Re-linearization and elimination of variables in Boolean equation systems

Bjørn Møller Greve[1,2] Håvard Raddum[2] Øyvind Ytrehus[2]

[1]Norwegian Defence Research Establishment

[2]Simula@UiB

4 September, 2017

simula@uib

FFI Forsvarets forskningsinstitutt
Norwegian Defence Research Establishment

simula@uib

Eliminating variables in Boolean equation systems

Elimination of variables from Boolean functions

- Consider the Boolean ring $B[1,n] = \mathbb{F}_2[x_1,\ldots,x_n]/(x_i^2 + x_i | i = 1,\ldots,n)$
- $$\begin{matrix} f_1(x_1,\ldots,x_n) = 0 \\ \vdots \\ f_m(x_1,\ldots,x_n) = 0 \end{matrix} \quad \longrightarrow \quad \begin{matrix} f_1'(x_2,\ldots,x_n) = 0 \\ \vdots \\ f_m'(x_2,\ldots,x_n) = 0 \end{matrix}$$
- Eliminate x_1 s.th $(a_1,\ldots,a_n)$ solution in left system $\Longrightarrow$ $(a_2,\ldots,a_n)$ is solution in right system.

Applications to ciphers

- Describe cipher as quadratic Boolean equation system.
- Variables: Secret key K + auxiliary variables (To keep equations simple)
- Is it possible to eliminate auxiliary variables and find some equations in only key variables?

Eliminating variables in Boolean equation systems

Elimination of variables from Boolean functions

- Consider the Boolean ring $B[1, n] = \mathbb{F}_2[x_1, \ldots, x_n]/(x_i^2 + x_i | i = 1, \ldots, n)$
- $$\begin{matrix} f_1(x_1, \ldots, x_n) = 0 & & f_1'(x_2, \ldots, x_n) = 0 \\ \vdots & \longrightarrow & \vdots \\ f_m(x_1, \ldots, x_n) = 0 & & f_m'(x_2, \ldots, x_n) = 0 \end{matrix}$$
- Eliminate x_1 s.th $(a_1, \ldots, a_n)$ solution in left system $\Longrightarrow$ $(a_2, \ldots, a_n)$ is solution in right system.

Applications to ciphers

- Describe cipher as quadratic Boolean equation system.
- Variables: Secret key K + auxiliary variables (To keep equations simple)
- Is it possible to eliminate auxiliary variables and find some equations in only key variables?

Eliminating variables in Boolean equation systems

Elimination of variables from Boolean functions

- Consider the Boolean ring $B[1,n] = \mathbb{F}_2[x_1,\ldots,x_n]/(x_i^2 + x_i | i = 1,\ldots,n)$
-

$$\begin{array}{ccc} f_1(x_1,\ldots,x_n) = 0 & & f_1'(x_2,\ldots,x_n) = 0 \\ \vdots & \longrightarrow & \vdots \\ f_m(x_1,\ldots,x_n) = 0 & & f_m'(x_2,\ldots,x_n) = 0 \end{array}$$

- Eliminate x_1 s.th $(a_1,\ldots,a_n)$ solution in left system $\Longrightarrow$ $(a_2,\ldots,a_n)$ is solution in right system.

Applications to ciphers

- Describe cipher as quadratic Boolean equation system.
- Variables: Secret key K + auxiliary variables (To keep equations simple)
- Is it possible to eliminate auxiliary variables and find some equations in only key variables?

Eliminating variables in Boolean equation systems

Elimination of variables from Boolean functions

- Consider the Boolean ring $B[1,n] = \mathbb{F}_2[x_1,\ldots,x_n]/(x_i^2 + x_i | i = 1,\ldots,n)$
-
$$\begin{array}{ccc} f_1(x_1,\ldots,x_n) = 0 & & f_1'(x_2,\ldots,x_n) = 0 \\ \vdots & \longrightarrow & \vdots \\ f_m(x_1,\ldots,x_n) = 0 & & f_m'(x_2,\ldots,x_n) = 0 \end{array}$$
- Eliminate x_1 s.th $(a_1,\ldots,a_n)$ solution in left system $\Longrightarrow$ $(a_2,\ldots,a_n)$ is solution in right system.

Applications to ciphers

- Describe cipher as quadratic Boolean equation system.
- Variables: Secret key K + auxiliary variables (To keep equations simple)
- Is it possible to eliminate auxiliary variables and find some equations in only key variables?

Eliminating variables in Boolean equation systems

Elimination of variables from Boolean functions

- Consider the Boolean ring $B[1,n] = \mathbb{F}_2[x_1,\ldots,x_n]/(x_i^2 + x_i | i = 1,\ldots,n)$
- $$\begin{matrix} f_1(x_1,\ldots,x_n) = 0 & & f_1'(x_2,\ldots,x_n) = 0 \\ \vdots & \longrightarrow & \vdots \\ f_m(x_1,\ldots,x_n) = 0 & & f_m'(x_2,\ldots,x_n) = 0 \end{matrix}$$
- Eliminate x_1 s.th $(a_1,\ldots,a_n)$ solution in left system $\Longrightarrow$ $(a_2,\ldots,a_n)$ is solution in right system.

Applications to ciphers

- Describe cipher as quadratic Boolean equation system.
- Variables: Secret key K + auxiliary variables (To keep equations simple)
- Is it possible to eliminate auxiliary variables and find some equations in only key variables?

Eliminating variables in Boolean equation systems

Elimination of variables from Boolean functions

- Consider the Boolean ring $B[1,n] = \mathbb{F}_2[x_1,\ldots,x_n]/(x_i^2 + x_i | i = 1,\ldots,n)$
- $$\begin{array}{ccc} f_1(x_1,\ldots,x_n) = 0 & & f_1'(x_2,\ldots,x_n) = 0 \\ \vdots & \longrightarrow & \vdots \\ f_m(x_1,\ldots,x_n) = 0 & & f_m'(x_2,\ldots,x_n) = 0 \end{array}$$
- Eliminate x_1 s.th $(a_1,\ldots,a_n)$ solution in left system $\Longrightarrow$ $(a_2,\ldots,a_n)$ is solution in right system.

Applications to ciphers

- Describe cipher as quadratic Boolean equation system.
- Variables: Secret key K + auxiliary variables (To keep equations simple)
- Is it possible to eliminate auxiliary variables and find some equations in only key variables?

Eliminating variables in Boolean equation systems

Elimination of variables from Boolean functions

- Consider the Boolean ring $B[1,n] = \mathbb{F}_2[x_1,\ldots,x_n]/(x_i^2 + x_i | i = 1,\ldots,n)$
- $$\begin{array}{ccc} f_1(x_1,\ldots,x_n) = 0 & & f_1^{'}(x_2,\ldots,x_n) = 0 \\ \vdots & \longrightarrow & \vdots \\ f_m(x_1,\ldots,x_n) = 0 & & f_m^{'}(x_2,\ldots,x_n) = 0 \end{array}$$
- Eliminate x_1 s.th $(a_1,\ldots,a_n)$ solution in left system $\Longrightarrow$ $(a_2,\ldots,a_n)$ is solution in right system.

Applications to ciphers

- Describe cipher as quadratic Boolean equation system.
- Variables: Secret key K + auxiliary variables (To keep equations simple)
- Is it possible to eliminate auxiliary variables and find some equations in only key variables?

If we are so lucky to find any (low degree) polynomials after elimination

The general method:

- Save intermediate systems after each elimination.
- Brute force possible solutions of final system, lift through intermediate systems to filter out false solutions.

The block cipher method:

Repeating the process of variable elimination using other known plaintext/ciphertext pairs and build up a low-degree system of equations in only user-selected key variables that has K as a unique solution.

Re-linearization

Solve by re-linearization if we can generate more linearly independent polynomials (in some acceptable degree) than there are monomials.

If we are so lucky to find any (low degree) polynomials after elimination

The general method:

- Save intermediate systems after each elimination.
- Brute force possible solutions of final system, lift through intermediate systems to filter out false solutions.

The block cipher method:

Repeating the process of variable elimination using other known plaintext/ciphertext pairs and build up a low-degree system of equations in only user-selected key variables that has K as a unique solution.

Re-linearization

Solve by re-linearization if we can generate more linearly independent polynomials (in some acceptable degree) than there are monomials.

If we are so lucky to find any (low degree) polynomials after elimination

The general method:

- Save intermediate systems after each elimination.
- Brute force possible solutions of final system, lift through intermediate systems to filter out false solutions.

The block cipher method:

Repeating the process of variable elimination using other known plaintext/ciphertext pairs and build up a low-degree system of equations in only user-selected key variables that has K as a unique solution.

Re-linearization

Solve by re-linearization if we can generate more linearly independent polynomials (in some acceptable degree) than there are monomials.

If we are so lucky to find any (low degree) polynomials after elimination

The general method:

- Save intermediate systems after each elimination.
- Brute force possible solutions of final system, lift through intermediate systems to filter out false solutions.

The block cipher method:

Repeating the process of variable elimination using other known plaintext/ciphertext pairs and build up a low-degree system of equations in only user-selected key variables that has K as a unique solution.

Re-linearization

Solve by re-linearization if we can generate more linearly independent polynomials (in some acceptable degree) than there are monomials.

If we are so lucky to find any (low degree) polynomials after elimination

The general method:

- Save intermediate systems after each elimination.
- Brute force possible solutions of final system, lift through intermediate systems to filter out false solutions.

The block cipher method:

Repeating the process of variable elimination using other known plaintext/ciphertext pairs and build up a low-degree system of equations in only user-selected key variables that has K as a unique solution.

Re-linearization

Solve by re-linearization if we can generate more linearly independent polynomials (in some acceptable degree) than there are monomials.

If we are so lucky to find any (low degree) polynomials after elimination

The general method:

- Save intermediate systems after each elimination.
- Brute force possible solutions of final system, lift through intermediate systems to filter out false solutions.

The block cipher method:

Repeating the process of variable elimination using other known plaintext/ciphertext pairs and build up a low-degree system of equations in only user-selected key variables that has K as a unique solution.

Re-linearization

Solve by re-linearization if we can generate more linearly independent polynomials (in some acceptable degree) than there are monomials.

If we are so lucky to find any (low degree) polynomials after elimination

The general method:

- Save intermediate systems after each elimination.
- Brute force possible solutions of final system, lift through intermediate systems to filter out false solutions.

The block cipher method:

Repeating the process of variable elimination using other known plaintext/ciphertext pairs and build up a low-degree system of equations in only user-selected key variables that has K as a unique solution.

Re-linearization

Solve by re-linearization if we can generate more linearly independent polynomials (in some acceptable degree) than there are monomials.

Previous work

GRFY (BFA 2017)

Elimination algorithm with degree restriction $\deg(f_i) \leq 3$.

"Naive" XL elimination

- Multiply each f_i with all monomials respecting degree restriction $\Rightarrow$ New polynomial set F.
- Gaussian elimination on F eliminating all monomials containing x_1.

Theorem

- *GRFY* elimination $=$ XL elimination when restricting the degree to ≤ 3.
- In general: Extended *GRFY* elimination $\supset$ XL elimination when restricting the degree to ≤ 3.
- In general extended *GRFY* elimination introduces less false solutions than the naive XL method when restricting the degree to ≤ 3.

Previous work

GRFY (BFA 2017)

Elimination algorithm with degree restriction $\deg(f_i) \leq 3$.

"Naive" XL elimination

- Multiply each f_i with all monomials respecting degree restriction $\Rightarrow$ New polynomial set F.
- Gaussian elimination on F eliminating all monomials containing x_1.

Theorem

- *GRFY* elimination $=$ XL elimination when restricting the degree to ≤ 3.
- In general: Extended *GRFY* elimination $\supset$ XL elimination when restricting the degree to ≤ 3.
- In general extended *GRFY* elimination introduces less false solutions than the naive XL method when restricting the degree to ≤ 3.

