma V2.28-2`)



Time of the computation (in sec) for $\mathscr{S}_{bs}$ and $\mathscr{S}_{ex}$ (`magma V2.28-2`)



Bayer and Stillman (1988) example

- parameter m,
- $10m+4$ equations (degrees `2:5m, 3:m, 4:4, 5:4m`),
- $10(m+1)$ variables.
- the Gröbner basis contains polynomials of degree $2^{2^m}+2$.
- the example was $m=3$: maximal degree $2^{2^3}+2=258$.

EX vs BS example $m = 4$

- 703 STEPS vs > 40770
- max degree 14 vs 65538
- time 27.5 sec vs > 1131 seconds (segfault...)

80 QUADRATIC EQUATIONS 80 VARIABLES IN $\mathbb{F}_{16}$

80 QUADRATIC EQUATIONS 80 VARIABLES IN $\mathbb{F}_{16}$

- regular?

80 quadratic equations 80 variables in $\mathbb{F}_{16}$

- regular? yes!

80 QUADRATIC EQUATIONS 80 VARIABLES IN $\mathbb{F}_{16}$

- regular? yes!
- Complexity?

80 QUADRATIC EQUATIONS 80 VARIABLES IN $\mathbb{F}_{16}$

- regular? yes!
- Complexity? $D = 81$, $\mathrm{Mon}_{81} = 2^{156}$

80 QUADRATIC EQUATIONS 80 VARIABLES IN $\mathbb{F}_{16}$

- regular? yes!
- Complexity? $D = 81$, $\mathrm{Mon}_{81} = 2^{156}$
- my system:

$$\begin{cases} x_1^2, \\ x_2^2, \\ \vdots \\ x_{80}^2. \end{cases}$$

<beamer>

Aguilar Melchor, Carlos, Nicolas Aragon, Slim Bettaieb, et al. (Apr. 2019). *Rank Quasi Cyclic (RQC)*. Second round submission to the NIST post-quantum cryptography call.

Aragon, N., P. Gaborit, A. Hauteville, et al. (2019). “Low Rank Parity Check Codes: New Decoding Algorithms and Application to Cryptography”. In: submitted to IEEE IT, preprint available on arXiv.

Aragon, Nicolas, Olivier Blazy, Jean-Christophe Deneuville, et al. (Mar. 2019). *ROLLO (merger of Rank-Ouroboros, LAKE and LOCKER)*. Second round submission to the NIST post-quantum cryptography call. NIST Round 2 submission for Post-Quantum Cryptography.

Baena, John, Pierre Briaud, Daniel Cabarcas, et al. (2022). “Improving Support-Minors Rank Attacks: Applications to GeMSS and Rainbow”. In: *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part III*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13509. LNCS. Springer, pp. 376–405.

Bardet, Magali and Manon Bertin (Sept. 2022). “Improvement of Algebraic Attacks for Solving Superdetermined MinRank Instances”. In: *Post-Quantum Cryptography 2022*. Ed. by Jung Hee Cheon and Thomas Johansson. Vol. 13512. LNCS. Springer International Publishing: Cham, pp. 107–123.

Bardet, Magali, Pierre Briaud, Maxime Bros, et al. (2023). “Revisiting Algebraic Attacks on MinRank and on the Rank Decoding Problem”. In: *Designs, Codes and Cryptography* 91, pp. 3671–3707.

Bardet, Magali, Maxime Bros, Daniel Cabarcas, et al. (2020). “Improvements of Algebraic Attacks for solving the Rank Decoding and MinRank problems”. In: *Advances in Cryptology - ASIACRYPT 2020, International Conference on the Theory and Application of Cryptology and Information Security, 2020. Proceedings*. Vol. 12491. LNCS, pp. 507–536.

Bardet, Magali, Jean-Charles Faugère, and Bruno Salvy (2004). “On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations”. In: *Proceedings of the International Conference on Polynomial System Solving ICPSS'04*, pp. 71–74.

Bardet, Magali, Jean-Charles Faugère, Bruno Salvy, and Bo-Yin Yang (2005). “Asymptotic expansion of the degree of regularity for semi-regular systems of equations”. In: *MEGA'05 – Effective Methods in Algebraic Geometry*, pp. 1–14.

Bayer, David and Michael Stillman (1988). “On the complexity of computing syzygies”. In: *Journal of Symbolic Computation* 6(2-3), pp. 135–147.

Bettale, Luk, Jean-Charles Faugere, and Ludovic Perret (2009). “Hybrid approach for solving multivariate systems over finite fields”. In: *Journal of Mathematical Cryptology* 3(3), pp. 177–197.

Buchberger, Bruno (1965). “Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal”. PhD thesis. Universitat Innsbruck.

Burle, Étienne, Philippe Gaborit, Younes Hatri, and Ayoub Otmani (2023). *Injective Rank Metric Trapdoor Functions with Homogeneous Errors*. arXiv: `2310.08962 [cs.CR]`.

Casanova, Antoine, Jean-Charles Faugère, Gilles Macario-Rat, et al. (Apr. 2019). *GeMSS: A Great Multivariate Short Signature*. Second round submission to the NIST post-quantum cryptography call.

Conca, Aldo and Jurgen Herzog (1994). “On the Hilbert function of determinantal rings and their canonical module”. In: *Proc. Amer. Math. Soc* 122, pp. 677–681.

Delsarte, Philippe (1978). “Bilinear Forms over a Finite Field, with Applications to Coding Theory”. In: *J. Comb. Theory, Ser. A* 25(3), pp. 226–241.

