IACR Communications in Cryptology
ISSN 3006-5496, Vol. 1, No. 3, 27 pages.

https://doi.org/10.62056/ak86cy7qiu

An analysis of the Crossbred Algorithm for the MQ Problem

Damien Vidal, Claire Delaplace and Sorina Ionica

Laboratoire MIS, Université de Picardie Jules Verne, Amiens, France

Abstract. The Crossbred algorithm is currently the state-of-the-art method for solving overdetermined multivariate polynomial systems over $\mathbb{F}_2$. Since its publication in 2017, several record breaking implementations have been proposed and demonstrate the power of this hybrid approach. Despite these practical results, the complexity of this algorithm and the choice of optimal parameters for it are difficult open questions. In this paper, we prove a bivariate generating series for potentially admissible parameters of the Crossbred algorithm.

Keywords: Gröbner basis · polynomial system · MQ problem · exhaustive search · Crossbred

1 Introduction

Solving a given polynomial system of m polynomials and n variables over a finite field $\mathbb{F}_p$ is proved to be NP-complete [FY79]. In this paper, we focus on the *Multivariate Quadratic Problem* (*MQ*), which means that we consider polynomials of degree 2.

The interest of cryptographers for the MQ problem can be traced back as early as 1988 with a public key encryption scheme due to Matsumoto and Imai [MI88]. Although this scheme was broken by Patarin [Pat95] in 1995, it lead the way to several other to be developed. Recently, with the growing interest for post-quantum cryptography, MQ-based cryptography has seen the emergence of many new schemes including UOV [KPG99] and its variants [CFF$^+$23a, WTKC22, BP17, DS05], as well as schemes derived from zero-knowledge proofs using the Fiat-Shamir transform (e.g., [CHR$^+$19]). It may be worth mentioning that four out of the nine signature schemes present in the second round of the NIST competition for post-quantum cryptography were multivariate ones [BP17, DS05, CFMR$^+$17, CHR$^+$19]. Note that all of these schemes were later broken for the parameters given in the specifications [MIS20, Beu22, TPD21, KZ20]. However, it does not mean that multivariate cryptography is dead yet. In the recent call for additional signatures, ten new Multivariate schemes were proposed including [JBH$^+$23, CFF$^+$23b, FHI$^+$23]. Moreover, some of the MPC-in-the-head schemes submitted to this call are also based on the MQ problem [BPKV24, BFR24].

Not all multivariate quadratic systems are hard to solve since the behaviour of algorithms allowing to solve them depends on the relative values of m and n. If $m \geq n(n-1)/2$ or if $n \geq m(m+1)$, it is possible to solve a polynomial systems with these parameters in polynomial time [TW12]. Most often in cryptography, we are confronted to solving polynomial systems for which $n^2 > m \geq n$. Commonly used methods to solve these systems are algorithms for computing Gröbner basis: Buchberger's algorithm [Buc65] or linear algebra-based algorithms (F_4 [Fau99], F_5 [Fau02], XL [CKPS00]). In the case of small finite fields (for example $\mathbb{F}_2$, $\mathbb{F}_3$ or $\mathbb{F}_5$), exhaustive search becomes a viable way to

E-mail: damien.vidal@u-picardie.fr (Damien Vidal), claire.delaplace@u-picardie.fr (Claire Delaplace), sorina.ionica@u-picardie.fr (Sorina Ionica)

Received: 2024-07-08 Accepted: 2024-09-02

solve a polynomial system (FES [BCC$^+$10]). It is also used to assign certain variables before running the linear algebra-based algorithm for solving the system (FXL [CKPS00], BooleanSolve [BFSS13], Crossbred [JV17]). In particular, we are interested in the case where the polynomial system is defined over $\mathbb{F}_2$.

The complexity analysis of the Crossbred algorithm is not clear, but the authors of the algorithm claim it to be similar to that of FXL [CKPS00] or BooleanSolve [BFSS13], without giving a proof. However, Joux and Vitse's original implementation as well as more recent open-source implementations [NNY18, NNY17, BS23] were used to break records several times on overdetermined systems coming from the Fukuoka Type I MQ challenge [Yas15]. For a polynomial system $\mathcal{F}$, the running time of the algorithm heavily depends on three input parameters: D, d and k. In a pre-processing step based on linear algebra on the degree D Macaulay matrix, the algorithm constructs r polynomials of total degree D and of degree d in the first k variables. After that, the last $n-k$ variables are assigned in $\mathcal{F}$ and the degree d Macaulay matrix is computed for the system obtained in this way. After specialization, the new polynomials obtained in the pre-processing step are also appended as rows in this matrix. If the resulting degree d system in the first k variables may be solved (for instance by linearization), then we are done.

A set of parameters D, d and k are called admissible if the degree d system obtained after specialization can be solved with echelonization. From a theoretical point of view, it is not easy to determine admissible parameters for the algorithm, let alone optimal choices. To the best of our knowledge, all existing implementations over $\mathbb{F}_2$ have focused on $D \leq 5$ and $d = 1$, which means that the system obtained after specialization is linear. From a practical point of view, it is difficult to handle higher values of D and d, since as soon as $D \geq 4$ lots of linear dependencies start to appear in the Macaulay matrix and the matrices are large enough that it is no longer possible to construct them due to memory issues. This is a common problem when implementing Gröbner basis algorithms, most of the computation time is lost in useless operations. For example, in the F5 algorithm, Faugère [Fau02] and later Bardet in her PhD thesis [Bar04], proposed two criteria to remove all linear dependencies, for regular and semi-regular polynomial systems, respectively.

Contributions and related work. In this paper, we first restate these criteria for the special form of Macaulay matrices of degree D appearing in the Crossbred algorithm and show that we can remove linear dependencies after specialization. Secondly, we propose a simplified variant of the algorithm, called Block Crossbred, which may be seen as a homogeneous variant of the algorithm, before specialization. We compute a bivariate generating series whose coefficients correspond to the number of newly generated polynomials in the pre-processing step of the algorithm for input parameters D, d and k, under semi-regularity assumptions. From this analysis, we deduce the generating series for the Crossbred algorithm, again under a semi-regularity hypothesis. We conclude by showing sets of admissible parameters for Crossbred obtained by looking at the coefficients of the series. Finally, we implemented in Magma and ran the Crossbred algorithm on the smallest sets of these parameters, to empirically verify our theoretical findings.

Recently, there have been several attempts to study the complexity of Crossbred. First Duarte [Dua23] computes a bivariate generating series for the pre-processing step of the algorithm. The author introduces the notion of semi-regularity for homogeneous systems of polynomials, but this notion is not used anywhere in his proof. On one hand, the polynomials appearing in the rows of the Macaulay matrices in Crossbred are affine, and on the other hand the proofs do not keep track of reductions to zero in the algorithm.

Nakamura [Nak24] proposes an analogue series for systems defined over finite fields of odd characteristic, however these results are based on unverified assumptions. Finally, the recent preprint [BCT$^+$24] revisits the notion of semi-regularity and focuses on admissible parameters for Crossbred under semi-regularity assumptions, provided that the bivariate generating series conjectured in the literature is correct. We stress here that the correct assumption for Crossbred, which is a hybrid algorithm, is that of strong semi-regularity. Roughly speaking, this means that after assigning $n-k$ variables in the initial system, the derived system in k variables is semi-regular, for almost all assignments. Theorem 1 in [Nak24] and Theorem 2 in [BCT$^+$24] came close to this idea, but the authors focus on a fixed assignment instead of looking at all possible assignments.

Our work is completely independent from [Dua23, Nak24, BCT$^+$24], but we certainly do not claim originality for this approach. Most of the techniques used in this paper are standard in the literature (see [Fau02, Bar04]) and we adapted them to the case of Crossbred.

This paper is organised as follows. In Section 2 we introduce the notion of semi-regular sequences of polynomials and briefly survey linear algebra based algorithms and the Crossbred algorithm. From that, we state our criteria for reduction to zero and present the Block Crossbred algorithm in Section 3. In Section 4 we show that there are no reductions to zero in the Block Crossbred algorithm if the two criteria are used. Based on this result, we compute the generating series of our algorithm in Section 5. Finally, under semi-regularity assumptions, we deduce a proof for the bivariate generating series of the original Crossbred in Section 6. We apply our results to compute admissible parameters for Crossbred in Section 7.

Acknowledgment. We are grateful to the anonymous reviewers for their helpful comments, which helped us improve on a preliminary version of this paper. The authors thank Charles Bouillaguet, Julia Sauvage and Momonari Kudo for helpful discussions. This work was partially funded by the French Agence Nationale de la Recherche under the projects Postcryptum ANR20-ASTR-0011 and CRYPTANALYSE 22-PECY-0010. The experiments were carried on the MatriCS platform of the Université de Picardie Jules Verne.

2 Notation and Background

In this section, we will introduce the notation and terminology used throughout the paper. We will use the polynomial ring $R = \mathbb{F}_p[x_1, \ldots, x_n]$, where $\mathbb{F}_p$ is any finite field. We write $x^b = x_1^{b_1} x_2^{b_2} \cdots x_n^{b_n}$ with $b = (b_1, \ldots, b_n)$. Then $|b| = \sum_{i=1}^{n} b_i$ is the degree of the monomial (also called total degree) and is written $\deg(x^b)$. We denote by $\deg_k$ the degree over the first k variables (i.e. $\deg_k(x^b) = \sum_{i=1}^{k} b_i$). We choose an admissible monomial ordering on R. For instance, we will need the glex order with $x_1 > x_2 > \ldots > x_n$ which is defined as $x^a >_{glex} x^b$ if $|a| > |b|$ or $|a| = |b|$ and the leftmost non-zero coefficient of $a - b$ is positive. The leading term of a polynomial f with respect to the chosen order is denoted by $LT(f)$. The total degree and the degree over k of f are the total degree and the degree over the first k variables, respectively, of its leading term $LT(f)$ with respect to the chosen order. Macaulay matrices, initially introduced by Lazard [Laz83], are at the heart of all linear algebra-based algorithms for computing Gröbner basis. The Macaulay matrix is defined as follows.

Definition 1. Fix an admissible monomial ordering on R. Given a homogeneous (affine) system of polynomials $\mathcal{F} = \{f_1, \ldots, f_m\}$ in R, we associate to it the Macaulay matrix of degree D (resp. $\leq D$), denoted by $Mac_{D,m}(\mathcal{F})$ (resp. $Mac_{\leq D,m}(\mathcal{F})$) and defined as follows: the columns of $Mac_{D,m}(\mathcal{F})$ (resp. $Mac_{\leq D,m}(\mathcal{F})$) are indexed by the monomials in $\mathbb{F}_p[x_1, \ldots, x_n]$ of degree D (resp. of degree $\leq D$), sorted in decreasing order from left to the right following the chosen order. Each row in this matrix is labeled by a tag (u, f_i), where u is a monomial in $\mathbb{F}_p[x_1, \ldots, x_n]$ and $f_i \in \mathcal{F}$ such that $deg(uf_i) = D$ (resp. $deg(uf_i) \leq D$), and contains the polynomial uf_i written as a vector of coefficients of monomials.

Example 2.1. Consider the polynomial system $\mathcal{F} = \{f_1, f_2\}$ with $f_1, f_2 \in \mathbb{F}_2[x_1, x_2, x_3]$ given by:

$$\begin{aligned} f_1 &= x_1x_3 + x_2x_3 + x_2 + 1, \\ f_2 &= x_1x_2 + x_1 + x_3 + 1. \end{aligned}$$

Since the goal is to compute roots of this polynomial system in $\mathbb{F}_2$, we add the polynomials $x_1^2 - x_1$, $x_2^2 - x_2$ and $x_3^2 - x_3$ to this system. This is equivalent to constructing the Macaulay matrix in $\mathbb{F}_2[x_1, x_2, x_3]/ < x_1^2 - x_1, x_2^2 - x_2, x_3^2 - x_3 >$. Then the corresponding Macaulay matrix of degree 3 for the glex order is:

$$Mac_{\leq 3,2}(\mathcal{F}) = \begin{array}{c|cccccccc|} & x_1x_2x_3 & x_1x_2 & x_1x_3 & x_2x_3 & x_1 & x_2 & x_3 & 1 \\ f_1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ f_2 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ x_1f_1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ x_2f_1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ x_3f_1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ x_1f_2 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ x_2f_2 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ x_3f_2 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}$$

2.1 Linear algebra-based Gröbner basis algorithms and their complexity

Let us fix a monomial ordering on R and denote by I the ideal in R generated by a sequence of polynomials $\mathcal{F}$. We denote by $LT(I)$ the set of all leading terms of non-zero polynomials in I. A Gröbner basis for I is a finite set of generators $G = \{g_1, \dots g_s\}$ such that the monomial ideal generated by elements of $LT(I)$ is given by:

$$\langle LT(I) \rangle = \langle LT(g_1), \dots, LT(g_s) \rangle.$$

Gröbner basis algorithms based on linear algebra compute the row echelon form of certain submatrices of the Macaulay matrix for some degree d of the system $\mathcal{F}$. The first nonzero element of each row corresponds to a leading monomial of an element of I, belonging to $LT(I)$. For large enough d, Dickson's lemma [CLO97, §2.4, Thm. 5] implies that the collection of those monomials up to degree d generates $LT(I)$ and thus the polynomials corresponding to those rows together form a Gröbner basis of I with respect to the chosen monomial ordering.

