Some elimination problems for matrices

Wilhelm Plesken and Daniel Robertz

Lehrstuhl B für Mathematik
RWTH-Aachen

CASC 2007

Introduction (Finite matrix group recognition)

The field approach

Degree steering

Summary

Outline

Introduction (Finite matrix group recognition)

The field approach

Degree steering

Summary

Matrix group recognition project

Given generators for a matrix group G over a finite field F.
Determine the isomorphism type of G.

Matrix group recognition project

Given generators for a matrix group G over a finite field F. Determine the isomorphism type of G.

Usually: G too big to enumerate elements or representatives in subgroup chain. (Typical : $|F| \leq 1000$, degree ≤ 100.)

Matrix group recognition project

Given generators for a matrix group G over a finite field F. Determine the isomorphism type of G.

Usually: G too big to enumerate elements or representatives in subgroup chain. (Typical : $|F| \leq 1000$, degree ≤ 100.)

Method (C. Leedham-Green e. a.) : Apply

- the classification of finite simple groups,
- general structure theorems for matrix groups
- what is known about the representation of the finite simple groups

Easy Example

Typical task: Decide whether a given matrix B is conjugate to the Kronecker product of two matrices X, Y of smaller degrees n, m.

Easy Example

Typical task: Decide whether a given matrix B is conjugate to the Kronecker product of two matrices X, Y of smaller degrees n, m.

Question ($n \otimes m$-problem): What are the resulting conditions for the characteristic polynomial χ_B of B?

Easy Example

Typical task: Decide whether a given matrix B is conjugate to the Kronecker product of two matrices X, Y of smaller degrees n, m.

Question ($n \otimes m$-problem): What are the resulting conditions for the characteristic polynomial χ_B of B?

Example: $2 \otimes 2$-problem. Let

$$\chi_B(t) := t^4 - b_1 t^3 + b_2 t^2 - b_3 t + b_4$$

be the characteristic polynomial of $B \in K^{4\times 4}$. If B is the Kronecker product of two matrices $X, Y \in K^{2\times 2}$ with

$$\chi_X(t) := t^2 - x_1 t + x_2, \qquad \chi_Y(t) := t^2 - y_1 t + y_2.$$

Example (cont.)

Resulting equations:

$$
\begin{aligned}
b_1 &= x_1 y_1 \\
b_2 &= -2\, x_2 y_2 + {y_1}^2 x_2 + {x_1}^2 y_2 \\
b_3 &= y_1 x_2 x_1 y_2 \\
b_4 &= {x_2}^2 {y_2}^2
\end{aligned}
$$

Example (cont.)

Resulting equations:

$$\begin{aligned} b_1 &= x_1 y_1 \\ b_2 &= -2\, x_2 y_2 + {y_1}^2 x_2 + {x_1}^2 y_2 \\ b_3 &= y_1 x_2 x_1 y_2 \\ b_4 &= {x_2}^2 {y_2}^2 \end{aligned}$$

eliminate x_1, x_2, y_1, y_2 to obtain

$$-{b_3}^2 + {b_1}^2 b_4 = 0 \qquad (*)$$

as unique generating relation for the b_i. Hence:

Example (cont.)

Resulting equations:

$$\begin{aligned} b_1 &= x_1 y_1 \\ b_2 &= -2\, x_2 y_2 + {y_1}^2 x_2 + {x_1}^2 y_2 \\ b_3 &= y_1 x_2 x_1 y_2 \\ b_4 &= {x_2}^2 {y_2}^2 \end{aligned}$$

eliminate x_1, x_2, y_1, y_2 to obtain

$$-{b_3}^2 + {b_1}^2 b_4 = 0 \qquad (*)$$

as unique generating relation for the b_i. Hence:

Proposition

$t^4 - b_1 t^3 + b_2 t^2 - b_3 t + b_4$ is characteristic polynomial of a Kroecker product of two 2×2-matrices iff $(*)$ holds.

$2 \otimes 3$-problem

Equations:

$$\begin{aligned} b_1 &= x_1 y_1 \\ b_2 &= -2\, x_2 y_2 + {y_1}^2 x_2 + {x_1}^2 y_2 \\ b_3 &= -3\, x_1 x_2 y_3 + y_1 x_2 x_1 y_2 + {x_1}^3 y_3 \\ b_4 &= -2\, {x_2}^2 y_3 y_1 + x_2 y_1 {x_1}^2 y_3 + {x_2}^2 {y_2}^2 \\ b_5 &= {x_2}^2 y_3 x_1 y_2 \\ b_6 &= {x_2}^3 {y_3}^2 \end{aligned}$$

Theorem

(R. Schwingel 1999) $t^6 - b_1 t^5 + b_2 t^4 - b_3 t^3 + b_4 t^2 - b_5 t + b_6$ *is characteristic polynomial of a Kroecker product of a* 2×2*-matrix with a* 3×3*-matrix iff certain* 16 *polynomials in the* b_i *are satisfied of degrees between* 19 *and* 30*, where* $\deg(b_i) := i$.