Previous work

GRFY (BFA 2017)

Elimination algorithm with degree restriction $\deg(f_i) \leq 3$.

"Naive" XL elimination

- Multiply each f_i with all monomials respecting degree restriction $\Rightarrow$ New polynomial set F.
- Gaussian elimination on F eliminating all monomials containing x_1.

Theorem

- *GRFY* elimination $=$ XL elimination when restricting the degree to ≤ 3.
- In general: Extended *GRFY* elimination $\supset$ XL elimination when restricting the degree to ≤ 3.
- In general extended *GRFY* elimination introduces less false solutions than the naive XL method when restricting the degree to ≤ 3.

Previous work

GRFY (BFA 2017)

Elimination algorithm with degree restriction $\deg(f_i) \leq 3$.

"Naive" XL elimination

- Multiply each f_i with all monomials respecting degree restriction $\Rightarrow$ New polynomial set F.
- Gaussian elimination on F eliminating all monomials containing x_1.

Theorem

- *GRFY* elimination = XL elimination when restricting the degree to ≤ 3.
- In general: Extended *GRFY* elimination $\supset$ XL elimination when restricting the degree to ≤ 3.
- In general extended *GRFY* elimination introduces less false solutions than the naive XL method when restricting the degree to ≤ 3.

Previous work

GRFY (BFA 2017)

Elimination algorithm with degree restriction $\deg(f_i) \leq 3$.

"Naive" XL elimination

- Multiply each f_i with all monomials respecting degree restriction $\Rightarrow$ New polynomial set F.
- Gaussian elimination on F eliminating all monomials containing x_1.

Theorem

- *G*RFY elimination = XL elimination when restricting the degree to ≤ 3.
- In general: Extended *GRFY* elimination $\supset$ XL elimination when restricting the degree to ≤ 3.
- In general extended *GRFY* elimination introduces less false solutions than the naive XL method when restricting the degree to ≤ 3.

Previous work

GRFY (BFA 2017)

Elimination algorithm with degree restriction $\deg(f_i) \leq 3$.

"Naive" XL elimination

- Multiply each f_i with all monomials respecting degree restriction $\Rightarrow$ New polynomial set F.
- Gaussian elimination on F eliminating all monomials containing x_1.

Theorem

- *G*RFY elimination $=$ XL elimination when restricting the degree to ≤ 3.
- In general: Extended *G*RFY elimination $\supset$ XL elimination when restricting the degree to ≤ 3.
- In general extended *GRFY* elimination introduces less false solutions than the naive XL method when restricting the degree to ≤ 3.

Previous work

GRFY (BFA 2017)

Elimination algorithm with degree restriction $\deg(f_i) \leq 3$.

"Naive" XL elimination

- Multiply each f_i with all monomials respecting degree restriction $\Rightarrow$ New polynomial set F.
- Gaussian elimination on F eliminating all monomials containing x_1.

Theorem

- *G*RFY elimination $=$ XL elimination when restricting the degree to ≤ 3.
- In general: Extended *G*RFY elimination $\supset$ XL elimination when restricting the degree to ≤ 3.
- In general extended *G*RFY elimination introduces less false solutions than the naive XL method when restricting the degree to ≤ 3.

Generalizations

Main idea

- Allow more computational complexity when eliminating variables $\rightarrow$ fixing the degree at a chosen parameter $d \geq 3$.
- F^d ={polynomials of deg d}, F^{d-1} ={pols. of deg $d-1$},$\dots$, F^1 ={linear polynomials}.

Objective

- Eliminate $x_1, \dots$, only computing with polynomials of degree d or less.
- $L^0 = \{1\}$, $L^1 = \{x_1, \dots, x_n\}, \dots, L^i$ ={monomials of degree i}.
- Bounding degree $d \rightarrow$ form any product of the form $L^i F^j = \{lf, l \in L^i, f \in F^j\}$ as long as $i + j \leq d$.
- Eliminate variables from the vectorspace $\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \dots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$.

Generalizations

Main idea

- Allow more computational complexity when eliminating variables → fixing the degree at a chosen parameter $d \geq 3$.
- F^d ={polynomials of deg d}, F^{d-1} ={pols. of deg $d-1$},..., F^1 ={linear polynomials}.

Objective

- Eliminate $x_1, \ldots$, only computing with polynomials of degree d or less.
- $L^0 = \{1\}$, $L^1 = \{x_1, \ldots, x_n\}, \ldots, L^i$ ={monomials of degree i}.
- Bounding degree d → form any product of the form $L^i F^j = \{lf, l \in L^i, f \in F^j\}$ as long as $i + j \leq d$.
- Eliminate variables from the vectorspace $\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$.

Generalizations

Main idea

- Allow more computational complexity when eliminating variables $\rightarrow$ fixing the degree at a chosen parameter $d \geq 3$.
- F^d ={polynomials of deg d}, F^{d-1} ={pols. of deg $d-1$},$\ldots$, F^1 ={linear polynomials}.

Objective

- Eliminate $x_1, \ldots$, only computing with polynomials of degree d or less.
- $L^0 = \{1\}$, $L^1 = \{x_1, \ldots, x_n\}, \ldots, L^i$ ={monomials of degree i}.
- Bounding degree $d \rightarrow$ form any product of the form $L^iF^j = \{lf, l \in L^i, f \in F^j\}$ as long as $i + j \leq d$.
- Eliminate variables from the vectorspace $\langle F^d \cup L^1F^{d-1} \cup L^2F^{d-2} \cup \cdots \cup L^{d-2}F^2 \cup L^{d-1}F^1 \rangle$.

Generalizations

Main idea

- Allow more computational complexity when eliminating variables $\rightarrow$ fixing the degree at a chosen parameter $d \geq 3$.
- F^d ={polynomials of deg d}, F^{d-1} ={pols. of deg $d-1$},$\ldots$, F^1 ={linear polynomials}.

Objective

- Eliminate $x_1, \ldots$, only computing with polynomials of degree d or less.
- $L^0 = \{1\}$, $L^1 = \{x_1, \ldots, x_n\}, \ldots, L^i$ ={monomials of degree i}.
- Bounding degree $d \rightarrow$ form any product of the form $L^iF^j = \{lf, l \in L^i, f \in F^j\}$ as long as $i + j \leq d$.
- Eliminate variables from the vectorspace $\langle F^d \cup L^1F^{d-1} \cup L^2F^{d-2} \cup \cdots \cup L^{d-2}F^2 \cup L^{d-1}F^1 \rangle$.

Generalizations

Main idea

- Allow more computational complexity when eliminating variables $\rightarrow$ fixing the degree at a chosen parameter $d \geq 3$.
- F^d ={polynomials of deg d}, F^{d-1} ={pols. of deg $d-1$},$\ldots$, F^1 ={linear polynomials}.

Objective

- Eliminate $x_1, \ldots$, only computing with polynomials of degree d or less.
- $L^0 = \{1\}$, $L^1 = \{x_1, \ldots, x_n\}, \ldots, L^i$ ={monomials of degree i}.
- Bounding degree $d \rightarrow$ form any product of the form $L^i F^j = \{lf, l \in L^i, f \in F^j\}$ as long as $i + j \leq d$.
- Eliminate variables from the vectorspace $\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$.

Generalizations

Main idea

- Allow more computational complexity when eliminating variables $\rightarrow$ fixing the degree at a chosen parameter $d \geq 3$.
- F^d ={polynomials of deg d}, F^{d-1} ={pols. of deg $d-1$},$\ldots$, F^1 ={linear polynomials}.

Objective

- Eliminate $x_1, \ldots$, only computing with polynomials of degree d or less.
- $L^0 = \{1\}$, $L^1 = \{x_1, \ldots, x_n\}, \ldots, L^i$ ={monomials of degree i}.
- Bounding degree $d \rightarrow$ form any product of the form $L^i F^j = \{lf, l \in L^i, f \in F^j\}$ as long as $i + j \leq d$.
- Eliminate variables from the vectorspace $\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$.

Generalizations

Main idea

- Allow more computational complexity when eliminating variables $\rightarrow$ fixing the degree at a chosen parameter $d \geq 3$.
- F^d ={polynomials of deg d}, F^{d-1} ={pols. of deg $d-1$},$\ldots$, F^1 ={linear polynomials}.

Objective

- Eliminate $x_1, \ldots$, only computing with polynomials of degree d or less.
- $L^0 = \{1\}$, $L^1 = \{x_1, \ldots, x_n\}, \ldots, L^i$ ={monomials of degree i}.
- Bounding degree $d \rightarrow$ form any product of the form $L^i F^j = \{lf, l \in L^i, f \in F^j\}$ as long as $i + j \leq d$.
- Eliminate variables from the vectorspace $\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$.

The monomial orders

A. "Naive" XL elimination Monomials containing x_1 are largest

For each $i = \{1 \ldots, d\}$, Gaussian elimination on $F^i \cup L^1 F^{i-1} \cup \ldots \cup L^{i-2} F^2 \cup L^{i-1} F^1$ to eliminate all monomials containing x_1.

B. Ordering the monomials with respect to degree

- For each $i = \{1 \ldots, d\}$, $\langle F^i \cup L^1 F^{i-1} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$ may contain more polynomials of degree $< i$.
- We can try to produce a larger set of polynomials $F^{i-1,(2)}, \ldots, F^{1,(2)}$ by Gaussian elimination with respect to degree. I.e $F^{i-1} \subseteq F^{i-1,(2)}, \ldots, F^1 \subseteq F^{1,(2)}$.

Normal forms

- Enable us to eliminate particular monomials containing x_1 from each F^i using the lower degree sets $F^{i-1}, \ldots, F^2, F^1$ as basis.
- The effect of normalization is that there is a rather large set of monomials containing x_1 that can not appear in each set F^i at the end.

The monomial orders

A. "Naive" XL elimination Monomials containing x_1 are largest

For each $i = \{1 \ldots, d\}$, Gaussian elimination on $F^i \cup L^1 F^{i-1} \cup \ldots \cup L^{i-2} F^2 \cup L^{i-1} F^1$ to eliminate all monomials containing x_1.

B. Ordering the monomials with respect to degree

- For each $i = \{1 \ldots, d\}$, $\langle F^i \cup L^1 F^{i-1} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$ may contain more polynomials of degree $< i$.
- We can try to produce a larger set of polynomials $F^{i-1,(2)}, \ldots, F^{1,(2)}$ by Gaussian elimination with respect to degree. I.e $F^{i-1} \subseteq F^{i-1,(2)}, \ldots, F^1 \subseteq F^{1,(2)}$.

Normal forms

- Enable us to eliminate particular monomials containing x_1 from each F^i using the lower degree sets $F^{i-1}, \ldots, F^2, F^1$ as basis.
- The effect of normalization is that there is a rather large set of monomials containing x_1 that can not appear in each set F^i at the end.

The monomial orders

A. "Naive" XL elimination Monomials containing x_1 are largest

For each $i = \{1 \ldots, d\}$, Gaussian elimination on $F^i \cup L^1 F^{i-1} \cup \ldots \cup L^{i-2} F^2 \cup L^{i-1} F^1$ to eliminate all monomials containing x_1.

B. Ordering the monomials with respect to degree

- For each $i = \{1 \ldots, d\}$, $\langle F^i \cup L^1 F^{i-1} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$ may contain more polynomials of degree $< i$.
- We can try to produce a larger set of polynomials $F^{i-1,(2)}, \ldots, F^{1,(2)}$ by Gaussian elimination with respect to degree. I.e $F^{i-1} \subseteq F^{i-1,(2)}, \ldots, F^1 \subseteq F^{1,(2)}$.