During the echelonization process, it may happen that a given row yields zero when reduced modulo the basis under construction. This is called *reduction to zero* in the literature. Ideally, one would like to avoid spending time on computations for rows whose result is zero. For this, several criteria have been proposed and allow to avoid the effective computation of useless reductions [Buc65, Fau02]. We briefly recall here the criteria used in the F5-like algorithms [Fau02, Bar04], which guarantee that there are no reductions to zero during the algorithm for semi-regular sequences of polynomials.

Let us first consider $\mathcal{F} = \{f_1, \dots, f_m\}$ a homogeneous system of polynomials in $\mathbb{F}_p[x_1, \dots, x_n]$ with $\deg f_i = d_i$. When applying the General Criterion [Fau02], the algorithm constructs the matrix $Mac_{d,i}(\mathcal{F})$ by adding to the matrix $Mac_{d,i-1}(\mathcal{F})$ all rows representing polynomials uf_i except for those where u is a leading term of a row in $\widetilde{Mac}_{d-d_i,i-1}(\mathcal{F})$, the row echelon form of $Mac_{d-d_i,i-1}(\mathcal{F})$.

Faugère shows that if the sequence of polynomials is regular, then the only reductions to zero during the execution of the algorithm for finite fields of characteristic strictly greater than 2 are those detected by the General Criterion. For a system of polynomials

$\mathcal{F} = \{f_1, \ldots, f_m\}$ in $\mathbb{F}_2[x_1, \ldots, x_n]$, if the goal is to find solutions in $\mathbb{F}_2$, we may as well add to this system the equations $\{x_1^2 - x_1, \ldots, x_n^2 - x_n\}$. Working with the system $\mathcal{F} \cup \{x_1^2 - x_1, \ldots, x_n^2 - x_n\}$ in $R = \mathbb{F}_2[x_1, \ldots, x_n]$ is equivalent to working with the polynomial system $\mathcal{F}$ in $R_n = R/\langle x_1^2 - x_1, \ldots, x_n^2 - x_n\rangle$. Consequently, when running the F5 algorithm, we also need to remove all reductions to zero coming from the fact that $f^2 = f$, for any $f \in R_n$.

The Frobenius Criterion [Fau02] states that a row representing the polynomial uf_i is a linear combination of previous rows if u is a leading term of a row in $\widetilde{Mac}_{d-d_i,i}(\mathcal{F})$, the row echelon form of $Mac_{d-d_i,i}(\mathcal{F})$.

Bardet [Bar04] extends these two criteria for reductions to zero to sequences of polynomials where $m > n$ by introducing the notion of semi-regularity. We recall this notion here, but let us first introduce some more notation.

For $d \geq 0$, we denote by $\mathbb{F}_p[x_1, \ldots, x_n]_d$ the $\mathbb{F}_p$-vector space of homogeneous polynomials of degree d. Let I be an ideal of dimension 0 generated by the sequence $\mathcal{F} = \{f_1, \ldots f_m\}$ and denote by $I_d = \mathbb{F}_p[x_1, \ldots, x_n]_d \cap I$. Then there exists $D \geq 0$ such that

$$\dim_{\mathbb{F}_p}(I_d) = \dim_{\mathbb{F}_p}(\mathbb{F}_p[x_1, \ldots, x_n]_d),$$

for all $d \geq D$ [CLO97] and we define D_{reg} to be the smallest degree with this property.

From now on, we consider systems defined over the finite field $\mathbb{F}_2$. Since the homogeneous part of highest degree of field equations $x_i^2 - x_i$ is x_i^2, we consider the ring $R_n^h = \mathbb{F}_2[x_1, \ldots, x_n]/\langle x_1^2, \ldots, x_n^2\rangle$. Any homogeneous polynomial of degree d in R_n^h verifies $f^2 = 0$. Following Bardet [Bar04], we directly state the definition of a semi-regular sequence of polynomials defined over $\mathbb{F}_2$.

Definition 2. A sequence of homogeneous polynomials $\{f_1, \ldots, f_m\}$ in R_n^h is called semi-regular over $\mathbb{F}_2$ if:

1. $\langle f_1, \ldots, f_m\rangle \neq R_n^h$,
2. For all $i \in \{1, \ldots, m\}$ if $g_i f_i = 0$ in $R_n^h/\langle f_1, \ldots, f_{i-1}\rangle$ and $\deg(g_i f_i) < D_{reg}$, then $g_i \in \langle f_1, \ldots, f_{i-1}, f_i\rangle$.

Given a power series $S \in \mathbb{Z}[[X]]$, the notation $[S]_n$ denotes the series obtained by truncating S just before the index n. Bardet computes the Hilbert series of the ideal generated by a semi-regular sequence of quadratic homogeneous polynomials $\mathcal{F} = \{f_1, \ldots f_m\}$ as

$$HF_{R/I}(X) = \left[\frac{(1+X)^n}{(1+X^2)^m}\right]_{D_{reg}}. \tag{1}$$

The degree of regularity of a semi-regular system is in fact given by the index of the first non-positive coefficient of the series in Equation (1).

As shown by Bardet [Bar04], if the sequence is semi-regular and the two criteria are used for recursively constructing the Macaulay matrices, then there are no reductions to zero in the F5 algorithm (called Matrix F5 in [Bar04]), until the degree $d = D_{reg} - 1$ is reached. When degree D_{reg} is reached, the algorithm outputs a Gröbner basis with respect to the chosen monomial order.

Consequently, the complexity of the Gröbner basis computation using the Matrix F5 algorithm is:

$$\mathcal{O}\left(\binom{n + D_{reg} - 1}{n}^{\omega}\right),$$

where ω is a linear algebra constant.

If the system is affine, it suffices to examine its homogeneous part of highest degree to ensure that there are no degree falls before reaching D_{reg}, during the execution of the algorithm. Following again [Bar04], we give the following definition.

Definition 3. Let $\mathcal{F} = \{f_1, \dots, f_m\}$ be an affine sequence of polynomials and denote by f_i^{top} the homogeneous part of highest degree of f_i, $1 \leq i \leq m$. Then $\mathcal{F}$ is called semi-regular if the sequence $\mathcal{F}^{top} = \{f_1^{top}, \dots, f_m^{top}\}$ is semi-regular.

By extension, we will call degree of regularity of an affine sequence the degree of regularity of $\mathcal{F}^{top}$ and denote it as well by D_{reg}.

In practice, if the sequence $\mathcal{F}$ is affine, Gröbner basis algorithms will perform Gaussian elimination on Macaulay matrices $Mac_{\leq d,m}(\mathcal{F})$, with $d \geq 0$. Then we call *solving degree* the smallest degree for which linear algebra on the Macaulay matrices will produce a Gröbner basis and denote its value by D_{sol}. This solving degree depends on the precise algorithm that one adopts. The interested reader is referred to [KY24] for a survey on known bounds on the solving degree. Under certain conditions, $D_{sol} < D_{reg}$ (see for instance [BND+22, Corollary 3.6]).

We denote :

$$I_{\leq d} = \left\{ p = \sum_{i=1}^{m} h_i f_i + \sum_{j=1}^{n} h_j (x_j^2 - x_j) \in I \middle| deg(h_i) \leq d-2, deg(h_j) \leq d-2 \right\},$$

and $R_{\leq d} = \oplus_{0 \leq d' \leq d} \left(\mathbb{F}_2[x_1, \dots, x_n]_{d'} \right)$. Then the generating series $G_{R/I}(X)$ is defined as follows:

$$G_{R/I}(X) = \sum_{d \geq 0} \dim(R_{\leq d}/I_{\leq d}) X^d. \tag{2}$$

If the sequence $\mathcal{F}$ is semi-regular, then up to the degree of regularity, the coefficient $\dim(R_{\leq d}/I_{\leq d})$ is equal to difference between the number of columns and rows of the matrix $Mac_{\leq d,m}(\mathcal{F})$. The following result is folklore, but we state it here for completeness.

Lemma 1. *Let $\mathcal{F} = \{f_1, \dots, f_m, x_1^2 - x, \dots, x_n^2 - x_n\}$ be an affine semi-regular sequence of quadratic polynomials defined over $\mathbb{F}_2$.*

Then, the coefficient corresponding to X^d of the generating series $G_{R/I}(X)$ is equal to the one in the expansion of the following generating function:

$$\frac{(1+X)^n}{(1-X)(1+X^2)^m}$$

for all $d \leq D_{reg}$.

Proof. See Appendix A. □

Finally, to analyze the complexity of Crossbred, we will make the standard assumption that the input sequence $\mathcal{F}$ for the algorithm is strong semi-regular (see for instance [BFSS13]). Roughly speaking, this means that for almost all possible assignments $x_{k+1} = a_{k+1}, \dots, x_n = a_n$, for some $k > 0$, the sequence

$$\{f_1(x_1, \dots, x_k, a_{k+1}, \dots, a_n), \dots, f_m(x_1, \dots, x_k, a_{k+1}, \dots a_n)\}$$

is semi-regular. We slightly adapt here the definition in [BFSS13].

Definition 4. Let $\mathcal{F} = \{f_1, \dots, f_m\}$ be a semi-regular sequence of polynomials in $\mathbb{F}_2[x_1, \dots, x_n]$ and let $0 \leq \gamma \leq 1$ such that $k = (1-\gamma)n$. We say that this sequence is γ-strong semi-regular if

$$\begin{aligned}&\mathcal{S}(I) = \{(a_{k+1}, \dots, a_n) \in \mathbb{F}_2^{n-k} | \\ &\{f_1(x_1, \dots, x_k, a_{k+1}, \dots, a_n), \dots, f_m(x_1, \dots, x_k, a_{k+1}, \dots a_n)\} \text{ is not semi-regular}\}\end{aligned}$$

has cardinality $O(2^{-\gamma n})$.

Algorithm 1: The Crossbred algorithm

```
Data: Polynomial system $\mathcal{F}$ of $m$ equations of $n$ variables, and $D, d, k$
Result: A solution of the system (or nothing otherwise)
Construct $Mac^k_{\leq D,\geq d,m}(\mathcal{F})$ and $\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})$
Compute a basis $(v_1, \ldots, v_r)$ of the left kernel of $\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})$
Construct polynomials $p_1, \ldots, p_r$ corresponding to $v_i \cdot Mac^k_{\leq D,\geq d,m}(\mathcal{F})$
for $a = (a_{k+1}, \ldots, a_n) \in \mathbb{F}_2^{n-k}$ do
    Evaluate the last $n-k$ variables in each $f \in \mathcal{F}$ at $(a_{k+1}, \ldots, a_n)$ and compute $\mathcal{F}^*$
    Compute $Mac_{\leq d,m}(\mathcal{F}^*)$
    Compute $\mathcal{F}'^*$ as the partial evaluation of $\mathcal{F}' = \{p_1, \ldots, p_r\}$ at $(a_{k+1}, \ldots, a_n)$
    Consider the system $S^*$ consisting of $Mac_{\leq d,m}(\mathcal{F}^*) \cup \mathcal{F}'^*$
    if $S^*$ is consistent then
        return the solution
    end
end
```

In particular, if a sequence $\mathcal{F}$ is γ-strong semi-regular, then, for any specialised sequence $\mathcal{F}^*$, up to the degree of regularity, there are no other linear dependencies between the rows of the matrix $Mac_{\leq d,m}(\mathcal{F}^*)$ than the one given by the criteria. In Appendix B we show a series of experiments which support the claim that random systems satisfy this property.