Faugère, Jean-Charles (1999). “A New Efficient Algorithm for Computing Gröbner Bases (F4)”. In: *J. Pure Appl. Algebra* 139(1-3), pp. 61–88.

Faugère, Jean-Charles (2002). “A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5)”. English. In: *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*. Ed. by Teo Mora. ACM Press: New York, 75–83 (electronic).

Faugère, Jean-Charles, Pierrick Gaudry, Louise Huot, and Guénaël Renault (2014). “Sub-Cubic Change of Ordering for GröBner Basis: A Probabilistic Approach”. In: *ISSAC*.

Faugère, Jean-Charles, Patrizia Gianni, Daniel Lazard, and Teo Mora (1993). “Efficient Computation of Zero-Dimensional Gröbner Bases by Change of Ordering”. In: *J. Symbolic Comput.* 16(4), pp. 329–344.

Faugère, Jean-Charles, Françoise Levy-dit-Vehel, and Ludovic Perret (2008). “Cryptanalysis of Minrank”. In: *Advances in Cryptology - CRYPTO 2008*. Ed. by David Wagner. Vol. 5157. LNCS, pp. 280–296.

- Faugère, Jean-Charles, Mohab Safey El Din, and Pierre-Jean Spaenlehauer (2010). "Computing loci of rank defects of linear matrices using Gröbner bases and applications to cryptology". In: *International Symposium on Symbolic and Algebraic Computation, ISSAC 2010, Munich, Germany, July 25-28, 2010*, pp. 257–264.
- Faugère, Jean-Charles, Mohab Safey El Din, and Pierre-Jean Spaenlehauer (2011). "Gröbner bases of bihomogeneous ideals generated by polynomials of bidegree (1,1): Algorithms and complexity". In: *J. Symbolic Comput.* 46(4), pp. 406–437.
- Fraenkel, A.S. and Y. Yesha (1979). "Complexity of problems in games, graphs and algebraic equations". In: *Discrete Applied Mathematics* 1(1), pp. 15–30.
- Gabidulin, Ernst M. (1985). "Theory of codes with maximum rank distance". In: *Problemy Peredachi Informatsii* 21(1), pp. 3–16.
- Gabidulin, Ernst M., A. V. Paramonov, and O. V. Tretjakov (Apr. 1991). "Ideals over a non-commutative ring and their applications to cryptography". In: *Advances in Cryptology - EUROCRYPT'91*. LNCS 547. Brighton, pp. 482–489.

Gaborit, Philippe, Adrien Hauteville, Duong Hieu Phan, and Jean-Pierre Tillich (May 2016). *Identity-based Encryption from Rank Metric*. IACR Cryptology ePrint Archive, Report2017/623. `http://eprint.iacr.org/`.

Gaborit, Philippe and Gilles Zémor (2016). "On the hardness of the decoding and the minimum distance problems for rank codes". In: *IEEE Trans. Inform. Theory* 62(12), pp. 7245–7252.

Giusti, M. (1984). "Some effectivity problems in polynomial ideal theory". In: *Eurosam 84*. Ed. by John Fitch. Vol. 174. Lecture Notes in Computer Science. Cambridge, 1984. Springer Berlin / Heidelberg: Berlin, pp. 159–171.

Guo, Hao and Jintai Ding (2022). "Algebraic Relation of Three MinRank Algebraic Modelings". In: *Arithmetic of Finite Fields*. LNCS. Springer.

Hodges, Timothy J., Sergio D. Molina, and Jacob Schlather (2017). "On the existence of homogeneous semi-regular sequences in $F_2[X_1,\dots,X_n]/(X_1^2,\dots,X_n^2)$". In: *Journal of Algebra* 476, pp. 519–547.

Jeannerod, Claude-Pierre, Clément Pernet, and Arne Storjohann (2013). "Rank-profile revealing Gaussian elimination and the CUP matrix decomposition". In: *Journal of Symbolic Computation* 56, pp. 46–68.

Kipnis, Aviad and Adi Shamir (Aug. 1999). “Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization”. In: *Advances in Cryptology - CRYPTO’99*. Vol. 1666. LNCS. Springer: Santa Barbara, California, USA, pp. 19–30.

Lazard, D. (1983). “Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations”. In: *Computer algebra*. Vol. 162. LNCS. Proceedings Eurocal’83, London, 1983. Springer: Berlin, pp. 146–156.

Macaulay, Francis Sowerby (1902). “Some formulae in elimination”. In: *Proceedings of the London Mathematical Society* 1(1), pp. 3–27.

Macaulay, Francis Sowerby (1994). *The algebraic theory of modular systems*. Vol. 19. Cambridge University Press.

Ourivski, Alexei V. and Thomas Johansson (2002). “New Technique for Decoding Codes in the Rank Metric and Its Cryptography Applications”. English. In: *Problems of Information Transmission* 38(3), pp. 237–246.

Overbeck, Raphael (2005). “A New Structural Attack for GPT and Variants”. In: *Mycrypt*. Vol. 3715. LNCS, pp. 50–63.

- Tao, Chengdong, Albrecht Petzoldt, and Jintai Ding (2021). "Efficient Key Recovery for All HFE Signature Variants". In: *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*. Ed. by Tal Malkin and Chris Peikert. Vol. 12825. Lecture Notes in Computer Science. Springer, pp. 70–93.
- Wiedemann, Douglas (1986). "Solving sparse linear equations over finite fields". In: *IEEE transactions on information theory* 32(1), pp. 54–62.