Normal forms

- Enable us to eliminate particular monomials containing x_1 from each F^i using the lower degree sets $F^{i-1}, \ldots, F^2, F^1$ as basis.
- The effect of normalization is that there is a rather large set of monomials containing x_1 that can not appear in each set F^i at the end.

The monomial orders

A. "Naive" XL elimination Monomials containing x_1 are largest

For each $i = \{1 \ldots, d\}$, Gaussian elimination on $F^i \cup L^1 F^{i-1} \cup \ldots \cup L^{i-2} F^2 \cup L^{i-1} F^1$ to eliminate all monomials containing x_1.

B. Ordering the monomials with respect to degree

- For each $i = \{1 \ldots, d\}$, $\langle F^i \cup L^1 F^{i-1} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$ may contain more polynomials of degree $< i$.
- We can try to produce a larger set of polynomials $F^{i-1,(2)}, \ldots, F^{1,(2)}$ by Gaussian elimination with respect to degree. I.e $F^{i-1} \subseteq F^{i-1,(2)}, \ldots, F^1 \subseteq F^{1,(2)}$.

Normal forms

- Enable us to eliminate particular monomials containing x_1 from each F^i using the lower degree sets $F^{i-1}, \ldots, F^2, F^1$ as basis.
- The effect of normalization is that there is a rather large set of monomials containing x_1 that can not appear in each set F^i at the end.

The monomial orders

A. "Naive" XL elimination Monomials containing x_1 are largest

For each $i = \{1 \ldots, d\}$, Gaussian elimination on $F^i \cup L^1 F^{i-1} \cup \ldots \cup L^{i-2} F^2 \cup L^{i-1} F^1$ to eliminate all monomials containing x_1.

B. Ordering the monomials with respect to degree

- For each $i = \{1 \ldots, d\}$, $\langle F^i \cup L^1 F^{i-1} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$ may contain more polynomials of degree $< i$.
- We can try to produce a larger set of polynomials $F^{i-1,(2)}, \ldots, F^{1,(2)}$ by Gaussian elimination with respect to degree. I.e $F^{i-1} \subseteq F^{i-1,(2)}, \ldots, F^1 \subseteq F^{1,(2)}$.

Normal forms

- Enable us to eliminate particular monomials containing x_1 from each F^i using the lower degree sets $F^{i-1}, \ldots, F^2, F^1$ as basis.
- The effect of normalization is that there is a rather large set of monomials containing x_1 that can not appear in each set F^i at the end.

The monomial orders

A. "Naive" XL elimination Monomials containing x_1 are largest

For each $i = \{1 \ldots, d\}$, Gaussian elimination on $F^i \cup L^1 F^{i-1} \cup \ldots \cup L^{i-2} F^2 \cup L^{i-1} F^1$ to eliminate all monomials containing x_1.

B. Ordering the monomials with respect to degree

- For each $i = \{1 \ldots, d\}$, $\langle F^i \cup L^1 F^{i-1} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$ may contain more polynomials of degree $< i$.
- We can try to produce a larger set of polynomials $F^{i-1,(2)}, \ldots, F^{1,(2)}$ by Gaussian elimination with respect to degree. I.e $F^{i-1} \subseteq F^{i-1,(2)}, \ldots, F^1 \subseteq F^{1,(2)}$.

Normal forms

- Enable us to eliminate particular monomials containing x_1 from each F^i using the lower degree sets $F^{i-1}, \ldots, F^2, F^1$ as basis.
- The effect of normalization is that there is a rather large set of monomials containing x_1 that can not appear in each set F^i at the end.

The monomial orders

A. "Naive" XL elimination Monomials containing x_1 are largest

For each $i = \{1 \ldots, d\}$, Gaussian elimination on
$F^i \cup L^1 F^{i-1} \cup \ldots \cup L^{i-2} F^2 \cup L^{i-1} F^1$ to eliminate all monomials containing x_1.

B. Ordering the monomials with respect to degree

- For each $i = \{1 \ldots, d\}$, $\langle F^i \cup L^1 F^{i-1} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$ may contain more polynomials of degree $< i$.
- We can try to produce a larger set of polynomials $F^{i-1,(2)}, \ldots, F^{1,(2)}$ by Gaussian elimination with respect to degree. I.e
$F^{i-1} \subseteq F^{i-1,(2)}, \ldots, F^1 \subseteq F^{1,(2)}$.

Normal forms

- Enable us to eliminate particular monomials containing x_1 from each F^i using the lower degree sets $F^{i-1}, \ldots, F^2, F^1$ as basis.
- The effect of normalization is that there is a rather large set of monomials containing x_1 that can not appear in each set F^i at the end.

The monomial orders

A. "Naive" XL elimination Monomials containing x_1 are largest

For each $i = \{1 \ldots, d\}$, Gaussian elimination on
$F^i \cup L^1 F^{i-1} \cup \ldots \cup L^{i-2} F^2 \cup L^{i-1} F^1$ to eliminate all monomials containing x_1.

B. Ordering the monomials with respect to degree

- For each $i = \{1 \ldots, d\}$, $\langle F^i \cup L^1 F^{i-1} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle$ may contain more polynomials of degree $< i$.
- We can try to produce a larger set of polynomials $F^{i-1,(2)}, \ldots, F^{1,(2)}$ by Gaussian elimination with respect to degree. I.e
$F^{i-1} \subseteq F^{i-1,(2)}, \ldots, F^1 \subseteq F^{1,(2)}$.

Normal forms

- Enable us to eliminate particular monomials containing x_1 from each F^i using the lower degree sets $F^{i-1}, \ldots, F^2, F^1$ as basis.
- The effect of normalization is that there is a rather large set of monomials containing x_1 that can not appear in each set F^i at the end.

Elimination tools

Resultants

- Given $f_i = a_i x_1 + b_i \in F^z$ and $f_j = a_j x_1 + b_j \in F^y$ satisfying $z + y \leq d + 1$, where $\deg a_i \leq z - 1$ and $\deg b_i \leq z$ (resp j)
- We can form the resultant with respect to x_1

$$\mathrm{Res}(f_i, f_j) = a_i b_j + a_j b_i = a_i f_j + a_j f_i \in B[2, n].$$

- The set of all resultants: $\mathrm{Res}_2^{y+z} = \{Res(f_i, f_j)\}$.

Coefficient constraints (GRFY 2017)

- Given $f = x_1 a + b \in F^i$ satisfying $2i \leq d + 1$, where $\deg a \leq i - 1$ and $\deg b \leq i$.
- We can form the coefficient constraint with respect to x_1

$$(a + 1)f = x_1 a(a + 1) + b(a + 1) = b(a + 1) \in B[2, n].$$

- The set of all coefficient constraints: $\mathrm{Co}_2^j = \{b_i(a_i + 1)\}$.

Elimination tools

Resultants

- Given $f_i = a_i x_1 + b_i \in F^z$ and $f_j = a_j x_1 + b_j \in F^y$ satisfying $z + y \leq d + 1$, where $\deg a_i \leq z - 1$ and $\deg b_i \leq z$ (resp j)
- We can form the resultant with respect to x_1

$$\mathrm{Res}(f_i, f_j) = a_i b_j + a_j b_i = a_i f_j + a_j f_i \in B[2, n].$$

- The set of all resultants: $\mathrm{Res}_2^{y+z} = \{Res(f_i, f_j)\}$.

Coefficient constraints (GRFY 2017)

- Given $f = x_1 a + b \in F^i$ satisfying $2i \leq d + 1$, where $\deg a \leq i - 1$ and $\deg b \leq i$.
- We can form the coefficient constraint with respect to x_1

$$(a + 1)f = x_1 a(a + 1) + b(a + 1) = b(a + 1) \in B[2, n].$$

- The set of all coefficient constraints: $\mathrm{Co}_2^j = \{b_i(a_i + 1)\}$.

Elimination tools

Resultants

- Given $f_i = a_i x_1 + b_i \in F^z$ and $f_j = a_j x_1 + b_j \in F^y$ satisfying $z + y \leq d + 1$, where $\deg a_i \leq z - 1$ and $\deg b_i \leq z$ (resp j)
- We can form the resultant with respect to x_1

$$\mathrm{Res}(f_i, f_j) = a_i b_j + a_j b_i = a_i f_j + a_j f_i \in B[2, n].$$

- The set of all resultants: $\mathrm{Res}_2^{y+z} = \{Res(f_i, f_j)\}$.

Coefficient constraints (GRFY 2017)

- Given $f = x_1 a + b \in F^i$ satisfying $2i \leq d + 1$, where $\deg a \leq i - 1$ and $\deg b \leq i$.
- We can form the coefficient constraint with respect to x_1

$$(a + 1)f = x_1 a(a + 1) + b(a + 1) = b(a + 1) \in B[2, n].$$

- The set of all coefficient constraints: $\mathrm{Co}_2^j = \{b_i(a_i + 1)\}$.

Elimination tools

Resultants

- Given $f_i = a_i x_1 + b_i \in F^z$ and $f_j = a_j x_1 + b_j \in F^y$ satisfying $z + y \leq d + 1$, where $\deg a_i \leq z - 1$ and $\deg b_i \leq z$ (resp j)
- We can form the resultant with respect to x_1

$$\mathrm{Res}(f_i, f_j) = a_i b_j + a_j b_i = a_i f_j + a_j f_i \in B[2, n].$$

- The set of all resultants: $\mathrm{Res}_2^{y+z} = \{Res(f_i, f_j)\}$.

Coefficient constraints (GRFY 2017)

- Given $f = x_1 a + b \in F^i$ satisfying $2i \leq d + 1$, where $\deg a \leq i - 1$ and $\deg b \leq i$.
- We can form the coefficient constraint with respect to x_1

$$(a + 1)f = x_1 a(a + 1) + b(a + 1) = b(a + 1) \in B[2, n].$$

- The set of all coefficient constraints: $\mathrm{Co}_2^j = \{b_i(a_i + 1)\}$.

Elimination tools

Resultants

- Given $f_i = a_i x_1 + b_i \in F^z$ and $f_j = a_j x_1 + b_j \in F^y$ satisfying $z + y \leq d + 1$, where $\deg a_i \leq z - 1$ and $\deg b_i \leq z$ (resp j)
- We can form the resultant with respect to x_1

$$\mathrm{Res}(f_i, f_j) = a_i b_j + a_j b_i = a_i f_j + a_j f_i \in B[2, n].$$

- The set of all resultants: $\mathrm{Res}_2^{y+z} = \{Res(f_i, f_j)\}$.

Coefficient constraints (GRFY 2017)

- Given $f = x_1 a + b \in F^i$ satisfying $2i \leq d + 1$, where $\deg a \leq i - 1$ and $\deg b \leq i$.
- We can form the coefficient constraint with respect to x_1

$$(a + 1)f = x_1 a(a + 1) + b(a + 1) = b(a + 1) \in B[2, n].$$

- The set of all coefficient constraints: $\mathrm{Co}_2^j = \{b_i(a_i + 1)\}$.

Elimination tools

Resultants

- Given $f_i = a_i x_1 + b_i \in F^z$ and $f_j = a_j x_1 + b_j \in F^y$ satisfying $z + y \leq d + 1$, where $\deg a_i \leq z - 1$ and $\deg b_i \leq z$ (resp j)
- We can form the resultant with respect to x_1

$$\mathrm{Res}(f_i, f_j) = a_i b_j + a_j b_i = a_i f_j + a_j f_i \in B[2, n].$$

- The set of all resultants: $\mathrm{Res}_2^{y+z} = \{Res(f_i, f_j)\}$.