We denote by $d_{reg}(k)$ (resp. $d_{sol}(k)$) the degree of regularity of $(\mathcal{F}^*)^{top}$ (resp. solving degree of $\mathcal{F}^*$) for all $(a_{k+1}, \ldots, a_n) \in \mathbb{F}_2^{n-k} \backslash \mathcal{S}(I)$.

2.2 The Crossbred algorithm

In all the rest of the paper, we choose the glex monomial order over $\mathbb{F}_2[x_1, \ldots, x_n]$ and consider a polynomial system $\mathcal{F} = \{f_1, \ldots, f_m\}$ such that $deg(f_i) = 2$ and $deg_k(f_i) = 2$. This is a tacit condition for the algorithm to work. In a nutshell, the Crossbred algorithm [JV17] for fixed input parameters D, d and k, runs in three steps:

(1) Construct r new polynomials $p_1, \ldots, p_r$ of total degree D and of degree d over the first k variables. These polynomials are added to the original system.

(2) Assign the last $n-k$ variables in the system obtained in this way.

(3) Try to solve the resulting system in k variables. If no solution is found, continue the exhaustive search through $\mathbb{F}_2^{n-k}$ and re-assign the values of the last $n-k$ variables.

The pre-processing step (1) of Crossbred performs linear algebra on certain submatrices of the Macaulay matrices introduced in Definition 1. We introduce these, as well as the submatrices we use in our simplified version of Crossbred in Section 3 in the following definition.

Definition 5. Given an affine system of polynomials $\mathcal{F} = \{f_1, \ldots, f_m\}$ in R, let $Mac^k_{\leq D,\geq d,m}(\mathcal{F})$ be the submatrix of the Macaulay matrix $Mac_{\leq D,m}(\mathcal{F})$ whose rows correspond to products of the form uf_i, $1 \leq i \leq m$, with $\deg_k(u) \geq d-1$. Let $\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})$ be the submatrix of $Mac^k_{\leq D,\geq d,m}(\mathcal{F})$ whose columns correspond to monomials M with $\deg_k(M) \geq d+1$.

The pre-processing step roughly works as follows: assuming that D, d and k are such that the number of rows of $\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})$ is greater than its number of columns, we can compute a basis of the co-kernel of $\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})$. We denote by $v_1, \ldots, v_r$ the vectors of

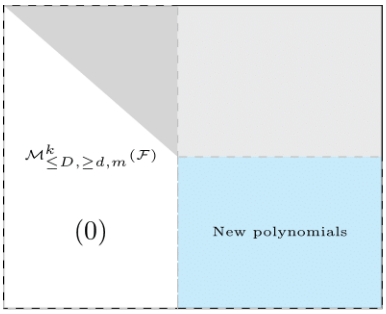

Figure 1: Partial Gaussian elimination of $Mac^k_{\le D,\ge d,m}(\mathcal{F})$

this basis. Now, each p_i is generated as the polynomial whose coefficients correspond to the product $v_i \cdot Mac^k_{\le D,\ge d,m}(\mathcal{F})$. From Definition 5, it is clear that $\deg_k(p_i) \le d$.

Note that for step (3) there exist multiple ways to solve the resulting system. In this paper, we only consider the resolution via linearization on the resulting Macaulay matrix of degree d of the specialized system. In other words, we think of each monomial in the system as an unknown, and try to solve the linear system obtained in this way. The pseudocode of the Crossbred algorithm is given in Algorithm 1.

Notation. Given a polynomial $f \in \mathbb{F}_2[x_1, \dots, x_n]$, we denote by f^* any polynomial in $\mathbb{F}_2[x_1, \dots, x_k]$ obtained from f after assigning the variables $x_{k+1}, \dots, x_n$. Similarly, given a sequence $\mathcal{F} = \{f_1, \dots, f_m\}$, we denote by $\mathcal{F}^* = \{f_1^*, \dots, f_m^*\}$. and call $\mathcal{F}^*$ the specialized sequence of $\mathcal{F}$.

Remark 1. In [JV17], Joux and Vitse considered an extra input parameter h which is used for parallelization of the algorithm. This parameter corresponds to the number of variables which are guessed before running the pre-processing step for each thread. In our analysis, we chose to ignore this parameter since this is equivalent to solving a polynomial system with $n - h$ variables on each thread.

2.2.1 Another look at the pre-processing step

To better understand the pre-processing step of Algorithm 1, we look at it from a different angle. We construct the matrix $Mac^k_{\le D,\ge d,m}(\mathcal{F})$. From left to right, we order the columns corresponding to monomials of degree D, $D-1$, $\dots$, $d+1$, d and $d-1$ over the first k variables. We denote by $M^k_{\le D,\ge d}$ ($M^k_{D,d}$) the number of monomials m such that $\deg(m) \le D$ and $\deg_k(m) \ge d$ ($\deg(m) = D$ and $\deg_k(m) = d$, respectively).

Instead of computing the kernel of $\mathcal{M}^k_{\le D,\ge d,m}(\mathcal{F})$, an equivalent way to generate new polynomials in the pre-processing step of Algorithm 1 is to partially echelonize $Mac^k_{\le D,\ge d,m}(\mathcal{F})$, such that the submatrix given by its first $\#M^k_{\le D,\ge d}$ rows is in row echelon form. New polynomials are obtained by taking the rows that have zero entries on the columns corresponding to monomials of degree $\ge d+1$ over the first k variables, as shown in Figure 1.

2.2.2 Limitations of the Crossbred algorithm

To the best of our knowledge, the question of determining a set of parameters (D, d, k) which are admissible (i.e. which ensure that the algorithm terminates) remains open. An even harder question is to determine an optimal choice of parameters D, d and k, which would balance both its running and its memory cost. There are several attempts to study the complexity of Crossbred [CHR+19, Dua23, BMSV22, Nak24]. For instance, Duarte proposes the following complexity for the algorithm:

$$\tilde{\mathcal{O}}\left(\left(\sum_{d_k=d+1}^{D}\sum_{d'=0}^{D-d_k}\binom{k}{d_k}\binom{n-k}{d'}\right)^{\omega}\right)+2^{n-k}\cdot\tilde{\mathcal{O}}\left(\left(\sum_{i=0}^{d}\binom{k}{i}\right)^{\omega}\right). \tag{3}$$

The first term of the sum in Equation (3) corresponds to the complexity to compute the co-kernel of $\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})$ (step (1), the pre-processing), 2^{n-k} represents the number of times we would have to assign the last $n-k$ variables in the worst case scenario and the last term corresponds to the complexity of solving the specialized system of k variables (step (3)).

In [JV17], the authors mostly experimented with $D = 3$ and expressly stated that when $D \geq 4$, linear dependencies appears between the rows of $Mac^k_{\leq D,\geq d,m}(\mathcal{F})$ Indeed, for larger degree D, we need to take in account trivial relations of the form $f_if_j = f_jf_i$. As such, the Crossbred algorithm gets slower from this point as it does useless operations. A way to improve the algorithm would be to remove those dependencies before performing linear algebra operations. This problem is addressed by applying the criteria presented in Section 2.1 to remove those dependencies. More specifically, we consider both criteria as proposed in [Bar04] and adapt them to the Crossbred algorithm.

3 A simplified version of Crossbred

Our first goal is to remove linear dependencies of the form $f_if_j = f_jf_i$ and $f^2 = f$ which appear while running the pre-processing step performed on the matrix $Mac^k_{\leq D,\geq d,m}(\mathcal{F})$ in the Crossbred algorithm. For that, let us consider a quadratic homogeneous system of polynomials $\mathcal{F} = \{f_1, \ldots, f_m\}$. Instead of working with $Mac^k_{\leq D,\geq d,m}(\mathcal{F})$ we will only consider blocks of this matrix which correspond to homogeneous polynomials (u, f) of total degree $\deg(uf) = D$ and $\deg_k(u) = d-1$.

Definition 6. Given an homogeneous system of polynomials $\mathcal{F} = \{f_1, \ldots, f_m\}$ in R, let $Mac^k_{D,d,m}(\mathcal{F})$ be the submatrix of the Macaulay matrix $Mac_{D,m}(\mathcal{F})$ whose rows correspond to products of the form uf_i, $1 \leq i \leq m$, with $\deg_k(u) = d-1$. Let $\mathcal{M}^k_{D,d,m}(\mathcal{F})$ be the submatrix of $Mac^k_{D,d,m}(\mathcal{F})$ whose columns correspond to monomials M with $\deg_k(M) = d+1$.

We will adapt Algorithm 1 to generate recursively matrices $Mac^k_{d_1,d_2,m}(\mathcal{F})$ and $\mathcal{M}^k_{d_1,d_2,m}(\mathcal{F})$, for all $d_1 \leq D$ and $d_2 \leq D$. During the process, we will apply two criteria to sieve out polynomials that give reductions to zero when performing linear algebra on this matrix.

We denote by $\widetilde{Mac}^k_{D,d,m}(\mathcal{F})$ the row-reduced echelon form of the matrix $Mac^k_{D,d,m}(\mathcal{F})$. The following proposition is an adaptation of Faugère's General Criterion [Fau02] to the matrices used in our algorithm and by extension, to those used in Crossbred.

Proposition 1. *(The General Criterion) Let $d \leq d_2 \leq d_1 \leq D$. For any row in $\widetilde{Mac}^k_{d_1-2,d_2-2,m-1}(\mathcal{F})$ having the monomial t with $\deg_k t = d_2 - 1$ as a leading term, the row labeled by (t, f_m) is a linear combination of previous rows in $Mac_{d_1,m}(\mathcal{F})$ and the polynomial generated from this row in the pre-processing step of the Crossbred algorithm is a linear combination of rows of the matrix $Mac_{\leq d_2,m}(\mathcal{F}^*)$, where $\mathcal{F}^*$ is the system obtained by assigning the last $n-k$ variables at any $(a_{k+1}, \ldots, a_n) \in \mathbb{F}_2^{n-k}$.*

Proof. Assume that $t = LT(f)$, where $f = \sum_{j=1}^{m-1} g_jf_j$ with g_j polynomials with $\deg(g_if_i) =$

$d_1 - 2$ and $\deg_k(g_i) = d_2 - 3$. We may then write:

$$\begin{aligned} tf_m &= ff_m - (f - LT(f))f_m \\ &= \sum_{j=1}^{m-1} (g_j f_m) f_j - (f - LT(f))f_m. \end{aligned}$$

Hence we have:

$$tf_m = \sum_{j=1}^{m-1} \left(\sum_i u_{i,j} + v_{i,j} + w_{i,j} \right) f_j - \left(\sum_i u_{i,m} + v_{i,m} + w_{i,m} \right) f_m, \tag{4}$$

with $deg_k(u_{i,j}) = d_2 - 1$, $deg_k(v_{i,j}) = d_2 - 2$ and $deg_k(w_{i,j}) = d_2 - 3$ and $u_{i,m} < t'$.

Denote by s the number of columns of $\mathcal{M}^k_{\leq D, > d, m}(\mathcal{F})$. As explained in Section 2.2.1, in order to generate the new polynomials in the Crossbred algorithm it suffices to partially echelonize the first s rows in $Mac^k_{\leq D, \geq d, m}(\mathcal{F})$ and extract the sub-matrix corresponding to rows that only have zero coefficients on these s columns.