Coefficient constraints (GRFY 2017)

- Given $f = x_1 a + b \in F^i$ satisfying $2i \leq d + 1$, where $\deg a \leq i - 1$ and $\deg b \leq i$.
- We can form the coefficient constraint with respect to x_1

$$(a + 1) f = x_1 a(a + 1) + b(a + 1) = b(a + 1) \in B[2, n].$$

- The set of all coefficient constraints: $\mathrm{Co}_2^j = \{b_i(a_i + 1)\}$.

Elimination tools

Resultants

- Given $f_i = a_i x_1 + b_i \in F^z$ and $f_j = a_j x_1 + b_j \in F^y$ satisfying $z + y \leq d + 1$, where $\deg a_i \leq z - 1$ and $\deg b_i \leq z$ (resp j)
- We can form the resultant with respect to x_1

$$\mathrm{Res}(f_i, f_j) = a_i b_j + a_j b_i = a_i f_j + a_j f_i \in B[2, n].$$

- The set of all resultants: $\mathrm{Res}_2^{y+z} = \{Res(f_i, f_j)\}$.

Coefficient constraints (GRFY 2017)

- Given $f = x_1 a + b \in F^i$ satisfying $2i \leq d + 1$, where $\deg a \leq i - 1$ and $\deg b \leq i$.
- We can form the coefficient constraint with respect to x_1

$$(a+1)f = x_1 a(a+1) + b(a+1) = b(a+1) \in B[2, n].$$

- The set of all coefficient constraints: $\mathrm{Co}_2^j = \{b_i(a_i + 1)\}$.

Extensions of GRFY elimination

Theorem 1

1. {Resultants + coefficient constraints + Normalization ++} = $\langle F^d \cup L^1F^{d-1} \cup L^2F^{d-2} \cup \cdots \cup L^{d-2}F^2 \cup L^{d-1}F^1 \rangle \cap B[2,n]$.
2. If we extend the above construction to include **B.**, we in general have {Resultants + coefficient constraints + Normalization ++} $\supset$ $\langle F^d \cup L^1F^{d-1} \cup L^2F^{d-2} \cup \cdots \cup L^{d-2}F^2 \cup L^{d-1}F^1 \rangle \cap B[2,n]$.

In general we expect that we can eliminate variables with lower (monomial) complexity with generalized GRFY framework $\rightarrow$ avoids multiplying with *all* variables.

In general we expect that generalized GRFY elimination introduces less false solutions than the XL method when restricting the degree $\leq d$.

Extensions of GRFY elimination

Theorem 1

1. {Resultants + coefficient constraints + Normalization ++} = $\langle F^d \cup L^1F^{d-1} \cup L^2F^{d-2} \cup \cdots \cup L^{d-2}F^2 \cup L^{d-1}F^1 \rangle \cap B[2,n]$.
2. If we extend the above construction to include **B.**, we in general have {Resultants + coefficient constraints + Normalization ++} $\supset$ $\langle F^d \cup L^1F^{d-1} \cup L^2F^{d-2} \cup \cdots \cup L^{d-2}F^2 \cup L^{d-1}F^1 \rangle \cap B[2,n]$.

In general we expect that we can eliminate variables with lower (monomial) complexity with generalized GRFY framework $\rightarrow$ avoids multiplying with *all* variables.

In general we expect that generalized GRFY elimination introduces less false solutions than the XL method when restricting the degree $\leq d$.

Extensions of GRFY elimination

Theorem 1

1. $\{$Resultants $+$ coefficient constraints $+$ Normalization $++\} =$ $\langle F^d \cup L^1F^{d-1} \cup L^2F^{d-2} \cup \cdots \cup L^{d-2}F^2 \cup L^{d-1}F^1 \rangle \cap B[2,n]$.
2. If we extend the above construction to include **B.**, we in general have $\{$Resultants $+$ coefficient constraints $+$ Normalization $++\} \supset$ $\langle F^d \cup L^1F^{d-1} \cup L^2F^{d-2} \cup \cdots \cup L^{d-2}F^2 \cup L^{d-1}F^1 \rangle \cap B[2,n]$.

In general we expect that we can eliminate variables with lower (monomial) complexity with generalized GRFY framework $\rightarrow$ avoids multiplying with *all* variables.

In general we expect that generalized GRFY elimination introduces less false solutions than the XL method when restricting the degree $\leq d$.

Extensions of GRFY elimination

Theorem 1

1. {Resultants + coefficient constraints + Normalization ++} = $\langle F^d \cup L^1F^{d-1} \cup L^2F^{d-2} \cup \cdots \cup L^{d-2}F^2 \cup L^{d-1}F^1\rangle \cap B[2,n]$.
2. If we extend the above construction to include **B.**, we in general have {Resultants + coefficient constraints + Normalization ++} $\supset \langle F^d \cup L^1F^{d-1} \cup L^2F^{d-2} \cup \cdots \cup L^{d-2}F^2 \cup L^{d-1}F^1\rangle \cap B[2,n]$.

In general we expect that we can eliminate variables with lower (monomial) complexity with generalized GRFY framework → avoids multiplying with *all* variables.

In general we expect that generalized GRFY elimination introduces less false solutions than the XL method when restricting the degree $\leq d$.

Extensions of GRFY elimination

Theorem 1

1. $\{$Resultants $+$ coefficient constraints $+$ Normalization $++\} =$
$\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle \cap B[2, n]$.
2. If we extend the above construction to include **B.**, we in general have
$\{$Resultants $+$ coefficient constraints $+$ Normalization $++\} \supset$
$\langle F^d \cup L^1 F^{d-1} \cup L^2 F^{d-2} \cup \cdots \cup L^{d-2} F^2 \cup L^{d-1} F^1 \rangle \cap B[2, n]$.

In general we expect that we can eliminate variables with lower (monomial) complexity with generalized GRFY framework $\rightarrow$ avoids multiplying with *all* variables.

In general we expect that generalized GRFY elimination introduces less false solutions than the XL method when restricting the degree $\leq d$.

Random system, 8 equations in variables $x_0, \ldots, x_7$, 1 unique solution

```
1+x1*x0+x1+x2+x3*x0+x3*x1+x3*x2+x3+x4*x0+x4*x2+x4+x5*x1+x5*x4+x7*x0+x7*x2+x7*x3+x7*x5

1+x0+x1+x2*x0+x2*x1+x3*x0+x3*x2+x3+x4*x0+x4*x2+x5*x0+x5*x2+x5*x3+x5*x4+x5+x6*x0+x6*x3+x6*x5+x7*x1+x7*x2+x7*x3+x7*x4+x7*x5+x7*x6

x0+x2*x1+x2+x3*x0+x3*x1+x3*x2+x3+x4*x0+x4*x2+x4*x3+x4+x6*x0+x6*x3+x6*x4+x7*x0+x7*x3+x7

1+x0+x1+x2+x4*x0+x4*x1+x4*x2+x4+x5*x0+x5*x1+x5*x3+x5+x6*x0+x6*x1+x6*x3+x6*x4+x6+x7*x0+x7*x1+x7*x2+x7*x3+x7*x4+x7*x5

1+x1+x2*x1+x2+x3*x1+x4*x2+x4+x5*x0+x5*x1+x5*x2+x5+x6*x1+x6*x2+x6*x3+x6*x5+x6+x7*x3+x7*x6

x2+x3*x2+x4*x1+x4*x3+x4+x5*x2+x5*x3+x6*x1+x6*x2+x6*x3+x6*x4+x6*x5+x6+x7*x0+x7*x1+x7

1+x2*x1+x2+x3*x2+x4*x1+x4*x2+x4*x3+x5*x4+x5+x6*x2+x6*x3+x7*x5+x7

1+x0+x3*x2+x3+x4*x2+x5*x1+x5*x4+x5+x6*x2+x6*x3+x6*x5+x7*x4+x7*x6+x7
```

Limiting degree to max 3, GRFY elimination of x_0, x_1

```
** Restricting degree to max. 3 **
After elimination of x0, got 7 polynomials:
x5x6x7 + x3x6x7 + x2x6x7 + x4x5x7 + x3x4x7 + x2x4x7 + x1x5x6 + x3x4x6 + x2x3x6 + x3x4x5 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x1x7 + x5x6 + x3x6 + x2x6 + x4x5 + x3x5 + x3x4 + x2x
3 + x1x2 + x7 + x6 + x4 + x2 + x1
x4x6x7 + x1x6x7 + x1x4x7 + x4x5x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x1x4x5 + x1x2x4 + x1x2x3 + x6x7 + x5x7 + x4x7 + x2x7 + x1x7 + x5x6 + x4x6 + x3x4 + x2x4 + x2x3 + x1x3 + x1
x2 + x6 + x5 + x3
x3x6x7 + x2x6x7 + x3x4x7 + x2x4x7 + x3x5x6 + x2x5x6 + x3x4x5 + x2x4x5 + x1x3x5 + x1x2x5 + x2x3x4 + x6x7 + x3x7 + x2x7 + x1x7 + x4x5 + x2x5 + x1x5 + x3x4 + x2x4 + x2x3 + x1x2 + x7
 + x3 + x2
x2x6x7 + x2x4x7 + x2x5x6 + x1x5x6 + x3x4x6 + x2x4x5 + x1x4x5 + x1x2x5 + x2x3x4 + x4x7 + x3x7 + x2x7 + x2x6 + x1x6 + x1x4 + x2x3 + x1x3 + x5 + x4 + x3 + x2 + 1
x1x5x7 + x3x4x7 + x2x3x7 + x1x5x6 + x3x4x6 + x2x3x6 + x3x4x5 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x6x7 + x5x7 + x4x7 + x3x7 + x4x6 + x1x6 + x4x5 + x2x5 + x1x4 + x1x2 + x1
x1x5x6 + x3x4x6 + x2x3x6 + x1x4x5 + x2x3x4 + x5x7 + x2x7 + x5x6 + x1x6 + x4x5 + x1x5 + x1x3 + x1x2 + x5 + x3 + x2 + 1
x5x7 + x3x6 + x2x6 + x4x5 + x3x4 + x2x4 + x1x4 + x2x3 + x1x2 + x7 + x5 + x2 + 1
After elimination of x1, got 1 polynomials:
x4x5x7 + x2x5x7 + x3x4x6 + x2x4x6 + x2x3x6 + x2x4x5 + x5x7 + x4x7 + x2x7 + x3x6 + x4x5 + x2x5 + x7 + x5 + x4 + x2 + 1
```