If a row labeled (t, f_m) (i.e. the row whose coefficients are given by those of tf_m) generates a new polynomial in the pre-processing step of Crossbred, then the row corresponding to this polynomial after partially echelonizing $Mac^k_{\leq D, \geq d, m}(\mathcal{F})$ is of the form:

$$p = tf_m + \sum_{j=1}^{m} \sum_i u_{i,j} f_j,$$

where $u_{i,j}$ correspond to the monomials of degree $d_2 - 1$ over the first k variables given in Equation (4). Using Equation (4), we have:

$$\begin{aligned} p &= \sum_{j=1}^{m-1} \left(\sum_i v_{i,j} + w_{i,j} \right) f_j - \left(\sum_i v_{i,m} + w_{i,m} \right) f_m \\ &= \sum_{j=1}^{m} \left(\sum_i v_{i,j} + w_{i,j} \right) f_j. \end{aligned}$$

After specialization, we have:

$$p^* = \sum_{j=1}^{m} \left(\sum_i v^*_{i,j} + w^*_{i,j} \right) f^*_j.$$

Note that $v^*_{i,j} f^*_j$ and $w^*_{i,j} f^*_j$ are rows of $Mac_{\leq d_2, m}(\mathcal{F}^*)$. Hence the polynomial generated with the row labeled (t, f_m) is a linear combination of rows of $Mac_{\leq d_2, m}(\mathcal{F}^*)$. □

As explained in Section 2.1, for polynomial systems defined over $\mathbb{F}_2$ another set of linear dependencies appear due to the fact that for any polynomial f we have $f^2 = f$. The following proposition is an adaption of the Frobenius criterion to the matrices used in Crossbred.

Proposition 2. *(The Frobenius Criterion) Let $d \leq d_2 \leq d_1 \leq D$. For any row in $\widetilde{Mac}^k_{d_1-2, d_2-2, m}(\mathcal{F})$ having the monomial t as a leading term, the row labeled by (t, f_m) is a linear combination of previous rows in $Mac_{d_1, m}(\mathcal{F})$ and the polynomial generated with this row in the pre-processing step of the Crossbred algorithm is a linear combination of rows of $Mac_{\leq d_2, m}(\mathcal{F}^*)$, where $\mathcal{F}^*$ is the system obtained by assigning the last $n - k$ variables at any $(a_{k+1}, \ldots, a_n) \in \mathbb{F}_2^{n-k}$.*

Algorithm 2: Block Crossbred

```
Data: A polynomial system $\mathcal{F}$ of $m$ homogeneous polynomials of $n$ variables over
      $\mathbb{F}_2$, three parameters $D, d, k$
Result: A solution of the system (if it exists)
for $d_1$ from 2 to $D$ do
    for $d_2$ from 1 to $d-1$ do
        for $m'$ from 1 to $m$ do
            $Mac^k_{d_1,d_2,m'}(\mathcal{F}) \leftarrow$ GenMat($Mac^k_{d_1,d_2,m'-1}(\mathcal{F})$)
            Compute the echelon form $\widetilde{Mac}^k_{d_1,d_2,m'}(\mathcal{F})$
        end
    end
end
Generate $\mathcal{M}^k_{D,d,m}(\mathcal{F})$ from $Mac^k_{D,d,m}(\mathcal{F})$
$\mathcal{F}' \leftarrow$ GenPoly($\mathcal{M}^k_{D,d,m}(\mathcal{F})$)
for $(a_{k+1},\dots,a_n) \in \mathbb{F}_2^{n-k}$ do
    Partially evaluate each polynomial $f \in \mathcal{F}$ at $(a_{k+1},\dots,a_n)$
    Compute $Mac_{\leq d,m}(\mathcal{F}^*)$
    Compute $\mathcal{F}'^*$ as the partial evaluation of the $\mathcal{F}'$ polynomials at $(a_{k+1},\dots,a_n)$
    Consider the system $S^*$ consisting of $Mac_{\leq d,m}(\mathcal{F}^*) \cup \mathcal{F}'^*$
    if $S^*$ is consistent then
        return the solution
    end
end
```

Proof. Let $t = LT(f)$, where $f = \sum_{j=1}^{m} h_j f_j$ with h_j polynomials with $\deg h_j f_j = d_1 - 2$ and $\deg_k h_j = d_2 - 3$. We may then write:

$$\begin{aligned} tf_m &= ff_m - (f - LT(f))f_m \\ &= \sum_{j=1}^{m}(h_j f_m)f_j - (f - LT(f))f_m \\ &= \sum_{j=1}^{m-1}(h_j f_m)f_j + h_m f_m^2 - (f - LT(f))f_m \\ &= \sum_{j=1}^{m-1}(h_j f_m)f_j + h_m f_m - (f - LT(f))f_m. \end{aligned}$$

Hence tf_m can be written as a sum of polynomials uf_j with either $j \leq m$ and $\deg_k u = \{d_2 - 3, d_2 - 2, d_2 - 1\}$ or $j = m$ and $u < t$ with respect to the chosen monomial ordering. The rest of the proof is similar to that of the General Criterion (Proposition 1). □

We may now present a simplified version of Algorithm 1, which takes in a homogeneous system of polynomials $\mathcal{F}$ and is based on linear algebra on the matrix $Mac^k_{D,d,m}(\mathcal{F})$ (instead of $Mac^k_{\leq D,\geq d,m}(\mathcal{F})$ as in Crossbred). We recursively construct matrices $Mac^k_{d_1,d_2,m}(\mathcal{F})$ and apply the criteria in Propositions 1 and 2 in the process. The pseudocode of our algorithm, that we call Block Crossbred, is given in Algorithm 2. We will see in Section 5 that this algorithm has the advantage that it is easier to analyze than the original Crossbred.

How to generate $Mac^k_{d_1,d_2,m'}(\mathcal{F})$. In Algorithm 2 the **GenMat** method computes $Mac^k_{d_1,d_2,m'}(\mathcal{F})$ by adding to $Mac^k_{d_1,d_2,m'-1}(\mathcal{F})$ all rows labeled $(u, f_{m'})$ where u is a monomial with $\deg u = d_1-2$ and $deg_k(u) = d_2-1$ that does not satisfy the conditions in the General and Frobenius criteria (i.e. u is not a leading term of a row in $\widetilde{Mac}^k_{d_1-2,d_2-2,m'}(\mathcal{F})$).

GenPoly. In Algorithm 2 the **GenPoly** method takes in the matrix $\mathcal{M}^k_{d_1,d_2,m}$ to generate new polynomials that will be added to the initial system $\mathcal{F}$. To do that, we first compute the left kernel LK of the matrix $\mathcal{M}^k_{D,d,m}(\mathcal{F})$. We then obtain new polynomials thanks to the operation $LK \cdot Mac^k_{D,d,m}(\mathcal{F})$. This computation is similar to the pre-processing step of the Crossbred algorithm, each row of the resulting matrix represents a polynomial of total degree D and of degree d over the first k variables.

4 Semi-regular sequences and Block Crossbred

A first step towards understanding the complexity of the Block Crossbred, and eventually that of Crossbred, is to evaluate the cost of its pre-processing step. In order to compute the number of new polynomials generated by the **GenPoly** procedure, which depends on the dimensions of the kernels of the matrices $Mac^k_{D,d,m}(\mathcal{F})$ and $\mathcal{M}^k_{D,d,m}(\mathcal{F})$, we need to account for reductions to zero while constructing these matrices. We will use here a standard assumption in the literature, that of semi-regularity.

We define

$$R^k_{d_1,d_2} = \mathbb{F}_2[x_{k+1}, \dots, x_n]_{d_1-d_2}[x_1, \dots, x_k]_{d_2} \quad \text{and} \quad I^k_{d_1,d_2} = I_{d_1} \cap R^k_{d_1,d_2},$$

regarded as $\mathbb{F}_p$-vector spaces.

Proposition 3. *Let I be a zero-dimensional ideal. There exists a tuple (D_1, D_2) such that*

$$\dim_{\mathbb{F}_2} R^k_{D_1,D_2} = \dim_{\mathbb{F}_2} I^k_{D_1,D_2}.$$

Proof. It is well known that for all $D \geq D_{reg}$ we have $I_D = \mathbb{F}_2[x_1, \dots, x_n]_D$. Fix $D = D_{reg}$. Then for any (d_1, d_2) such that $d_1 \geq D_{reg}$ we have that

$$R^k_{d_1,d_2} \subseteq I_{d_1} = \mathbb{F}_2[x_1, \dots, x_n]_{d_1}.$$

It follows that $I^k_{d_1,d_2} = I_{d_1} \cap R^k_{d_1,d_2} = R^k_{d_1,d_2}$. □

The proof of this proposition suggests that there exists a set of parameters (D, d) with $d < D \leq D_{reg}$ such that the left kernel of the matrix $\mathcal{M}^k_{D,d,m}(\mathcal{F})$ has positive dimension (i.e. the **GenPoly** procedure generates new polynomials). Note that $d < D < D_{reg}$ is the only interesting case for Crossbred anyway, since when $D = D_{reg}$ the cost of the linear algebra in the pre-processing is asymptotically close to that of linear algebra in the F5 algorithm. Whenever the sequence of polynomials $\mathcal{F}$ is γ-strong semi-regular, we will show that there are no reductions to zero in the **GenMat** procedure of the Block Crossbred algorithm.

Proposition 4. *Let $\mathcal{F} = \{f_1, \dots, f_m\}$ be a homogeneous sequence of polynomials such that the ideal $I = \langle f_1, \dots, f_m \rangle$ has dimension 0. Let $0 < k < n$ and $0 < \gamma < 1$ such that $k = (1-\gamma)n$. Assume that the sequence $\mathcal{F}$ is γ-strong semi-regular. Then there are no reductions to zero in the matrix $Mac^k_{d_1,d_2,m}(\mathcal{F})$ constructed by the **GenMat** procedure of the Block Crossbred algorithm with $d_2 < d_{reg}(k)$ and $d_1 < D_{reg}$.*

Proof. Assume that there is a reduction to zero in the matrix $Mac^k_{d_1,d_2,m}(\mathcal{F})$, with $d_1 < D_{reg}$. Then there exist g_i and h_j, $j \in \{1, \dots, i-1\}$, such that $g_i f_i = \sum_{j=1}^{i-1} h_j f_j$, with $\deg(g_i f_i) = d_1$ and $g_i, h_j \in R^k_{d_1-2,d_2-1}$. From the semi-regularity hypothesis, it follows that $g_i = \sum_{j=1}^{i} h'_j f_j$. Since the sequence is γ-strong semi-regular, there is no fall of the degree over the first k variables for the specialized sequence, for $d_2 < d_{reg}(k)$. We deduce that $\deg_k(h'_j) = d_2 - 3$, hence $LT(g_i)$ is the leading term of a row in $Mac^k_{d_1-2,d_2-2,i}(\mathcal{F})$. These are exactly the rows that are removed when applying the General and the Frobenius Criteria. As such, there is no reduction to zero in the Block Crossbred pre-processing up to the degree D_{reg}.

□

5 A bivariate generating series for Block Crossbred

In this section, we investigate the complexity of the Block Crossbred algorithm for solving a system of polynomials. To this purpose, we have to estimate first the number of new polynomials obtained when running the **GenPoly** procedure in Algorithm 2.

Let $\mathcal{F} = \{f_1, \dots, f_m\}$ be a sequence of homogeneous quadratic polynomials in R and denote by $U^k_{d_1,d_2,m}$, $d_1, d_2 \geq 0$, the number of rows of the matrix $Mac^k_{d_1,d_2,m}(\mathcal{F})$, and thus of $\mathcal{M}^k_{d_1,d_2,m}(\mathcal{F})$. The number of columns of $\mathcal{M}^k_{d_1,d_2,m}(\mathcal{F})$ is given by $M^k_{d_1,d_2+1}$, which corresponds to the number of monomials v of total degree d_1 such that $\deg_k v = d_2 + 1$.

We define the following sequence:

$$h^k_{d_1,d_2,m} = \begin{cases} U^k_{d_1,d_2,m} - M^k_{d_1,d_2+1}, & \text{if } d_1 \geq d_2 \geq 0, m \geq 1, \\ -M^k_{d_1,0}, & \text{if } d_1 \geq 0, d_2 = -1, m \geq 0, \\ -M^k_{d_1,d_2+1}, & d_1 \geq d_2 \geq 0, m = 0, \\ 0 & \text{in all other cases.} \end{cases} \tag{5}$$

This sequence gives the number of new "independent" polynomials generated during the pre-processing step of Algorithm 2.