Limiting degree to max 4, General GRFY elimination of x_0, x_1

```
** Restricting degree to max. 4 **
After elimination of x0, got 14 polynomials:
x4x5x6x7 + x1x5x6x7 + x2x4x6x7 + x2x3x6x7 + x1x3x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x3x5x7 + x1x2x5x7 + x1x3x4x7 + x3x4x5x6 + x2x3x5x6 + x1x2x5x6 + x2x3x4x6 + x1x2x3x6 + x1
x3x4x5 + x1x2x3x4 + x5x6x7 + x3x6x7 + x4x5x7 + x2x5x7 + x1x5x7 + x3x4x7 + x2x3x7 + x1x3x7 + x4x5x6 + x2x5x6 + x1x5x6 + x2x4x6 + x2x3x6 + x1x3x6 + x3x4x5 + x2x3x5 + x1x3x5 + x1x2x
5 + x2x3x4 + x1x2x3 + x2x7 + x5x6 + x3x6 + x2x5 + x7 + x5 + x3 + x2 + 1
x3x5x6x7 + x1x5x6x7 + x2x4x6x7 + x2x3x6x7 + x1x2x6x7 + x3x4x5x7 + x1x3x4x7 + x1x2x4x7 + x1x2x3x7 + x2x4x5x6 + x1x4x5x6 + x1x3x4x6 + x2x3x4x5 + x1x3x4x5 + x1x2x3x5 + x3x5x7 + x3x4
x7 + x2x4x7 + x2x3x7 + x2x4x5 + x1x4x5 + x1x3x5 + x2x3x4 + x1x3x4 + x6x7 + x1x7 + x5x6 + x4x6 + x2x6 + x1x6 + x3x5 + x1x5 + x2x4 + x1x4 + x1x3 + x7 + x5 + x3 + x1 + 1
x2x5x6x7 + x1x5x6x7 + x1x4x6x7 + x2x4x5x7 + x1x4x5x7 + x1x2x5x7 + x2x3x4x7 + x2x3x5x6 + x1x3x5x6 + x1x3x4x6 + x1x2x4x6 + x2x3x4x5 + x1x2x3x4 + x5x6x7 + x2x6x7 + x4x5x7 + x2x5x7 +
 x1x5x7 + x2x4x7 + x2x3x7 + x3x5x6 + x2x4x6 + x2x3x6 + x1x3x5 + x1x2x5 + x2x3x4 + x1x3x4 + x1x2x4 + x6x7 + x5x7 + x4x7 + x2x7 + x4x6 + x3x6 + x2x6 + x3x5 + x2x5 + x1x5 + x1x4 + x
7 + x6 + x5 + x3 + x2 + 1
x1x5x6x7 + x2x3x6x7 + x1x3x6x7 + x1x3x5x7 + x1x2x3x7 + x2x3x5x6 + x1x2x5x6 + x1x3x4x6 + x1x3x4x5 + x1x2x3x4 + x2x6x7 + x2x3x7 + x1x3x7 + x1x2x7 + x2x5x6 + x1x5x6 + x3x4x6 + x1x3x
6 + x1x2x6 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x1x3x4 + x1x2x4 + x1x2x3 + x4x7 + x3x7 + x2x7 + x1x7 + x1x6 + x4x5 + x3x5 + x2x5 + x2x4 + x1x4 + x1x3 + x1x2 + x6 + x2 + x1
x3x4x6x7 + x1x4x6x7 + x1x2x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x2x5x7 + x1x3x4x7 + x1x2x4x7 + x1x2x3x7 + x3x4x5x6 + x1x4x5x6 + x1x2x4x6 + x2x3x4x5 + x1x3x4x5 + x1x2x4x5 + x1
x2x3x4 + x4x6x7 + x1x6x7 + x2x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x2x3x7 + x4x5x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x2x4x5 + x2x3x4 + x1x2x3 + x6x7 + x5x7 + x5x6 + x3x6 + x2x6 +
 x1x6 + x3x5 + x2x4 + x7 + x5 + x3 + 1
x2x4x6x7 + x1x3x6x7 + x1x2x6x7 + x1x3x4x7 + x1x2x4x7 + x3x4x5x6 + x2x4x5x6 + x1x4x5x6 + x2x3x5x6 + x1x2x5x6 + x1x3x4x6 + x1x2x3x6 + x2x3x4x5 + x5x6x7 + x4x6x7 + x3x6x7 + x2x6x7 +
 x1x6x7 + x4x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x1x3x7 + x1x2x7 + x4x5x6 + x2x5x6 + x1x5x6 + x3x4x6 + x2x4x6 + x1x2x6 + x2x4x5 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x2x3x4 + x1x3x4
 + x1x2x4 + x6x7 + x5x7 + x4x7 + x3x7 + x2x7 + x1x7 + x5x6 + x4x6 + x3x6 + x1x6 + x4x5 + x2x5 + x1x4 + x1x3 + x1x2 + x7 + x5 + x3 + x2 + x1 + 1
x1x4x6x7 + x2x3x6x7 + x1x3x6x7 + x1x2x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x3x5x7 + x1x3x4x7 + x1x2x4x7 + x1x4x5x6 + x2x3x5x6 + x1x2x5x6 + x1x2x4x6 + x1x2x3x6 + x2x3x4x5 + x1
x3x4x5 + x1x2x3x4 + x5x6x7 + x2x6x7 + x4x5x7 + x2x5x7 + x3x4x7 + x1x3x7 + x1x2x7 + x3x5x6 + x1x5x6 + x3x4x6 + x2x3x6 + x1x3x6 + x2x4x5 + x1x3x5 + x1x2x5 + x1x2x4 + x1x2x3 + x6x7
+ x3x7 + x1x6 + x4x5 + x3x5 + x3x4 + x2x4 + x2x3 + x1x2 + x4
x5x6x7 + x4x6x7 + x2x6x7 + x1x6x7 + x4x5x7 + x1x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x2x3x7 + x4x5x6 + x3x5x6 + x1x5x6 + x3x4x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x3x4x5 + x1x3x5
+ x1x2x4 + x1x2x3 + x6x7 + x4x7 + x3x7 + x2x7 + x1x7 + x3x6 + x4x5 + x3x5 + x1x5 + x3x4 + x3 + x2 + 1
x4x6x7 + x3x6x7 + x1x6x7 + x1x5x7 + x1x4x7 + x2x3x7 + x4x5x6 + x3x5x6 + x2x4x6 + x1x3x6 + x1x2x6 + x1x4x5 + x2x3x5 + x1x2x5 + x1x2x4 + x1x2x3 + x6x7 + x4x7 + x3x7 + x2x7 + x5x6 +
 x2x6 + x1x5 + x2x3 + x1x2 + x7 + x6 + x4 + x3 + x1 + 1
x3x6x7 + x1x5x7 + x2x3x7 + x3x5x6 + x2x3x6 + x2x3x5 + x1x2x5 + x5x7 + x3x7 + x1x7 + x4x6 + x2x6 + x1x5 + x3x4 + x2x4 + x1x3 + x7 + x5 + x4 + x1 + 1
x2x6x7 + x2x4x7 + x2x5x6 + x1x5x6 + x3x4x6 + x2x4x5 + x1x4x5 + x1x2x5 + x2x3x4 + x4x7 + x3x7 + x2x7 + x2x6 + x1x6 + x1x4 + x2x3 + x1x3 + x5 + x4 + x3 + x2 + 1
x1x5x7 + x3x4x7 + x2x3x7 + x1x5x6 + x3x4x6 + x2x3x6 + x3x4x5 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x6x7 + x5x7 + x4x7 + x3x7 + x4x6 + x1x6 + x4x5 + x2x5 + x1x4 + x1x2 + x1
x1x5x6 + x3x4x6 + x2x3x6 + x1x4x5 + x2x3x4 + x5x7 + x2x7 + x5x6 + x1x6 + x4x5 + x1x5 + x1x3 + x1x2 + x5 + x3 + x2 + 1
x5x7 + x3x6 + x2x6 + x4x5 + x3x4 + x2x4 + x1x4 + x2x3 + x1x2 + x7 + x5 + x2 + 1
After elimination of x1, got 4 polynomials:
x4x5x6x7 + x2x5x6x7 + x2x4x5x7 + x3x4x5x6 + x2x4x5x6 + x2x3x5x6 + x2x3x4x5 + x5x6x7 + x4x6x7 + x2x6x7 + x3x5x7 + x3x4x7 + x2x3x7 + x4x5x6 + x2x5x6 + x2x4x5 + x6x7 + x5x7 + x3x7 +
 x2x7 + x5x6 + x4x6 + x3x6 + x4x5 + x3x5 + x2x5 + x7 + x6 + x5 + x3 + x2 + 1
x3x5x6x7 + x3x4x5x7 + x3x4x5x6 + x5x6x7 + x4x6x7 + x2x5x7 + x3x4x7 + x3x5x6 + x2x4x6 + x3x4x5 + x2x4x5 + x2x3x4 + x5x7 + x2x7 + x4x5 + x3x5 + x2x5 + x3x4 + x2x3 + x7 + x6 + x5 +
x2 + 1
x2x5x6x7 + x2x4x6x7 + x2x3x6x7 + x3x4x5x7 + x2x4x5x7 + x2x4x5x6 + x2x3x5x6 + x2x3x4x6 + x2x3x4x5 + x2x6x7 + x3x5x7 + x3x4x7 + x2x4x6 + x3x4x5 + x2x4x5 + x2x3x5 + x2x3x4 + x4x7 +
x3x7 + x2x7 + x3x5 + x3x4 + x2x4 + x2x3 + x7 + x3 + x2 + 1
x4x5x7 + x2x5x7 + x3x4x6 + x2x4x6 + x2x3x6 + x2x4x5 + x5x7 + x4x7 + x2x7 + x3x6 + x4x5 + x2x5 + x7 + x5 + x4 + x2 + 1
```

Increasing degree to max 5, General GRFY elimination of x_0, x_1

- Eliminating x_0 gives same 14 polynomials as over.
- Eliminating x_1 gives 16 polynomials.

Limiting degree to max 4, General GRFY elimination of x_0, x_1

```
** Restricting degree to max. 4 **
After elimination of x0, got 14 polynomials:
x4x5x6x7 + x1x5x6x7 + x2x4x6x7 + x2x3x6x7 + x1x3x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x3x5x7 + x1x2x5x7 + x1x3x4x7 + x3x4x5x6 + x2x3x5x6 + x1x2x5x6 + x2x3x4x6 + x1x2x3x6 + x1
x3x4x5 + x1x2x3x4 + x5x6x7 + x3x6x7 + x4x5x7 + x2x5x7 + x1x5x7 + x3x4x7 + x2x3x7 + x1x3x7 + x4x5x6 + x2x5x6 + x1x5x6 + x2x4x6 + x2x3x6 + x1x3x6 + x3x4x5 + x2x3x5 + x1x3x5 + x1x2x
5 + x2x3x4 + x1x2x3 + x2x7 + x5x6 + x3x6 + x2x5 + x7 + x5 + x3 + x2 + 1
x3x5x6x7 + x1x5x6x7 + x2x4x6x7 + x2x3x6x7 + x1x2x6x7 + x3x4x5x7 + x1x3x4x7 + x1x2x4x7 + x1x2x3x7 + x2x4x5x6 + x1x4x5x6 + x1x3x4x6 + x2x3x4x5 + x1x3x4x5 + x1x2x3x5 + x3x5x7 + x3x4
x7 + x2x4x7 + x2x3x7 + x2x4x5 + x1x4x5 + x1x3x5 + x2x3x4 + x1x3x4 + x6x7 + x1x7 + x5x6 + x4x6 + x2x6 + x1x6 + x3x5 + x1x5 + x2x4 + x1x4 + x1x3 + x7 + x5 + x3 + x1 + 1
x2x5x6x7 + x1x5x6x7 + x1x4x6x7 + x2x4x5x7 + x1x4x5x7 + x1x2x5x7 + x2x3x4x7 + x2x3x5x6 + x1x3x5x6 + x1x3x4x6 + x1x2x4x6 + x2x3x4x5 + x1x2x3x4 + x5x6x7 + x2x6x7 + x4x5x7 + x2x5x7 +
 x1x5x7 + x2x4x7 + x2x3x7 + x3x5x6 + x2x4x6 + x2x3x6 + x1x3x5 + x1x2x5 + x2x3x4 + x1x3x4 + x1x2x4 + x6x7 + x5x7 + x4x7 + x2x7 + x4x6 + x3x6 + x2x6 + x3x5 + x2x5 + x1x5 + x1x4 + x
7 + x6 + x5 + x3 + x2 + 1
x1x5x6x7 + x2x3x6x7 + x1x3x6x7 + x1x3x5x7 + x1x2x3x7 + x2x3x5x6 + x1x2x5x6 + x1x3x4x6 + x1x3x4x5 + x1x2x3x4 + x2x6x7 + x2x3x7 + x1x3x7 + x1x2x7 + x2x5x6 + x1x5x6 + x3x4x6 + x1x3x
6 + x1x2x6 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x1x3x4 + x1x2x4 + x1x2x3 + x4x7 + x3x7 + x2x7 + x1x7 + x1x6 + x4x5 + x3x5 + x2x5 + x2x4 + x1x4 + x1x3 + x1x2 + x6 + x2 + x1
x3x4x6x7 + x1x4x6x7 + x1x2x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x2x5x7 + x1x3x4x7 + x1x2x4x7 + x1x2x3x7 + x3x4x5x6 + x1x4x5x6 + x1x2x4x6 + x2x3x4x5 + x1x3x4x5 + x1x2x4x5 + x1
x2x3x4 + x4x6x7 + x1x6x7 + x2x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x2x3x7 + x4x5x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x2x4x5 + x2x3x4 + x1x2x3 + x6x7 + x5x7 + x5x6 + x3x6 + x2x6 +
 x1x6 + x3x5 + x2x4 + x7 + x5 + x3 + 1
x2x4x6x7 + x1x3x6x7 + x1x2x6x7 + x1x3x4x7 + x1x2x4x7 + x3x4x5x6 + x2x4x5x6 + x1x4x5x6 + x2x3x5x6 + x1x2x5x6 + x1x3x4x6 + x1x2x3x6 + x2x3x4x5 + x5x6x7 + x4x6x7 + x3x6x7 + x2x6x7 +
 x1x6x7 + x4x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x1x3x7 + x1x2x7 + x4x5x6 + x2x5x6 + x1x5x6 + x3x4x6 + x2x4x6 + x1x2x6 + x2x4x5 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x2x3x4 + x1x3x4
 + x1x2x4 + x6x7 + x5x7 + x4x7 + x3x7 + x2x7 + x1x7 + x5x6 + x4x6 + x3x6 + x1x6 + x4x5 + x2x5 + x1x4 + x1x3 + x1x2 + x7 + x5 + x3 + x2 + x1 + 1
x1x4x6x7 + x2x3x6x7 + x1x3x6x7 + x1x2x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x3x5x7 + x1x3x4x7 + x1x2x4x7 + x1x4x5x6 + x2x3x5x6 + x1x2x5x6 + x1x2x4x6 + x1x2x3x6 + x2x3x4x5 + x1
x3x4x5 + x1x2x3x4 + x5x6x7 + x2x6x7 + x4x5x7 + x2x5x7 + x3x4x7 + x1x3x7 + x1x2x7 + x3x5x6 + x1x5x6 + x3x4x6 + x2x3x6 + x1x3x6 + x2x4x5 + x1x3x5 + x1x2x5 + x1x2x4 + x1x2x3 + x6x7
+ x3x7 + x1x6 + x4x5 + x3x5 + x3x4 + x2x4 + x2x3 + x1x2 + x4
x5x6x7 + x4x6x7 + x2x6x7 + x1x6x7 + x4x5x7 + x1x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x2x3x7 + x4x5x6 + x3x5x6 + x1x5x6 + x3x4x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x3x4x5 + x1x3x5
+ x1x2x4 + x1x2x3 + x6x7 + x4x7 + x3x7 + x2x7 + x1x7 + x3x6 + x4x5 + x3x5 + x1x5 + x3x4 + x3 + x2 + 1
x4x6x7 + x3x6x7 + x1x6x7 + x1x5x7 + x1x4x7 + x2x3x7 + x4x5x6 + x3x5x6 + x2x4x6 + x1x3x6 + x1x2x6 + x1x4x5 + x2x3x5 + x1x2x5 + x1x2x4 + x1x2x3 + x6x7 + x4x7 + x3x7 + x2x7 + x5x6 +
 x2x6 + x1x5 + x2x3 + x1x2 + x7 + x6 + x4 + x3 + x1 + 1
x3x6x7 + x1x5x7 + x2x3x7 + x3x5x6 + x2x3x6 + x2x3x5 + x1x2x5 + x5x7 + x3x7 + x1x7 + x4x6 + x2x6 + x1x5 + x3x4 + x2x4 + x1x3 + x7 + x5 + x4 + x1 + 1
x2x6x7 + x2x4x7 + x2x5x6 + x1x5x6 + x3x4x6 + x2x4x5 + x1x4x5 + x1x2x5 + x2x3x4 + x4x7 + x3x7 + x2x7 + x2x6 + x1x6 + x1x4 + x2x3 + x1x3 + x5 + x4 + x3 + x2 + 1
x1x5x7 + x3x4x7 + x2x3x7 + x1x5x6 + x3x4x6 + x2x3x6 + x3x4x5 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x6x7 + x5x7 + x4x7 + x3x7 + x4x6 + x1x6 + x4x5 + x2x5 + x1x4 + x1x2 + x1
x1x5x6 + x3x4x6 + x2x3x6 + x1x4x5 + x2x3x4 + x5x7 + x2x7 + x5x6 + x1x6 + x4x5 + x1x5 + x1x3 + x1x2 + x5 + x3 + x2 + 1
x5x7 + x3x6 + x2x6 + x4x5 + x3x4 + x2x4 + x1x4 + x2x3 + x1x2 + x7 + x5 + x2 + 1
After elimination of x1, got 4 polynomials:
x4x5x6x7 + x2x5x6x7 + x2x4x5x7 + x3x4x5x6 + x2x4x5x6 + x2x3x5x6 + x2x3x4x5 + x5x6x7 + x4x6x7 + x2x6x7 + x3x5x7 + x3x4x7 + x2x3x7 + x4x5x6 + x2x5x6 + x2x4x5 + x6x7 + x5x7 + x3x7 +
 x2x7 + x5x6 + x4x6 + x3x6 + x4x5 + x3x5 + x2x5 + x7 + x6 + x5 + x3 + x2 + 1
x3x5x6x7 + x3x4x5x7 + x3x4x5x6 + x5x6x7 + x4x6x7 + x2x5x7 + x3x4x7 + x3x5x6 + x2x4x6 + x3x4x5 + x2x4x5 + x2x3x4 + x5x7 + x2x7 + x4x5 + x3x5 + x2x5 + x3x4 + x2x3 + x7 + x6 + x5 +
x2 + 1
x2x5x6x7 + x2x4x6x7 + x2x3x6x7 + x3x4x5x7 + x2x4x5x7 + x2x4x5x6 + x2x3x5x6 + x2x3x4x6 + x2x3x4x5 + x2x6x7 + x3x5x7 + x3x4x7 + x2x4x6 + x3x4x5 + x2x4x5 + x2x3x5 + x2x3x4 + x4x7 +
x3x7 + x2x7 + x3x5 + x3x4 + x2x4 + x2x3 + x7 + x3 + x2 + 1
x4x5x7 + x2x5x7 + x3x4x6 + x2x4x6 + x2x3x6 + x2x4x5 + x5x7 + x4x7 + x2x7 + x3x6 + x4x5 + x2x5 + x7 + x5 + x4 + x2 + 1
```

Increasing degree to max 5, General GRFY elimination of x_0, x_1

- Eliminating x_0 gives same 14 polynomials as over.
- Eliminating x_1 gives 16 polynomials.