Proposition 5. *If $h^k_{D,d,m} > 0$ and there are no reductions to zero in $Mac^k_{D,d,m}(\mathcal{F})$, then the number of polynomials computed with the **GenPoly** procedure is $h^k_{D,d,m}$.*

Proof. The number of polynomials computed with the **GenPoly** procedure is given by $corank(\mathcal{M}^k_{D,d,m}(\mathcal{F})) - corank(Mac^k_{D,d,m}(\mathcal{F}))$. As there is no reduction to zero in $Mac^k_{D,d,m}(\mathcal{F})$ we have that

$$corank(Mac^k_{D,d,m}(\mathcal{F})) = 0.$$

On the other hand we compute:

$$corank(\mathcal{M}^k_{D,d,m}(\mathcal{F})) = U^k_{D,d,m} - rank(\mathcal{M}^k_{D,d,m}(\mathcal{F})).$$

Since $h^k_{D,d,m} > 0$, the number of rows of $\mathcal{M}^k_{D,d,m}(\mathcal{F})$ is greater to the number of columns. Furthermore, Since $\mathcal{M}^k_{D,d,m}(\mathcal{F})$ is a sub-matrix given by the first columns of the full-rank matrix $Mac^k_{D,d,m}(\mathcal{F})$, it is also full-rank. Hence the rank of $\mathcal{M}^k_{D,d,m}(\mathcal{F})$ is its number of columns.

We get:

$$corank(\mathcal{M}^k_{D,d,m}(\mathcal{F})) = U^k_{D,d,m} - \#Col(\mathcal{M}^k_{D,d,m}(\mathcal{F})) = h^k_{D,d,m}.$$

□

Proposition 6. *Assume that there are no reductions to zero in the Block Crossbred algorithm. Then the sequence $h^k_{d_1,d_2,m}$ satisfies the following recurrence relation:*

$$h^k_{d_1,d_2,m} = h^k_{d_1,d_2,m-1} - h^k_{d_1-2,d_2-2,m}. \tag{6}$$

Proof. The number of rows added to $Mac^k_{d_1,d_2,m-1}(\mathcal{F})$ to get $Mac^k_{d_1,d_2,m}(\mathcal{F})$ is equal to the number of monomials u with $\deg(u) = d_1 - deg(f_m) = d_1 - 2$ and $\deg_k(u) = d_2 - 1$. From this number we subtract the number of monomials which satisfy the General and the Frobenius Criterion. As such, the number of rows of the matrix $Mac^k_{d_1,d_2,m}(\mathcal{F})$ verifies the following equation:

$$U^k_{d_1,d_2,m} - U^k_{d_1,d_2,m-1} = M^k_{d_1-2,d_2-1} - U^k_{d_1-2,d_2-2,m}.$$

By using this formula and Equation (5) we get:

$$h^k_{d_1,d_2,m} - h^k_{d_1,d_2,m-1} = -h^k_{d_1-2,d_2-2,m},$$

which concludes the proof. □

Using the recurrence relation in Equation (6) we may now compute the generating bivariate series which will allow us to determine admissible parameters for Algorithm 2.

Theorem 7. *Let $H^k_{m,n}(X,Y) = \sum_{d_1\geq 0, d_2\geq 0} h^k_{d_1,d_2,m} X^{d_1}Y^{d_2}$ be the bivariate series with coefficients defined by Equation (5). This series is given by:*

$$H^k_{m,n}(X,Y) \quad = \quad \frac{1}{Y}\left((1+X)^{n-k} - \frac{(1+XY)^k(1+X)^{n-k}}{(1+X^2Y^2)^m}\right).$$

Proof. Since the values of k and n are fixed, let $H_m(X,Y) = H^k_{m,n}(X,Y)$ and write $h_{d_1,d_1,m}$ instead of $h^k_{d_1,d_2,m}$. Moreover, we define

$$\hat{H}_m(X,Y) = \sum_{d_1\geq 0, d_2\geq 0} h_{d_1-2,d_2-2,m} X^{d_1}Y^{d_2}.$$

Then, we have

$$\begin{aligned}
\hat{H}_m(X,Y) \quad &= \quad \sum_{d_1\geq 0,d_2\geq 0} U^k_{d_1-2,d_2-2,m}X^{d_1}Y^{d_2} - \sum_{d_1\geq 0,d_2\geq 0} M^k_{d_1-2,d_2-1}X^{d_1}Y^{d_2}\\
&= \quad X^2Y^2 \sum_{d_1\geq 0,d_2\geq 0} U^k_{d_1,d_2,m}X^{d_1}Y^{d_2}\\
&- \quad X^2Y^2 \sum_{d_1\geq 0,d_2\geq 0} M^k_{d_1,d_2+1}X^{d_1}Y^{d_2} - X^2Y\sum_{d_1\geq 0} M^k_{d_1,0}X^{d_1}\\
&= \quad X^2Y^2 \sum_{d_1\geq 0,d_2\geq 0} h_{d_1,d_2,m}X^{d_1}Y^{d_2} - X^2Y\sum_{d_1\geq 0} M^k_{d_1,0}X^{d_1}\\
&= \quad X^2Y^2H_m(X,Y) - X^2Y\sum_{d_1\geq 0} M^k_{d_1,0}X^{d_1}.
\end{aligned}$$

Using the recurrence relation obtained in Equation (6) we obtain:

$$\begin{aligned}
H_m(X,Y) \quad &= \quad \sum_{d_1\geq 0,d_2\geq 0} h_{d_1,d_2,m}X^{d_1}Y^{d_2}\\
&= \quad \sum_{d_1\geq 0,d_2\geq 0} h_{d_1,d_2,m-1}X^{d_1}Y^{d_2} - \sum_{d_1\geq 0,d_2\geq 0} h_{d_1-2,d_2-2,m}X^{d_1}Y^{d_2}\\
&= \quad H_{m-1}(X,Y) - \hat{H}_m(X,Y)\\
&= \quad H_{m-1}(X,Y) - X^2Y^2H_m(X,Y) + X^2Y\sum_{d_1\geq 0} M^k_{d_1,0}X^{d_1}.
\end{aligned}$$

Hence we get

$$\begin{aligned} H_m(X,Y) &= (1+X^2Y^2)^{-1}H_{m-1}(X,Y) + \frac{X^2Y}{1+X^2Y^2}\sum_{d_1\geq 0} M^k_{d_1,0}X^{d_1} \\ &= (1+X^2Y^2)^{-m}H_0(X,Y) - \frac{1-(1+X^2Y^2)^m}{Y(1+X^2Y^2)^m}\sum_{d_1\geq 0} M^k_{d_1,0}X^{d_1}. \end{aligned}$$

By Equation (5), for a fixed value of k, we have that $h_{d_1,d_2,0} = -M^k_{d_1,d_2+1}$. Hence we get :

$$H_0(X,Y) = -\sum_{d_1\geq 0, d_2\geq 0} M^k_{d_1,d_2+1}X^{d_1}Y^{d_2}.$$

Since $M^k_{d_1,d_2+1} = \binom{k}{d_2+1}\binom{n-k}{d_1-d_2-1}$ (with the convention that $\binom{n-k}{d_1-d_2-1} = 0$ for $d_2 \geq d_1 - 1$), we get:

$$\begin{aligned} &\sum_{d_1\geq 0, d_2\geq 0} M^k_{d_1,d_2+1}X^{d_1}Y^{d_2} = \sum_{d_1\geq 0, d_2\geq 0}\binom{k}{d_2+1}\binom{n-k}{d_1-d_2-1}X^{d_1}Y^{d_2} \\ &= \sum_{d_2\geq 0}\binom{k}{d_2+1}Y^{d_2}\sum_{d_1\geq 0}\binom{n-k}{d_1-d_2-1}X^{d_1} = \sum_{d_2\geq 0}\binom{k}{d_2+1}Y^{d_2}X^{d_2+1}\sum_{d_1\geq 0}\binom{n-k}{d_1}X^{d_1} \\ &= \frac{(1+X)^{n-k}}{Y}\sum_{d_2\geq 0}\binom{k}{d_2+1}Y^{d_2+1}X^{d_2+1} = \frac{(1+X)^{n-k}}{Y}\left(\sum_{d_2\geq 0}\binom{k}{d_2}Y^{d_2}X^{d_2} - 1\right) \\ &= \frac{(1+X)^{n-k}}{Y}\left((1+XY)^k - 1\right). \end{aligned}$$

In conclusion, we have:

$$\begin{aligned} H_m(X,Y) &= -\frac{(1+X)^{n-k}}{Y(1+X^2Y^2)^m}\left((1+XY)^k - 1\right) - \frac{1-(1+X^2Y^2)^m}{Y(1+X^2Y^2)^m}\sum_{d_1\geq 0} M^k_{d_1,0}X^{d_1} \\ &= \frac{1}{Y(1+X^2Y^2)^m}\left(-(1+XY)^k(1+X)^{n-k} + (1+X^2Y^2)^m(1+X)^{n-k}\right) \\ &= \frac{1}{Y}\left((1+X)^{n-k} - \frac{(1+XY)^k(1+X)^{n-k}}{(1+X^2Y^2)^m}\right). \end{aligned}$$

□

6 From Block Crossbred to Joux-Vitse's Crossbred

The previous analysis was made assuming that the input system was homogeneous. However, the original Joux-Vitse Crossbred Algorithm has been designed to work with an affine system $\mathcal{F}$. We notice that even if $\mathcal{F}$ is affine, the matrix $Mac^k_{\leq D,\geq d,m}(\mathcal{F})$ can be constructed by concatenating the matrices $Mac^k_{d_1,d_2,m}(\mathcal{F})$, $d \leq d_2 \leq d_1 \leq D$, constructed in Algorithm 2. This observation allows us to remove reductions to zero arising from the criteria in the pre-processing step and compute the corank of the matrix $\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})$ in terms of the coefficients of the generating series $H^k_{m,n}(X,Y)$ examined in Proposition 7.

Proposition 8. *Assuming that there are no reductions to zero in the pre-processing step of the Crossbred algorithm, the corank of the matrix $\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})$ is given by the following formula:*

$$corank(\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})) = \sum_{d_1\leq D, d_2\geq d} h^k_{d_1,d_2,m},$$

where the sequence $h^k_{d_1,d_2,m}$ is defined by Equation (5).

Proof. Indeed, we have that:

$$\begin{aligned} corank(\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})) &= \#Rows(\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})) - \#Col(\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})) \\ &= \sum_{d_1\leq D, d_2\geq d} U^k_{d_1,d_2} - \sum_{d_1\leq D, d_2\geq d} M^k_{d_1,d_2+1} = \sum_{d_1\leq D, d_2\geq d} h^k_{d_1,d_2,m}. \end{aligned}$$

□

We are now in position to compute the generating bivariate series which will eventually allow us to determine admissible parameters for Algorithm 1.

Proposition 9. *For fixed values of m, n and k the bivariate series $G^k_{m,n}(X,Y) = \sum_{d_1\geq 0, d_2\geq 0}\left(\sum_{d'_1\leq d_1, d'_2\geq d_2} h^k_{d'_1,d'_2,m}\right) X^{d_1}Y^{d_2}$ is given by the formula:*

$$G^k_{m,n}(X,Y) = -\frac{YH^k_{m,n}(X,Y) - H^k_{m,n}(X,1)}{(1-X)(1-Y)}. \tag{7}$$

Proof. Since m, n and k are fixed, we denote by $G(X,Y) = G^k_{m,n}(X,Y)$ and $H(X,Y) = H^k_{m,n}(X,Y)$. We compute the bivariate series $G(X,Y)$ as follows:

$$G(X,Y) = \sum_{d_1\geq 0, d_2\geq 0}\left(\sum_{d'_1\leq d_1, d'_2\geq d_2} h_{d'_1,d'_2,m}\right) X^{d_1}Y^{d_2}.$$

First note that

$$\begin{aligned} \sum_{d'_1\leq d_1, d'_2\geq d_2} h_{d'_1,d'_2,m}X^{d_1}Y^{d_2} =& h_{d_1,d_2,m}X^{d_1}Y^{d_2} + \sum_{d'_1\leq d_1-1} h_{d'_1,d_2,m}X^{d_1}Y^{d_2} \\ &+ \sum_{d'_2\geq d_2+1} h_{d_1,d'_2,m}X^{d_1}Y^{d_2} + \sum_{d'_1\leq d_1-1, d'_2\geq d_2+1} h_{d'_1,d'_2,m}X^{d_1}Y^{d_2}. \end{aligned}$$