Limiting degree to max 4, General GRFY elimination of x_0, x_1

```
** Restricting degree to max. 4 **
After elimination of x0, got 14 polynomials:
x4x5x6x7 + x1x5x6x7 + x2x4x6x7 + x2x3x6x7 + x1x3x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x3x5x7 + x1x2x5x7 + x1x3x4x7 + x3x4x5x6 + x2x3x5x6 + x1x2x5x6 + x2x3x4x6 + x1x2x3x6 + x1
x3x4x5 + x1x2x3x4 + x5x6x7 + x3x6x7 + x4x5x7 + x2x5x7 + x1x5x7 + x3x4x7 + x2x3x7 + x1x3x7 + x4x5x6 + x2x5x6 + x1x5x6 + x2x4x6 + x2x3x6 + x1x3x6 + x3x4x5 + x2x3x5 + x1x3x5 + x1x2x
5 + x2x3x4 + x1x2x3 + x2x7 + x5x6 + x3x6 + x2x5 + x7 + x5 + x3 + x2 + 1
x3x5x6x7 + x1x5x6x7 + x2x4x6x7 + x2x3x6x7 + x1x2x6x7 + x3x4x5x7 + x1x3x4x7 + x1x2x4x7 + x1x2x3x7 + x2x4x5x6 + x1x4x5x6 + x1x3x4x6 + x2x3x4x5 + x1x3x4x5 + x1x2x3x5 + x3x5x7 + x3x4
x7 + x2x4x7 + x2x3x7 + x2x4x5 + x1x4x5 + x1x3x5 + x2x3x4 + x1x3x4 + x6x7 + x1x7 + x5x6 + x4x6 + x2x6 + x1x6 + x3x5 + x1x5 + x2x4 + x1x4 + x1x3 + x7 + x5 + x3 + x1 + 1
x2x5x6x7 + x1x5x6x7 + x1x4x6x7 + x2x4x5x7 + x1x4x5x7 + x1x2x5x7 + x2x3x4x7 + x2x3x5x6 + x1x3x5x6 + x1x3x4x6 + x1x2x4x6 + x2x3x4x5 + x1x2x3x4 + x5x6x7 + x2x6x7 + x4x5x7 + x2x5x7 +
 x1x5x7 + x2x4x7 + x2x3x7 + x3x5x6 + x2x4x6 + x2x3x6 + x1x3x5 + x1x2x5 + x2x3x4 + x1x3x4 + x1x2x4 + x6x7 + x5x7 + x4x7 + x2x7 + x4x6 + x3x6 + x2x6 + x3x5 + x2x5 + x1x5 + x1x4 + x
7 + x6 + x5 + x3 + x2 + 1
x1x5x6x7 + x2x3x6x7 + x1x3x6x7 + x1x3x5x7 + x1x2x3x7 + x2x3x5x6 + x1x2x5x6 + x1x3x4x6 + x1x3x4x5 + x1x2x3x4 + x2x6x7 + x2x3x7 + x1x3x7 + x1x2x7 + x2x5x6 + x1x5x6 + x3x4x6 + x1x3x
6 + x1x2x6 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x1x3x4 + x1x2x4 + x1x2x3 + x4x7 + x3x7 + x2x7 + x1x7 + x1x6 + x4x5 + x3x5 + x2x5 + x2x4 + x1x4 + x1x3 + x1x2 + x6 + x2 + x1
x3x4x6x7 + x1x4x6x7 + x1x2x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x2x5x7 + x1x3x4x7 + x1x2x4x7 + x1x2x3x7 + x3x4x5x6 + x1x4x5x6 + x1x2x4x6 + x2x3x4x5 + x1x3x4x5 + x1x2x4x5 + x1
x2x3x4 + x4x6x7 + x1x6x7 + x2x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x2x3x7 + x4x5x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x2x4x5 + x2x3x4 + x1x2x3 + x6x7 + x5x7 + x5x6 + x3x6 + x2x6 +
 x1x6 + x3x5 + x2x4 + x7 + x5 + x3 + 1
x2x4x6x7 + x1x3x6x7 + x1x2x6x7 + x1x3x4x7 + x1x2x4x7 + x3x4x5x6 + x2x4x5x6 + x1x4x5x6 + x2x3x5x6 + x1x2x5x6 + x1x3x4x6 + x1x2x3x6 + x2x3x4x5 + x5x6x7 + x4x6x7 + x3x6x7 + x2x6x7 +
 x1x6x7 + x4x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x1x3x7 + x1x2x7 + x4x5x6 + x2x5x6 + x1x5x6 + x3x4x6 + x2x4x6 + x1x2x6 + x2x4x5 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x2x3x4 + x1x3x4
 + x1x2x4 + x6x7 + x5x7 + x4x7 + x3x7 + x2x7 + x1x7 + x5x6 + x4x6 + x3x6 + x1x6 + x4x5 + x2x5 + x1x4 + x1x3 + x1x2 + x7 + x5 + x3 + x2 + x1 + 1
x1x4x6x7 + x2x3x6x7 + x1x3x6x7 + x1x2x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x3x5x7 + x1x3x4x7 + x1x2x4x7 + x1x4x5x6 + x2x3x5x6 + x1x2x5x6 + x1x2x4x6 + x1x2x3x6 + x2x3x4x5 + x1
x3x4x5 + x1x2x3x4 + x5x6x7 + x2x6x7 + x4x5x7 + x2x5x7 + x3x4x7 + x1x3x7 + x1x2x7 + x3x5x6 + x1x5x6 + x3x4x6 + x2x3x6 + x1x3x6 + x2x4x5 + x1x3x5 + x1x2x5 + x1x2x4 + x1x2x3 + x6x7
+ x3x7 + x1x6 + x4x5 + x3x5 + x3x4 + x2x4 + x2x3 + x1x2 + x4
x5x6x7 + x4x6x7 + x2x6x7 + x1x6x7 + x4x5x7 + x1x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x2x3x7 + x4x5x6 + x3x5x6 + x1x5x6 + x3x4x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x3x4x5 + x1x3x5
+ x1x2x4 + x1x2x3 + x6x7 + x4x7 + x3x7 + x2x7 + x1x7 + x3x6 + x4x5 + x3x5 + x1x5 + x3x4 + x3 + x2 + 1
x4x6x7 + x3x6x7 + x1x6x7 + x1x5x7 + x1x4x7 + x2x3x7 + x4x5x6 + x3x5x6 + x2x4x6 + x1x3x6 + x1x2x6 + x1x4x5 + x2x3x5 + x1x2x5 + x1x2x4 + x1x2x3 + x6x7 + x4x7 + x3x7 + x2x7 + x5x6 +
 x2x6 + x1x5 + x2x3 + x1x2 + x7 + x6 + x4 + x3 + x1 + 1
x3x6x7 + x1x5x7 + x2x3x7 + x3x5x6 + x2x3x6 + x2x3x5 + x1x2x5 + x5x7 + x3x7 + x1x7 + x4x6 + x2x6 + x1x5 + x3x4 + x2x4 + x1x3 + x7 + x5 + x4 + x1 + 1
x2x6x7 + x2x4x7 + x2x5x6 + x1x5x6 + x3x4x6 + x2x4x5 + x1x4x5 + x1x2x5 + x2x3x4 + x4x7 + x3x7 + x2x7 + x2x6 + x1x6 + x1x4 + x2x3 + x1x3 + x5 + x4 + x3 + x2 + 1
x1x5x7 + x3x4x7 + x2x3x7 + x1x5x6 + x3x4x6 + x2x3x6 + x3x4x5 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x6x7 + x5x7 + x4x7 + x3x7 + x4x6 + x1x6 + x4x5 + x2x5 + x1x4 + x1x2 + x1
x1x5x6 + x3x4x6 + x2x3x6 + x1x4x5 + x2x3x4 + x5x7 + x2x7 + x5x6 + x1x6 + x4x5 + x1x5 + x1x3 + x1x2 + x5 + x3 + x2 + 1
x5x7 + x3x6 + x2x6 + x4x5 + x3x4 + x2x4 + x1x4 + x2x3 + x1x2 + x7 + x5 + x2 + 1
After elimination of x1, got 4 polynomials:
x4x5x6x7 + x2x5x6x7 + x2x4x5x7 + x3x4x5x6 + x2x4x5x6 + x2x3x5x6 + x2x3x4x5 + x5x6x7 + x4x6x7 + x2x6x7 + x3x5x7 + x3x4x7 + x2x3x7 + x4x5x6 + x2x5x6 + x2x4x5 + x6x7 + x5x7 + x3x7 +
 x2x7 + x5x6 + x4x6 + x3x6 + x4x5 + x3x5 + x2x5 + x7 + x6 + x5 + x3 + x2 + 1
x3x5x6x7 + x3x4x5x7 + x3x4x5x6 + x5x6x7 + x4x6x7 + x2x5x7 + x3x4x7 + x3x5x6 + x2x4x6 + x3x4x5 + x2x4x5 + x2x3x4 + x5x7 + x2x7 + x4x5 + x3x5 + x2x5 + x3x4 + x2x3 + x7 + x6 + x5 +
x2 + 1
x2x5x6x7 + x2x4x6x7 + x2x3x6x7 + x3x4x5x7 + x2x4x5x7 + x2x4x5x6 + x2x3x5x6 + x2x3x4x6 + x2x3x4x5 + x2x6x7 + x3x5x7 + x3x4x7 + x2x4x6 + x3x4x5 + x2x4x5 + x2x3x5 + x2x3x4 + x4x7 +
x3x7 + x2x7 + x3x5 + x3x4 + x2x4 + x2x3 + x7 + x3 + x2 + 1
x4x5x7 + x2x5x7 + x3x4x6 + x2x4x6 + x2x3x6 + x2x4x5 + x5x7 + x4x7 + x2x7 + x3x6 + x4x5 + x2x5 + x7 + x5 + x4 + x2 + 1
```

Increasing degree to max 5, General GRFY elimination of x_0, x_1

- Eliminating x_0 gives same 14 polynomials as over.
- Eliminating x_1 gives 16 polynomials.

Limiting degree to max 4, General GRFY elimination of x_0, x_1

```
** Restricting degree to max. 4 **
After elimination of x0, got 14 polynomials:
x4x5x6x7 + x1x5x6x7 + x2x4x6x7 + x2x3x6x7 + x1x3x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x3x5x7 + x1x2x5x7 + x1x3x4x7 + x3x4x5x6 + x2x3x5x6 + x1x2x5x6 + x2x3x4x6 + x1x2x3x6 + x1
x3x4x5 + x1x2x3x4 + x5x6x7 + x3x6x7 + x4x5x7 + x2x5x7 + x1x5x7 + x3x4x7 + x2x3x7 + x1x3x7 + x4x5x6 + x2x5x6 + x1x5x6 + x2x4x6 + x2x3x6 + x1x3x6 + x3x4x5 + x2x3x5 + x1x3x5 + x1x2x
5 + x2x3x4 + x1x2x3 + x2x7 + x5x6 + x3x6 + x2x5 + x7 + x5 + x3 + x2 + 1
x3x5x6x7 + x1x5x6x7 + x2x4x6x7 + x2x3x6x7 + x1x2x6x7 + x3x4x5x7 + x1x3x4x7 + x1x2x4x7 + x1x2x3x7 + x2x4x5x6 + x1x4x5x6 + x1x3x4x6 + x2x3x4x5 + x1x3x4x5 + x1x2x3x5 + x3x5x7 + x3x4
x7 + x2x4x7 + x2x3x7 + x2x4x5 + x1x4x5 + x1x3x5 + x2x3x4 + x1x3x4 + x6x7 + x1x7 + x5x6 + x4x6 + x2x6 + x1x6 + x3x5 + x1x5 + x2x4 + x1x4 + x1x3 + x7 + x5 + x3 + x1 + 1
x2x5x6x7 + x1x5x6x7 + x1x4x6x7 + x2x4x5x7 + x1x4x5x7 + x1x2x5x7 + x2x3x4x7 + x2x3x5x6 + x1x3x5x6 + x1x3x4x6 + x1x2x4x6 + x2x3x4x5 + x1x2x3x4 + x5x6x7 + x2x6x7 + x4x5x7 + x2x5x7 +
 x1x5x7 + x2x4x7 + x2x3x7 + x3x5x6 + x2x4x6 + x2x3x6 + x1x3x5 + x1x2x5 + x2x3x4 + x1x3x4 + x1x2x4 + x6x7 + x5x7 + x4x7 + x2x7 + x4x6 + x3x6 + x2x6 + x3x5 + x2x5 + x1x5 + x1x4 + x
7 + x6 + x5 + x3 + x2 + 1
x1x5x6x7 + x2x3x6x7 + x1x3x6x7 + x1x3x5x7 + x1x2x3x7 + x2x3x5x6 + x1x2x5x6 + x1x3x4x6 + x1x3x4x5 + x1x2x3x4 + x2x6x7 + x2x3x7 + x1x3x7 + x1x2x7 + x2x5x6 + x1x5x6 + x3x4x6 + x1x3x
6 + x1x2x6 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x1x3x4 + x1x2x4 + x1x2x3 + x4x7 + x3x7 + x2x7 + x1x7 + x1x6 + x4x5 + x3x5 + x2x5 + x2x4 + x1x4 + x1x3 + x1x2 + x6 + x2 + x1
x3x4x6x7 + x1x4x6x7 + x1x2x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x2x5x7 + x1x3x4x7 + x1x2x4x7 + x1x2x3x7 + x3x4x5x6 + x1x4x5x6 + x1x2x4x6 + x2x3x4x5 + x1x3x4x5 + x1x2x4x5 + x1
x2x3x4 + x4x6x7 + x1x6x7 + x2x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x2x3x7 + x4x5x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x2x4x5 + x2x3x4 + x1x2x3 + x6x7 + x5x7 + x5x6 + x3x6 + x2x6 +
 x1x6 + x3x5 + x2x4 + x7 + x5 + x3 + 1
x2x4x6x7 + x1x3x6x7 + x1x2x6x7 + x1x3x4x7 + x1x2x4x7 + x3x4x5x6 + x2x4x5x6 + x1x4x5x6 + x2x3x5x6 + x1x2x5x6 + x1x3x4x6 + x1x2x3x6 + x2x3x4x5 + x5x6x7 + x4x6x7 + x3x6x7 + x2x6x7 +
 x1x6x7 + x4x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x1x3x7 + x1x2x7 + x4x5x6 + x2x5x6 + x1x5x6 + x3x4x6 + x2x4x6 + x1x2x6 + x2x4x5 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x2x3x4 + x1x3x4
 + x1x2x4 + x6x7 + x5x7 + x4x7 + x3x7 + x2x7 + x1x7 + x5x6 + x4x6 + x3x6 + x1x6 + x4x5 + x2x5 + x1x4 + x1x3 + x1x2 + x7 + x5 + x3 + x2 + x1 + 1
x1x4x6x7 + x2x3x6x7 + x1x3x6x7 + x1x2x6x7 + x3x4x5x7 + x1x4x5x7 + x2x3x5x7 + x1x3x5x7 + x1x3x4x7 + x1x2x4x7 + x1x4x5x6 + x2x3x5x6 + x1x2x5x6 + x1x2x4x6 + x1x2x3x6 + x2x3x4x5 + x1
x3x4x5 + x1x2x3x4 + x5x6x7 + x2x6x7 + x4x5x7 + x2x5x7 + x3x4x7 + x1x3x7 + x1x2x7 + x3x5x6 + x1x5x6 + x3x4x6 + x2x3x6 + x1x3x6 + x2x4x5 + x1x3x5 + x1x2x5 + x1x2x4 + x1x2x3 + x6x7
+ x3x7 + x1x6 + x4x5 + x3x5 + x3x4 + x2x4 + x2x3 + x1x2 + x4
x5x6x7 + x4x6x7 + x2x6x7 + x1x6x7 + x4x5x7 + x1x5x7 + x3x4x7 + x2x4x7 + x1x4x7 + x2x3x7 + x4x5x6 + x3x5x6 + x1x5x6 + x3x4x6 + x2x4x6 + x2x3x6 + x1x3x6 + x1x2x6 + x3x4x5 + x1x3x5
+ x1x2x4 + x1x2x3 + x6x7 + x4x7 + x3x7 + x2x7 + x1x7 + x3x6 + x4x5 + x3x5 + x1x5 + x3x4 + x3 + x2 + 1
x4x6x7 + x3x6x7 + x1x6x7 + x1x5x7 + x1x4x7 + x2x3x7 + x4x5x6 + x3x5x6 + x2x4x6 + x1x3x6 + x1x2x6 + x1x4x5 + x2x3x5 + x1x2x5 + x1x2x4 + x1x2x3 + x6x7 + x4x7 + x3x7 + x2x7 + x5x6 +
 x2x6 + x1x5 + x2x3 + x1x2 + x7 + x6 + x4 + x3 + x1 + 1
x3x6x7 + x1x5x7 + x2x3x7 + x3x5x6 + x2x3x6 + x2x3x5 + x1x2x5 + x5x7 + x3x7 + x1x7 + x4x6 + x2x6 + x1x5 + x3x4 + x2x4 + x1x3 + x7 + x5 + x4 + x1 + 1
x2x6x7 + x2x4x7 + x2x5x6 + x1x5x6 + x3x4x6 + x2x4x5 + x1x4x5 + x1x2x5 + x2x3x4 + x4x7 + x3x7 + x2x7 + x2x6 + x1x6 + x1x4 + x2x3 + x1x3 + x5 + x4 + x3 + x2 + 1
x1x5x7 + x3x4x7 + x2x3x7 + x1x5x6 + x3x4x6 + x2x3x6 + x3x4x5 + x1x4x5 + x2x3x5 + x1x3x5 + x1x2x5 + x6x7 + x5x7 + x4x7 + x3x7 + x4x6 + x1x6 + x4x5 + x2x5 + x1x4 + x1x2 + x1
x1x5x6 + x3x4x6 + x2x3x6 + x1x4x5 + x2x3x4 + x5x7 + x2x7 + x5x6 + x1x6 + x4x5 + x1x5 + x1x3 + x1x2 + x5 + x3 + x2 + 1
x5x7 + x3x6 + x2x6 + x4x5 + x3x4 + x2x4 + x1x4 + x2x3 + x1x2 + x7 + x5 + x2 + 1
After elimination of x1, got 4 polynomials:
x4x5x6x7 + x2x5x6x7 + x2x4x5x7 + x3x4x5x6 + x2x4x5x6 + x2x3x5x6 + x2x3x4x5 + x5x6x7 + x4x6x7 + x2x6x7 + x3x5x7 + x3x4x7 + x2x3x7 + x4x5x6 + x2x5x6 + x2x4x5 + x6x7 + x5x7 + x3x7 +
 x2x7 + x5x6 + x4x6 + x3x6 + x4x5 + x3x5 + x2x5 + x7 + x6 + x5 + x3 + x2 + 1
x3x5x6x7 + x3x4x5x7 + x3x4x5x6 + x5x6x7 + x4x6x7 + x2x5x7 + x3x4x7 + x3x5x6 + x2x4x6 + x3x4x5 + x2x4x5 + x2x3x4 + x5x7 + x2x7 + x4x5 + x3x5 + x2x5 + x3x4 + x2x3 + x7 + x6 + x5 +
x2 + 1
x2x5x6x7 + x2x4x6x7 + x2x3x6x7 + x3x4x5x7 + x2x4x5x7 + x2x4x5x6 + x2x3x5x6 + x2x3x4x6 + x2x3x4x5 + x2x6x7 + x3x5x7 + x3x4x7 + x2x4x6 + x3x4x5 + x2x4x5 + x2x3x5 + x2x3x4 + x4x7 +
x3x7 + x2x7 + x3x5 + x3x4 + x2x4 + x2x3 + x7 + x3 + x2 + 1
x4x5x7 + x2x5x7 + x3x4x6 + x2x4x6 + x2x3x6 + x2x4x5 + x5x7 + x4x7 + x2x7 + x3x6 + x4x5 + x2x5 + x7 + x5 + x4 + x2 + 1
```