It follows that

$$\begin{aligned} G(X,Y) =& \sum_{d_1\geq 0, d_2\geq 0} h_{d_1,d_2,m}X^{d_1}Y^{d_2} + \frac{X}{Y}\sum_{d_1\geq 0, d_2\geq 0}\left(\sum_{d'_1\leq d_1-1, d'_2\geq d_2+1} h_{d'_1,d'_2,m}\right) X^{d_1-1}Y^{d_2+1} \\ &+ \sum_{d_1\geq 0, d_2\geq 0}\left(\sum_{d'_1\leq d_1-1} h_{d'_1,d_2,m}\right) X^{d_1}Y^{d_2} + \sum_{d_1\geq 0, d_2\geq 0}\left(\sum_{d'_2\geq d_2+1} h_{d_1,d'_2,m}\right) X^{d_1}Y^{d_2}. \end{aligned} \tag{8}$$

We denote by S the sequence

$$S(X,Y) = \sum_{d_1\geq 0, d_2\geq 0}\left(\sum_{d'_1\leq d_1-1} h_{d'_1,d_2,m}\right) X^{d_1-1}Y^{d_2},$$

and by T the sequence

$$T(X,Y) = \sum_{d_1\geq 0, d_2\geq 0}\left(\sum_{d'_2\geq d_2+1} h_{d_1,d'_2,m}\right) X^{d_1}Y^{d_2+1}.$$

Then we write

$$\begin{aligned}S(X,Y) &= \sum_{d_1\geq 1, d_2\geq 0} h_{d_1-1,d_2,m}X^{d_1-1}Y^{d_2} + \sum_{d_1\geq 1, d_2\geq 0}\sum_{d_1'\leq d_1-2} h_{d_1',d_2,m}X^{d_1-1}Y^{d_2}\\ &= \sum_{d_1\geq 0, d_2\geq 0} h_{d_1,d_2,m}X^{d_1}Y^{d_2} + X\sum_{d_1\geq 0, d_2\geq 0}\sum_{d_1'\leq d_1-2} h_{d_1',d_2,m}X^{d_1-2}Y^{d_2}\\ &= H(X,Y) + XS(X,Y),\end{aligned}$$

and get that

$$S(X,Y) = \frac{H(X,Y)}{1-X}. \tag{9}$$

To compute $T(X,Y)$ we follow a similar approach:

$$\begin{aligned}T(X,Y) &= \sum_{d_1\geq 0, d_2\geq 0}\sum_{d_2'\geq d_2+1} h^k_{d_1,d_2',m}X^{d_1}Y^{d_2+1} = \sum_{d_1\geq 0, d_2\geq 1}\sum_{d_2'\geq d_2} h^k_{d_1,d_2',m}X^{d_1}Y^{d_2}\\ &= \sum_{d_1\geq 0, d_2\geq 0}\sum_{d_2'\geq d_2} h^k_{d_1,d_2',m}X^{d_1}Y^{d_2} - \sum_{d_1\geq 0}\sum_{d_2'\geq 0} h^k_{d_1,d_2',m}X^{d_1}\\ &= \sum_{d_1\geq 0, d_2\geq 0}\sum_{d_2'\geq d_2+1} h^k_{d_1,d_2',m}X^{d_1}Y^{d_2} + \sum_{d_1\geq 0, d_2\geq 0}\sum_{d_2'= d_2} h^k_{d_1,d_2',m}X^{d_1}Y^{d_2}\\ &- \sum_{d_1\geq 0}\sum_{d_2'\geq 0} h^k_{d_1,d_2',m}X^{d_1}\\ &=\frac{1}{Y}T(X,Y) + H(X,Y) - H(X,1).\end{aligned}$$

We conclude that :

$$T(X,Y) = \frac{Y}{Y-1}\left(H(X,Y) - H(X,1)\right). \tag{10}$$

Now let us focus on the series

$$\hat{G} = \sum_{d_1\geq 0, d_2\geq 0}\left(\sum_{d_1'\leq d_1-1, d_2'\geq d_2+1} h_{d_1',d_2',m}\right)X^{d_1-1}Y^{d_2+1},$$

which appears in the second term of the sum in Equation (8). We have that

$$\begin{aligned}\hat{G} &= \sum_{d_1\geq 0, d_2\geq 0}\left(\sum_{d_1'\leq d_1, d_2'\geq d_2+1} h_{d_1',d_2',m}X^{d_1}Y^{d_2+1}\right)\\ &= \sum_{d_1\geq 0, d_2\geq 0}\left(\sum_{d_1'\leq d_1-1, d_2'\geq d_2+1} h_{d_1',d_2',m}\right)X^{d_1}Y^{d_2+1} + \sum_{d_1\geq 0, d_2\geq 0}\left(\sum_{d_2'\geq d_2+1} h_{d_1,d_2',m}\right)X^{d_1}Y^{d_2+1}.\end{aligned}$$

We have

$$\hat{G}(X,Y) = X\hat{G}(X,Y) + T(X,Y),$$

hence we compute

$$\hat{G}(X,Y) = \frac{1}{1-X}T(X,Y).$$

Finally, we obtain

$$\begin{aligned} G(X,Y) &= H(X,Y) + \frac{X}{Y(1-X)}T(X,Y) + XS(X,Y) + \frac{1}{Y}T(X,Y) \\ &= H(X,Y) + XS(X,Y) + \frac{1}{(1-X)Y}T(X,Y). \end{aligned} \tag{11}$$

We plug in the expressions obtained in Equations (9) and (10) in the last equality in Equation (11) and conclude that

$$G(X,Y) = \frac{H(X,1)}{(1-X)(1-Y)} - \frac{YH(X,Y)}{(1-X)(1-Y)}.$$

□

For a fixed value of k, the positive coefficients of $G^k_{m,n}$ give us values of (d_1, d_2) for which the left kernel of $Mac^k_{\leq d_1, \geq d_2, m}(\mathcal{F})$ is non-trivial. Consequently, for these pairs (d_1, d_2) the number of polynomials generated during the pre-processing step, taking into account the criteria, is given by the coefficient of $X^{d_1}Y^{d_2}$.

Example 6.1. We are interested in solving a semi-regular polynomial system with $m = 160$ polynomials and $n = 80$ variables, which is the set of parameters for one of the recent record of a polynomial system that was solved over $\mathbb{F}_2$ in the Fukuoka Type I MQ challenge [BS23]. By choosing $k = 24$, we get the following series:

$$\begin{aligned} G^{24}_{160,80}(X,Y) = \; & - \; 24X - 1484X^2 - 116X^2Y - 43124X^3 - 4796X^3Y + 1816X^3Y^2 \\ & - \; 764694X^4 - 61086X^4Y + 124166X^4Y^2 + 20654X^4Y^3 \\ & - \; 8869694X^5 + 648874X^5Y + 4049646X^5Y^2 + 1149494X^5Y^3 \\ & - \; 27784X^5Y^4 + \mathcal{O}(X^6). \end{aligned}$$

Using this series, we deduce that the number of polynomials generated by the pre-processing of the Crossbred algorithm (when applying the criteria) for parameters $(5, 1, 24)$ is 648874.

Now that we know how many polynomials are generated by the pre-processing step of the algorithm, we need to check if it generates enough polynomials for the algorithm to terminate. To tackle the question of determining admissible parameters for Crossbred, let us look at a toy example.

Example 6.2. For a system of $m = 49$ polynomials and $n = 23$ variables, if we choose $k = 18$, the corresponding series is :

$$G^{18}_{49,23}(X,Y) = -18X - 212X^2 - 104X^2Y - 846X^3 - 558X^3Y + 66X^3Y^2 + \mathcal{O}(X^4).$$

Choosing parameters $(D, d) = (3, 2)$, we will generate 66 polynomials. Assuming these polynomials are linearly independent after assignment of the last $n - k$ variables, we claim that this is not enough to linearize the specialized system. Indeed, since $d = 2$, the Macaulay matrix $Mac_{\leq 2,49}(\mathcal{F}^*)$ has $M^{18}_{\leq 2} = 172$ columns. After adding the 66 new polynomials, this matrix has 115 rows. We conclude that $(D, d) = (3, 2)$ and $k = 18$ are not admissible parameters for this polynomial system.

Let $R' = \mathbb{F}_2[x_1, \ldots, x_k]$. Let $\mathcal{F} = \{f_1, \ldots f_m\}$ be a system of polynomials in R and denote as usual by I the ideal generated by $f_1, \ldots, f_m$. Then we denote by $I^* = \langle f_1^*, \ldots, f_m^* \rangle$, where f_i^* are obtained by assigning the variables $x_{k+1}, \ldots, x_n$ at any values $(a_1, \ldots, a_{n-k}) \in \mathbb{F}_2^{n-k}$.

When $d \geq d_{sol}(k)$, the matrix $Mac_{\leq d,m}(\mathcal{F}^*)$ has more rows than columns and it has full rank. In this case, d is not interesting as input parameter for the Crossbred algorithm since we do not need any new polynomials generated in the pre-processing step. Indeed, in this case it suffices to perform exhaustive search, assign the last $n-k$ variables in the system and solve it (for instance by linearization on the $Mac_{\leq d,m}(\mathcal{F}^*)$ matrix). This leads to the following definition.

Definition 7. Let $\mathcal{F} = \{f_1, \ldots, f_m\}$ be a sequence of polynomials in $\mathbb{F}_2[x_1, \ldots, x_n]$ and k and γ are such that $0 \leq k = (1-\gamma)n \leq n$ and $\mathcal{F}$ is γ-strong semi-regular. The set of parameters (D, d, k) is called potentially admissible for the Crossbred algorithm on $\mathcal{F}$ if the following hold:

(1) $d < d_{sol}(k)$ and $D < D_{reg}$,

(2) For all $(a_{k+1}, \ldots, a_n) \in \mathbb{F}_2^{n-k} \backslash \mathcal{S}(I)$ and the ideal $I^* = \langle f_1^*, \ldots, f_m^* \rangle$ obtained by evaluating $f_1, \ldots, f_m$ at $(a_{k+1}, \ldots, a_n)$ we have that:

$$\sum_{d_1 \leq D, d_2 \geq d} h^k_{d_1,d_2,m} + \dim I^*_{\leq d} \geq \dim R'_{\leq d}.$$

Given $d < d_{reg}(k)$, $D < D_{reg}$ and $\mathcal{F}$ a γ-strong semi-regular sequence, condition (2) of Definition 7 can be written in terms of linear algebra as :

$$corank(\mathcal{M}^k_{\leq D, \geq d, m}(\mathcal{F})) - corank(Mac_{\leq d,m}(\mathcal{F}^*)) \geq 0.$$

Starting from this observation, we compute the generating series which determines a certain range of potentially admissible parameters for the Crossbred algorithm.

Theorem 10. *Let $\mathcal{F} = \{f_1, \ldots, f_m\}$ be a γ-strong semi-regular sequence of polynomials in $\mathbb{F}_2[x_1, \ldots x_n]$ and consider $D < D_{reg}$, $d < d_{reg}(k)$. Then k, D and d are potentially admissible parameters for the Crossbred algorithm if the coefficient corresponding to $X^D Y^d$ of the following bivariate series*

$$J^k_{m,n}(X,Y) = \frac{1}{(1-X)(1-Y)} \left(\frac{(1+X)^{n-k}(1+XY)^k}{(1+X^2Y^2)^m} - \frac{(1+X)^n}{(1+X^2)^m} - \frac{(1+Y)^k}{(1+Y^2)^m} \right) \quad (12)$$

is non-negative.

Proof. To check that condition (2) in Definition 7 is verified, we look at the coefficients of the following bivariate series :

$$G^k_{m,n}(X,Y) - \sum_{D \geq 0} G_{R/I^*}(Y) X^D, \quad (13)$$

where G_{R/I^*} is defined as in Equation (2). For a fixed value of k, under strong semi-regularity assumption, the d-th coefficient in the expansion of the series G_{R/I^*} gives us the difference between the number of columns and the number of rows in $Mac_{\leq d,m}(\mathcal{F}^*)$, while the coefficient before $X^D Y^d$ in the series $G^k_{m,n}$ gives us the number of new polynomials that have been generated during the pre-processing step for parameters D and d, and thus the number of rows that have been appended to $Mac_{\leq d,m}(\mathcal{F}^*)$ in step (3) of the Crossbred algorithm. From Lemma 1, we have:

$$\left[G_{R/I^*}(Y) \right]_{d_{reg}(k)} = \left[\frac{(1+Y)^k}{(1-Y)(1+Y^2)^m} \right]_{d_{reg}(k)}. \quad (14)$$

Table 1: Example with 5 polynomial systems

seed	(D,d)	m	n	k	r	$M^k_{\leq d}$	2^{n-k}	(#Ind. pol., #Iteration)
261	$(4,1)$	59	28	20	108	21	256	$(20,1)$ $(21,255)$
262	$(4,1)$	59	28	20	108	21	256	$(20,1)$ $(21,255)$
263	$(4,1)$	59	28	20	108	21	256	$(20,1)$ $(21,255)$
264	$(4,1)$	59	28	20	108	21	256	$(20,1)$ $(21,255)$
265	$(4,1)$	59	28	20	108	21	256	$(20,2)$ $(21,254)$
-	$(4,1)$	59	28	20	108	21	256	$(20,1.2)$ $(21,254.8)$

Then by replacing $G^k_{m,n}(X,Y)$ with its expression computed in Proposition 9, we get the series claimed in the statement of the theorem. □

We can see that the computed generating series for admissible parameters $J^k_{m,n}(X,Y)$ is the the one claimed by Joux and Vitse in [JV17].

Example 6.3. We revisit Example 6.1. We compute the degree of regularity of a semi-regular sequence of polynomials with $m = 160$ and $n = 80$ and get $D_{reg} = 8$. If the sequence is γ-strong semi-regular, we get that $d_{reg}(24) = 3$. We compute the series $J^k_{m,n}(X,Y)$:

$$\begin{aligned} J^{24}_{160,80}(X,Y) = & - & 24X - 1484X^2 - 141X^2Y - 43124X^3 - 4821X^3Y + 1675X^3Y^2 \\ & - & 764694X^4 - 61111X^4Y + 124025X^4Y^2 + 22329X^4Y^3 \\ & - & 8869694X^5 + 648849X^5Y + 4049505X^5Y^2 + 1151169X^5Y^3 \\ & - & 5455X^5Y^4 + \mathcal{O}(X^6). \end{aligned}$$

The parameters $(D,d) = (3,2)$ and $(D,d) = (5,1)$ are potentially admissible as they satisfy the conditions of Definition 7. By picking $(3,2)$, we will have a less costly pre-processing (in terms of both time and memory), while by choosing $(5,1)$ the resolution during the specialization will be faster. This is impactful since we have to test up to $2^{n-k} = 2^{56}$ values. The choice of optimal parameters depends on the implementation and resources available and is beyond the scope of this paper.

7 Experiments

In this section we show experimental evidence supporting the conjecture that potentially admissible parameters are indeed admissible (see Table 2).

We implemented Algorithm 1 in Magma including the General and the Frobenius criterion when constructing Macaulay matrices (see Prop. 1 and 2) and ran experiments over pseudo-random polynomial systems. Polynomials systems used in the experiments are obtained by using Sedlacek's implementation [Sed22] of Beullens's differential attack on Rainbow instances [DS05, Beu22]. We also experimented with polynomials systems from the Fukuoka Type I MQ challenge [YDH+15, Yas15]. Data in Tables 1, 2 and 3 is obtained by generating polynomials in the pre-processing step of Algorithm 1 for different choices of parameters. In these Tables we use the following notation :

- As usual D, d and k are the input parameters for the algorithm and m and n denote the number of polynomials and the number of variables of the system, respectively.
- r corresponds to the number of polynomials generated by the pre-processing step of the Crossbred algorithm.
- $M^k_{\leq d}$ denotes the number of monomials of degree $\leq d$ over k variables. This is the number of polynomials needed to successfully solve the degree d system obtained after assigning the last $n-k$ variables, by linearization.

Table 2: Experimental data on the Crossbred algorithm for admissible parameters

(D,d)	m	n	k	r	$M^k_{\leq d}$	2^{n-k}	(#Ind. pol., #Iteration)	D_{reg}	$d_{reg}(k)$
$(4,2)$	49	23	18	3608	172	32	$(171,1.3)$ $(172,30.7)$	4	3
$(4,2)$	49	23	17	4130	154	64	$(153,1.1)$ $(154,62.9)$	4	3
$(4,2)$	40	20	17	2240	154	8	$(153,1)$ $(154,7)$	4	3
$(4,1)$	49	23	18	1944	19	32	$(18,1.3)$ $(19,30.7)$	4	3
$(4,1)$	49	23	17	2216	18	64	$(17,1.1)$ $(18,62.9)$	4	3
$(4,1)$	40	20	17	1568	18	8	$(17,1)$ $(18,7)$	4	3
$(3,1)$	47	22	11	256	12	2048	$(11,1.2)$ $(12,2046.8)$	4	3

Table 3: Experimental data on the Crossbred algorithm for non-admissible parameters

(D,d)	m	n	k	r	$M^k_{\leq d}$	2^{n-k}	(#Ind. pol., #Iteration)	D_{reg}	$d_{reg}(k)$
$(3,2)$	49	23	18	66	172	32	$(115,32)$	4	3
$(3,2)$	53	25	19	38	191	64	$(91,64)$	4	3
$(3,2)$	55	26	19	76	191	128	$(131,128)$	4	3
$(3,2)$	57	27	19	114	191	256	$(171,256)$	4	3

Recall that during the exhaustive search step of the algorithm we evaluate the newly generated polynomials in the last $n-k$ variables and add them to the degree d Macaulay matrix of the specialized system $Mac_{\leq d,m}(\mathcal{F}^*)$. Then we count how many independent polynomials there are for each iteration of the exhaustive search. Each of the couples for an entry in the last column of Table 1 gives the number of independent polynomials and the number of iterations of the exhaustive search for which we obtained this value.

In Table 1, we experimented using 5 polynomial systems obtained using the generator in [Sed22], using a different seed each time to ensure that these systems are distinct. As expected, for each of these 5 polynomial systems, the pre-processing step of Algorithm 1 outputs exactly the same numbers of polynomials, which is 108. Since $n = 28$ and $k = 20$, we search through $2^{n-k} = 256$ different values for the last $n-k$ variables. We see that for all possible values, except for one or two, the maximal number of independent polynomial after specialization is $21 = k+1$. This is similar to the test of consistency done in [BFSS13] in the sense that if the ideal I^* has no solution, then $corank(Mac_{\leq 1,59}(\mathcal{F}^*)) = 0$, which means that the matrix has full rank. Otherwise, if the ideal I^* has a solution, then $corank(Mac_{\leq 1,59}(\mathcal{F}^*)) \neq 0$ which implies the matrix will not reach full rank. We see that each seed has one solution except for seed 265 which has two solutions.

The last row in Table 1 computes the average number of independent polynomials obtained after specialization, for this set of 5 polynomial systems.

In Table 2 we re-do the experiment and compute the same average, for different sets of admissible parameters. Whenever $m = 2n$, the data is obtained with polynomials from the Fukuoka MQ challenge. To obtain this data for the Fukuoka MQ challenge, we took the five available polynomial systems available in [Yas15] for any n and m, and computed the average of the result for each system. Every polynomial system gave the same result in the experiment. When $m \neq 2n$, the data is obtained with polynomials systems generated by Sedlacek's implementation. For that, we generated distinct polynomials systems with different seeds. The number of generated polynomials is the same for each seed, which was expected, and the number of solution in each system varies between one or two depending on the seed.

Finally, Table 3 shows similar experiments for non-admissible parameters, (i.e., when $m+r < M^k_{\leq d}$). In this case, we see that the $m+r$ polynomials of degree d are independent after specialization, which confirms our γ-strong semi-regularity hypothesis.

References

[Bar04] Magali Bardet. *Etude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie*. PhD thesis, Université Paris 6, 2004.

[BCC+10] Charles Bouillaguet, Hsieh-Chung Chen, Chen-Mou Cheng, Tung Chou, Ruben Niederhagen, Adi Shamir, and Bo-Yin Yang. Fast exhaustive search for polynomial systems in $\mathbb{F}_2$. In *Cryptographic Hardware and Embedded Systems, CHES 2010*, pages 203–218. Springer Berlin Heidelberg, 2010. doi:10.1007/978-3-642-15031-9_14.

[BCT+24] John Baena, Daniel Cabarcas, Sharwan K. Tiwari, Javier Verbel, and Luis Villota. Admissible parameters for the Crossbred algorithm and semi-regular sequences over finite fields. https://eprint.iacr.org/2024/758, 2024.

[Beu22] Ward Beullens. Breaking Rainbow takes a weekend on a laptop. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, Proceedings, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 464–479. Springer, 2022. doi:10.1007/978-3-031-15979-4_16.

[BFR24] Ryad Benadjila, Thibauld Feneuil, and Matthieu Rivain. MQ on my Mind: Post-Quantum Signatures from the Non-Structured Multivariate Quadratic Problem. *2024 IEEE 9th European Symposium on Security and Privacy (EuroS&P)*, pages 468–485, 2024. URL: https://eprint.iacr.org/2023/1719, doi:https://doi.ieeecomputersociety.org/10.1109/EuroSP60621.2024.00032.

[BFS03] Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. Complexity of Gröbner basis computation for semi-regular overdetermined sequences over $\mathbb{F}_2$ with solutions in $\mathbb{F}_2$. Research Report RR-5049, INRIA, 2003. URL: https://inria.hal.science/inria-00071534.

[BFSS13] Magali Bardet, Jean-Charles Faugère, Bruno Salvy, and Pierre-Jean Spaenlehauer. On the complexity of solving quadratic Boolean systems. *Journal of Complexity*, 29(1):53–75, 2013. doi:10.1016/j.jco.2012.07.001.

[BMSV22] Emanuele Bellini, Rusydi H. Makarim, Carlo Sanna, and Javier Verbel. An estimator for the Hardness of the MQ Problem. In Lejla Batina and Joan Daemen, editors, *Progress in Cryptology - AFRICACRYPT 2022*, pages 323–347. Springer Nature Switzerland, 2022. doi:10.1007/978-3-031-17433-9_14.

[BND+22] Mina Bigdeli, Emanuela De Negri, Manuela M. Dizdarevic, Romy Minko, and Sulamithe Tsakou. Semi-regular sequences and other random systems of equations. In Sorina Ionica Alina Cojocaru and Elisa Lorenzo Garcia, editors, *Women in Numbers Europe III: Research Directions in Number Theory*. Springer Berlin Heidelberg, 2022. doi:10.1007/978-3-030-77700-5_3.

[BP17] Ward Beullens and Bart Preneel. Field Lifting for Smaller UOV Public Keys. In Arpita Patra and Nigel P. Smart, editors, *Progress in Cryptology – INDOCRYPT 2017*, pages 227–246. Springer International Publishing, 2017. doi:10.1007/978-3-319-71667-1_12.

[BPKV24] Luk Bettale, Ludovic Perret, Delaram Kahrobaei, and Javier Verbel. Biscuit: Shorter MPC-based Signature from PoSSo. In Christina Pöpper and Lejla Batina, editors, *Applied Cryptography and Network Security - 22nd International Conference, ACNS 2024, Proceedings, Part I*, volume 14583 of *Lecture Notes in Computer Science*, pages 457–486. Springer, 2024.

[BS23] Charles Bouillaguet and Julia Sauvage. High-Performance Xbred. https://gitlab.lip6.fr/almasty/hpXbred, 2023.

[Buc65] Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* PhD thesis, University of Innsbruck, 1965.

[CFF$^+$23a] Benoît Cogliati, Jean-Charles Faugère, Pierre-Alain Fouque, Louis Goubin, Robin Larrieu, Gilles Macario-Rat, Brice Minaud, and Jacques Patarin. PROV: PRovable unbalanced Oil and Vinegar, 2023. URL: https://prov-sign.github.io.

[CFF$^+$23b] Benoît Cogliati, Jean-Charles Faugère, Pierre-Alain Fouque, Louis Goubin, Robin Larrieu, Gilles Macario-Rat, Brice Minaud, and Jacques Patarin. Vox specification v1.0, 2023. URL: https://vox-sign.com.

[CFMR$^+$17] Antoine Casanova, Jean-Charles Faugere, Gilles Macario-Rat, Jacques Patarin, Ludovic Perret, and Jocelyn Ryckeghem. GeMSS: a great multivariate short signature, 2017.

[CHR$^+$19] Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. MQDSS specifications, 2019. URL: https://mqdss.org/files/MQDSS_Ver2.pdf.

[CKPS00] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In Bart Preneel, editor, *Advances in Cryptology — EUROCRYPT 2000*, pages 392–407. Springer Berlin Heidelberg, 2000. doi:10.1007/3-540-45539-6_27.