Increasing degree to max 5, General GRFY elimination of x_0, x_1

- Eliminating x_0 gives same 14 polynomials as over.
- Eliminating x_1 gives 16 polynomials.

Re-linearization analysis of quadratic random systems with 1 unique solution, m equations in n variables

First elimination ideal, degree ≤ 3

- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3}(\binom{n}{i})$
- $\binom{m}{2} + m < \sum_{i=1}^{3}(\binom{n}{i}) \rightarrow$ no re-linearization.

Second elimination ideal, degree ≤ 5

- Number of resultants $\binom{\binom{m}{2}+m}{2}$, of coefficient constraints $\binom{m}{2} + m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$.
- Number of monomials of degree 5: $\sum_{i=1}^{5}(\binom{n}{i})$
- $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m > \sum_{i=1}^{5}(\binom{n}{i})$?

Re-linearization analysis of quadratic random systems with 1 unique solution, m equations in n variables

First elimination ideal, degree ≤ 3

- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3}(\binom{n}{i})$
- $\binom{m}{2} + m < \sum_{i=1}^{3}(\binom{n}{i}) \to$ no re-linearization.

Second elimination ideal, degree ≤ 5

- Number of resultants $\binom{\binom{m}{2}+m}{2}$, of coefficient constraints $\binom{m}{2} + m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$.
- Number of monomials of degree 5: $\sum_{i=1}^{5}(\binom{n}{i})$
- $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m > \sum_{i=1}^{5}(\binom{n}{i})$?

Re-linearization analysis of quadratic random systems with 1 unique solution, m equations in n variables

First elimination ideal, degree ≤ 3

- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3}(\binom{n}{i})$
- $\binom{m}{2} + m < \sum_{i=1}^{3}(\binom{n}{i}) \rightarrow$ no re-linearization.

Second elimination ideal, degree ≤ 5

- Number of resultants $\binom{\binom{m}{2}+m}{2}$, of coefficient constraints $\binom{m}{2} + m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$.
- Number of monomials of degree 5: $\sum_{i=1}^{5}(\binom{n}{i})$
- $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m > \sum_{i=1}^{5}(\binom{n}{i})$?

Re-linearization analysis of quadratic random systems with 1 unique solution, m equations in n variables

First elimination ideal, degree ≤ 3

- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3}(\binom{n}{i})$
- $\binom{m}{2} + m < \sum_{i=1}^{3}(\binom{n}{i}) \rightarrow$ no re-linearization.

Second elimination ideal, degree ≤ 5

- Number of resultants $\binom{\binom{m}{2}+m}{2}$, of coefficient constraints $\binom{m}{2} + m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$.
- Number of monomials of degree 5: $\sum_{i=1}^{5}(\binom{n}{i})$
- $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m > \sum_{i=1}^{5}(\binom{n}{i})$?

Re-linearization analysis of quadratic random systems with 1 unique solution, m equations in n variables

First elimination ideal, degree ≤ 3

- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3}(\binom{n}{i})$
- $\binom{m}{2} + m < \sum_{i=1}^{3}(\binom{n}{i}) \rightarrow$ no re-linearization.

Second elimination ideal, degree ≤ 5

- Number of resultants $\binom{\binom{m}{2}+m}{2}$, of coefficient constraints $\binom{m}{2} + m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$.
- Number of monomials of degree 5: $\sum_{i=1}^{5}(\binom{n}{i})$
- $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m > \sum_{i=1}^{5}(\binom{n}{i})$?

Re-linearization analysis of quadratic random systems with 1 unique solution, m equations in n variables

First elimination ideal, degree ≤ 3

- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3}(\binom{n}{i})$
- $\binom{m}{2} + m < \sum_{i=1}^{3}(\binom{n}{i}) \rightarrow$ no re-linearization.

Second elimination ideal, degree ≤ 5

- Number of resultants $\binom{\binom{m}{2}+m}{2}$, of coefficient constraints $\binom{m}{2} + m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$.
- Number of monomials of degree 5: $\sum_{i=1}^{5}(\binom{n}{i})$
- $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m > \sum_{i=1}^{5}(\binom{n}{i})$?

Re-linearization analysis of quadratic random systems with 1 unique solution, m equations in n variables

First elimination ideal, degree ≤ 3

- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3}(\binom{n}{i})$
- $\binom{m}{2} + m < \sum_{i=1}^{3}(\binom{n}{i}) \rightarrow$ no re-linearization.

Second elimination ideal, degree ≤ 5

- Number of resultants $\binom{\binom{m}{2}+m}{2}$, of coefficient constraints $\binom{m}{2} + m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$.
- Number of monomials of degree 5: $\sum_{i=1}^{5}(\binom{n}{i})$
- $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m > \sum_{i=1}^{5}(\binom{n}{i})$?

Re-linearization analysis of quadratic random systems with 1 unique solution, m equations in n variables

First elimination ideal, degree ≤ 3

- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3}(\binom{n}{i})$
- $\binom{m}{2} + m < \sum_{i=1}^{3}(\binom{n}{i}) \rightarrow$ no re-linearization.

Second elimination ideal, degree ≤ 5

- Number of resultants $\binom{\binom{m}{2}+m}{2}$, of coefficient constraints $\binom{m}{2} + m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$.
- Number of monomials of degree 5: $\sum_{i=1}^{5}(\binom{n}{i})$
- $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m > \sum_{i=1}^{5}(\binom{n}{i})$?

Re-linearization analysis of quadratic random systems with 1 unique solution, m equations in n variables

First elimination ideal, degree ≤ 3

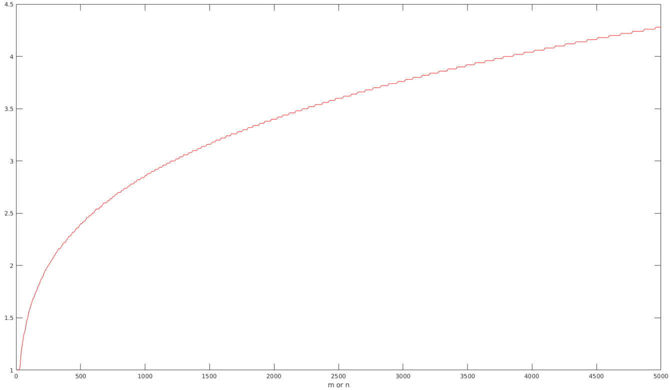
- Number of resultants $\binom{m}{2}$, of coefficient constraints m
- Number of polynomials produced of elimination: $\binom{m}{2} + m$
- Number of monomials of degree 3: $\sum_{i=1}^{3}(\binom{n}{i})$
- $\binom{m}{2} + m < \sum_{i=1}^{3}(\binom{n}{i}) \rightarrow$ no re-linearization.

Second elimination ideal, degree ≤ 5

- Number of resultants $\binom{\binom{m}{2}+m}{2}$, of coefficient constraints $\binom{m}{2} + m$.
- Number of polynomials produced of elimination: $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m$.
- Number of monomials of degree 5: $\sum_{i=1}^{5}(\binom{n}{i})$
- $\binom{\binom{m}{2}+m}{2} + \binom{m}{2} + m > \sum_{i=1}^{5}(\binom{n}{i})$?

Re-linearization curve

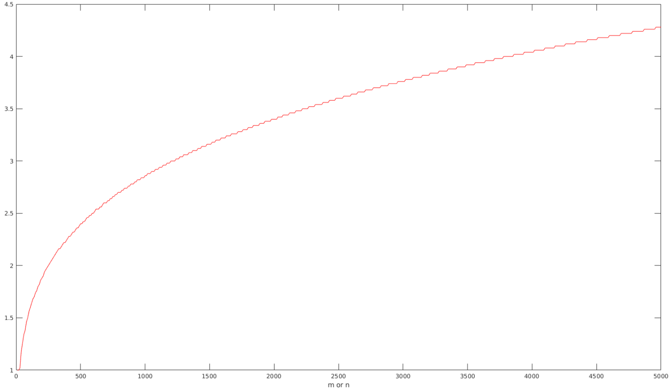
3-5 version2.png



When $m = n$ this holds true for $1 \leq n \leq\approx 25$

Re-linearization curve

3-5 version2.png



When $m = n$ this holds true for $1 \leq n \leq\approx 25$