[CLO97] David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties and Algorithms.* Springer Berlin, Heidelberg, 1997. doi:10.1007/978-3-319-16721-3.

[DS05] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In John Ioannidis, Angelos Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security*, pages 164–175, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. doi:10.1007/11496137_12.

[Dua23] João Diogo Duarte. On the Complexity and Admissible Parameters of the Crossbred Algorithm in $\mathbb{F}_{q\geq 2}$. https://eprint.iacr.org/2023/1664.pdf, 2023.

[Fau99] Jean-Charles Faugére. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139(1):61–88, 1999. doi:10.1016/S0022-4049(99)00005-5.

[Fau02] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner basis without reduction to zero (F5). In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*, ISSAC '02, pages 75–83, New York, NY, USA, 2002. ACM. doi:10.1145/780506.780516.

[FHI$^+$23] Hiroki Furue, Fumitaka Hoshino, Yasuhiko Ikematsu, Toshiyuki Miyazawa, Akira Nagai, Tsunekazu Saito, Tsuyoshi Takagi, and Kan Yasuda. QR-UOV, 2023. URL: https://info.isl.ntt.co.jp/crypt/qruov/.

[FY79] A.S. Fraenkel and Y. Yesha. Complexity of problems in games, graphs and algebraic equations. *Discrete Applied Mathematics*, 1(1):15–30, 1979. doi:10.1016/0166-218X(79)90012-X.

[JBH$^+$23] Ding Jintai, Gong Boru, Guo Hao, He Xiaoou, Jin Yi, Pan Yuansheng, Dieter Schmidt, Tao Chengdong, Xie Danli, Yang Bo-Yin, and Zhao Ziyu. TUOV: Triangular Unbalanced Oil and Vinegar, 2023. URL: https://www.tuovsig.org/.

[JV17] Antoine Joux and Vanessa Vitse. A Crossbred algorithm for solving boolean polynomial systems. In *Number-Theoretic Methods in Cryptology - NuTMiC 2017*, volume 10737 of *Lecture Notes in Computer Science*, pages 3–21. Springer, 2017. doi:10.1007/978-3-319-76620-1_1.

[KPG99] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar Signature Schemes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, pages 206–222, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. doi:10.1007/3-540-48910-X_15.

[KY24] Momonari Kudo and Kazuhiro Yokoyama. The solving degrees for computing Gröbner bases of affine semi-regular polynomial sequences. https://arxiv.org/abs/2404.03530, 2024.

[KZ20] Daniel Kales and Greg Zaverucha. An attack on some signature schemes constructed from five-pass identification schemes. In Stephan Krenn, Haya Shulman, and Serge Vaudenay, editors, *Cryptology and Network Security - 19th International Conference, CANS 2020, Proceedings*, Lecture Notes in Computer Science, pages 3–22, 2020. doi:10.1007/978-3-030-65411-5_1.

[Laz83] Daniel Lazard. Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. In J.A. van Hulzen, editor, *EUROCAL 1983*, volume 162 of *Lecture Notes in Computer Science*, page 146–156. Springer, Heidelberg, 1983. doi:10.1007/3-540-12868-9_99.

[MI88] Tsutomu Matsumoto and Hideki Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In D. et al. Barstow, editor, *Advances in Cryptology — EUROCRYPT '88*, pages 419–453. Springer Berlin Heidelberg, 1988. doi:10.1007/3-540-45961-8_39.

[MIS20] Koksal Mus, Saad Islam, and Berk Sunar. QuantumHammer: A practical hybrid attack on the LUOV signature scheme. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1071–1084. ACM, 2020. doi:https://dl.acm.org/doi/10.1145/3372297.3417272.

[Nak24] Shuhei Nakamura. Admissible parameter sets and complexity estimation of Crossbred algorithm. https://eprint.iacr.org/2023/1687.pdf, 2024.

[NNY17] Ruben Niederhagen, Kai-Chun Ning, and Bo-Yin Yang. https://github.com/kcning/mqsolver, 2017.

[NNY18] Ruben Niederhagen, Kai-Chun Ning, and Bo-Yin Yang. Implementing Joux-Vitse's Crossbred Algorithm for Solving MQ Systems over GF(2) on GPUs. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018, Proceedings*, volume 10786 of *Lecture Notes in Computer Science*, pages 121–141. Springer, 2018. doi:10.1007/978-3-319-79063-3_6.

[Pat95] Jacques Patarin. Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88. In Don Coppersmith, editor, *Advances in Cryptology — CRYPT0' 95*, pages 248–261. Springer Berlin Heidelberg, 1995. doi:10.1007/3-540-44750-4_20.

[Sed22] Vladimír Sedláček. mq-comparaison-suite. https://github.com/VladaSedlacek/mq-comparison-suite, 2022.

[TPD21] Chengdong Tao, Albrecht Petzoldt, and Jintai Ding. Efficient key recovery for all HFE signature variants. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, pages 70–93, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-84242-0_4.

[TW12] Enrico Thomae and Christopher Wolf. Solving underdetermined systems of multivariate quadratic equations revisited. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *Public Key Cryptography – PKC 2012*, pages 156–171. Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-30057-8_10.

[WTKC22] Lih-Chung Wang, Po-En Tseng, Yen-Liang Kuan, and Chun-Yen Chou. A simple noncommutative UOV scheme, 2022. URL: https://eprint.iacr.org/2022/1742.

[Yas15] Takanori Yasuda. Fukuoka MQ Challenge. https://www.mqchallenge.org, 2015.

[YDH+15] Takanori Yasuda, Xavier Dahan, Yun-Ju Huang, Tsuyoshi Takagi, and Kouichi Sakurai. MQ Challenge: Hardness Evaluation of Solving Multivariate Quadratic Problems. https://eprint.iacr.org/2015/275, 2015.

A Proof of Lemma 1

Proof. Since $\mathcal{F}$ is affine semi-regular, from Definition 3 the homogeneous sequence $\mathcal{F}^{top} = \{f_1^{top}, \ldots, f_m^{top}\}$ is semi-regular in R_n^h. We denote by D_{reg} the degree of regularity of $\mathcal{F}^{top}$. Since there are no falls of degree when echelonizing the matrices $Mac_{\leq d,m}(\mathcal{F})$, for $d < D_{reg}$ ([Bar04, Section 3.5]), the truncation up to D_{reg} of the expansion of the series $G_{R/I}(X)$ coincides to that of the following series:

$$\tilde{G}_{R/I}(X) = \sum_{d \geq 0} \big(\#\mathsf{Col}(Mac_{\leq d,m}(\mathcal{F})) - \#\mathsf{Row}(Mac_{\leq d,m}(\mathcal{F}))\big) X^d,$$

where $\#\mathsf{Row}$ and $\#\mathsf{Col}$ represent respectively the number of rows and of columns of the matrix given in argument.

We denote by M_d the number of monomials of degree d in R_n^h. Note that

$$\#\mathsf{Col}(Mac_{\leq d,m}(\mathcal{F})) = \sum_{d' \leq d} M_{d'}.$$

We denote by $\Delta_d(\mathcal{F}) = M_d - \#\mathsf{Row}(Mac_{d,m}(\mathcal{F}))$ and compute $\tilde{G}_{R/I}(X)$ as follows:

$$\begin{aligned}\tilde{G}_{R/I}(X) &= \sum_{d\geq 0}\left(\sum_{d'\leq d}\Delta_{d'}(\mathcal{F})\right)X^d\\ &= \sum_{d\geq 0}\Delta_d(\mathcal{F})X^d + X\sum_{d\geq 0}\left(\sum_{d'\leq d}\Delta_{d'}(\mathcal{F})\right)X^d.\end{aligned}$$

We consider the series:

$$\widetilde{HF}_{R/I^{top}}(X) = \sum_{0\leq d}\Delta_d(\mathcal{F})X^d = \sum_{0\leq d}\big(M_d - \#\mathsf{Row}(Mac_{d,m}(\mathcal{F}))\big)X^d,$$

Since $\mathcal{F}^{top} = \{f_1^{top},\ldots,f_m^{top}\}$ is semi-regular in R_n^h, $[\widetilde{HF}_{R/I^{top}}]_{D_{reg}}$ is exactly the Hilbert series of $\mathcal{F}^{top}$. We have:

$$\tilde{G}_{R/I}(X) = \widetilde{HF}_{R/I^{top}}(X) + X\tilde{G}_{R/I}(X).$$

Hence

$$\tilde{G}_{R/I}(X) = \frac{\widetilde{HF}_{R/I^{top}}(X)}{(1-X)}.$$

Then, replacing $[\widetilde{HF}_{R/I^{top}}(X)]_{D_{reg}}$ by the expression given in Equation (1) concludes the proof. □

B γ-strong semi-regularity

In this appendix, we show experimental evidence supporting the claim that random polynomials sequences are γ-strong semi-regular, for small enough $\gamma = 1 - k/n$. In particular, in our experiments, random polynomial sequences, which we know to be semi-regular [BFS03], remain semi-regular, for all assignment of the last $n-k$ variables, when γ is smaller than an upper bound that we give below.

To test the semi-regularity of the specialized sequence $\mathcal{F}^*$, we compute the Macaulay matrix $Mac_{d,m}((\mathcal{F}^*)^{top})$, for all d up to $d_{reg}(k)$. If there is no other reduction to zero between the rows of this matrix than the ones arising from the criteria when $d < d_{reg}(k)$, then $\mathcal{F}^*$ is semi-regular.

In Table 4 we show that experimental results confirm our assumption. In the fourth column of this Table, we give the value of $d_{reg}(k)$, computed using the series in Equation (1). In the fifth column, we give the value of γ, rounded with three decimals. In the seventh column, we compute the numbers of rows and columns of $Mac_{d,m}((\mathcal{F}^*)^{top})$ and its rank, for successive values of d. For a fixed value of d, the rank did not change depending on the specialisation. Since the matrix has less rows than columns and it has full rank for $d < d_{reg}(k)$, for all possible values of the last $n-k$ variables, we conclude that all $\mathcal{F}^*$ are semi-regular. Thus, $\mathcal{F}$ is γ-strong semi-regular.

Note that this does not hold for every value of γ. Indeed, if k is too small and the number of rows of $Mac_{2,m}((\mathcal{F}^*)^{top})$ is greater than its number of columns, then condition 1. of Definition 2 is not fulfilled. Hence the sequence $\mathcal{F}^*$ is not semi-regular. That leads us to compute a lower bound on k (which is equivalent to an upper bound on γ). To find it, we search k such that the number of columns of $Mac_{2,m}((\mathcal{F}^*)^{top})$ is less that the number of rows, which gives the following inequality:

$$\binom{k}{2} < m. \tag{15}$$

This inequality has one positive root k_1, which yields a lower bound on the values of k for which $\mathcal{F}$ is γ-strong semi-regular series. For $m = 49$, this is equal to $k_1 \approx 10.41$. As such, for a sequence $\mathcal{F}$ of 49 polynomials and 23 variables, if $k \leq 10$ (which corresponds to $\gamma \approx 0.565$), then $\mathcal{F}$ is not γ-strong semi-regular as the specialized sequence is not semi-regular.

Table 4: Experimental data for γ-strong semi-regularity

m	n	k	$d_{reg}(k)$	γ	d	(# rows, # columns)	Rank of $Mac_{d,m}((\mathcal{F}^*)^{top})$
49	23	18	3	0.217	2	$(\mathbf{49}, 153)$	**49**
					3	$(882, \mathbf{816})$	**816**
49	23	17	3	0.261	2	$(\mathbf{49}, 136)$	**49**
					3	$(833, \mathbf{680})$	**680**
49	23	12	3	0.478	2	$(\mathbf{49}, 66)$	**49**
					3	$(588, \mathbf{220})$	**220**
53	25	19	3	0.24	2	$(\mathbf{53}, 171)$	**53**
					3	$(1007, \mathbf{969})$	**969**
59	28	25	4	0.107	2	$(\mathbf{59}, 300)$	**59**
					3	$(\mathbf{1475}, 2300)$	**1475**
					4	$(36875, \mathbf{12650})$	**12650**
59	28	18	3	0.357	2	$(\mathbf{59}, 153)$	**59**
					3	$(1062, \mathbf{816})$	**816**