Comments

- The result was obtained at the time with MAGMA (about 1 week running time)

Comments

- The result was obtained at the time with MAGMA (about 1 week running time)
- With Involutive and/or GINV we can now do it less than 5 minutes running time and even obtain the Hilbert-series:

$$(1 + t^5 + t^6 + t^{10} + t^{11} + t^{12} + t^{15} + t^{16} + t^{17} + t^{18} \\ -t^{19} - t^{21} - t^{22} - 2\,t^{23} - t^{25} + t^{26} + t^{27} + t^{29} - t^{30}) \,/ \\ ((1-t)(1-t^2)(1-t^3)(1-t^4))\,.$$

Comments

- The result was obtained at the time with MAGMA (about 1 week running time)
- With Involutive and/or GINV we can now do it less than 5 minutes running time and even obtain the Hilbert-series:

$$\begin{gathered}(1+t^5+t^6+t^{10}+t^{11}+t^{12}+t^{15}+t^{16}+t^{17}+t^{18}\\ -t^{19}-t^{21}-t^{22}-2\,t^{23}-t^{25}+t^{26}+t^{27}+t^{29}-t^{30})\,/\\ \left((1-t)(1-t^2)(1-t^3)(1-t^4)\right).\end{gathered}$$

- We can do the full $2 \otimes 4$-problem by first restricting to determinant 1.

Comments

- The result was obtained at the time with MAGMA (about 1 week running time)
- With Involutive and/or GINV we can now do it less than 5 minutes running time and even obtain the Hilbert-series:

$$(1 + t^5 + t^6 + t^{10} + t^{11} + t^{12} + t^{15} + t^{16} + t^{17} + t^{18}$$
$$-t^{19} - t^{21} - t^{22} - 2\,t^{23} - t^{25} + t^{26} + t^{27} + t^{29} - t^{30}) \,/$$
$$((1-t)(1-t^2)(1-t^3)(1-t^4))\,.$$

- We can do the full $2 \otimes 4$-problem by first restricting to determinant 1.
- We can do the full $3 \otimes 3$-problem with determinant 1.

Comments

- The result was obtained at the time with MAGMA (about 1 week running time)
- With Involutive and/or GINV we can now do it less than 5 minutes running time and even obtain the Hilbert-series:

$$(1 + t^5 + t^6 + t^{10} + t^{11} + t^{12} + t^{15} + t^{16} + t^{17} + t^{18} \\ -t^{19} - t^{21} - t^{22} - 2\,t^{23} - t^{25} + t^{26} + t^{27} + t^{29} - t^{30}) \,/ \\ \left((1-t)(1-t^2)(1-t^3)(1-t^4)\right).$$

- We can do the full $2 \otimes 4$-problem by first restricting to determinant 1.
- We can do the full $3 \otimes 3$-problem with determinant 1.
- The results can be obtained over $\mathbb{Q}$, and -with slightly more work- over $\mathbb{Z}$.

General context for matrix group recognition:

Problem

Given a classical group G defined over a field K of characteristic zero and any finite dimensional representation ρ of G.
Find a generating set of the polynomial relations for the coefficients of the characteristic polynomial $\chi_{\rho(g)}(t)$ of $\rho(g)$, $g \in G$.

General context for matrix group recognition:

Problem

Given a classical group G defined over a field K of characteristic zero and any finite dimensional representation ρ of G.
Find a generating set of the polynomial relations for the coefficients of the characteristic polynomial $\chi_{\rho(g)}(t)$ of $\rho(g)$, $g \in G$.

Rough measures for difficulty:
1.) Krull dimension ($=$ rank of the classical group, e.g. $n-1$ for SL(n, K)). (At present Krull dimension 5 with good luck doable.)
2.) Degree of representation ($=$ number of variables).

General context for matrix group recognition (cont.):

Example

1.) $n \otimes m$*-problem :* $G = \mathsf{GL}(n, K) \times \mathsf{GL}(m, K)$ *(resp.* $G = \mathsf{SL}(n, K) \times \mathsf{SL}(m, K)$*) and* $\rho(X, Y) := X \otimes Y$*.*

General context for matrix group recognition (cont.):

Example

1.) $n \otimes m$*-problem :* $G = \mathsf{GL}(n, K) \times \mathsf{GL}(m, K)$ *(resp.* $G = \mathsf{SL}(n, K) \times \mathsf{SL}(m, K)$*) and* $\rho(X, Y) := X \otimes Y$.
2.) (Tensor square) $G = \mathsf{GL}(n, K)$ *(resp.* $\mathsf{SL}(n, K)$*) and* $\rho(X) = X \otimes X$.

General context for matrix group recognition (cont.):

Example

1.) $n \otimes m$*-problem :* $G = \mathsf{GL}(n, K) \times \mathsf{GL}(m, K)$ *(resp.* $G = \mathsf{SL}(n, K) \times \mathsf{SL}(m, K)$*) and* $\rho(X, Y) := X \otimes Y$*.*
2.) (Tensor square) $G = \mathsf{GL}(n, K)$ *(resp.* $\mathsf{SL}(n, K)$*) and* $\rho(X) = X \otimes X$*.*
3.) (Compound representation) $G = \mathsf{GL}(n, K)$ *(resp.* $\mathsf{SL}(n, K)$*) and* $\rho(X) = \wedge^k X$ *for* $k \leq n$*.*

General context for matrix group recognition (cont.):

Example

1.) $n \otimes m$*-problem :* $G = \mathsf{GL}(n, K) \times \mathsf{GL}(m, K)$ *(resp.* $G = \mathsf{SL}(n, K) \times \mathsf{SL}(m, K)$*) and* $\rho(X, Y) := X \otimes Y$*.*
2.) (Tensor square) $G = \mathsf{GL}(n, K)$ *(resp.* $\mathsf{SL}(n, K)$*) and* $\rho(X) = X \otimes X$*.*
3.) (Compound representation) $G = \mathsf{GL}(n, K)$ *(resp.* $\mathsf{SL}(n, K)$*) and* $\rho(X) = \wedge^k X$ *for* $k \leq n$*.*
4.) (Exterior and (reduced) symmetric square) $G = \mathsf{SO}(n, K)$ *and* ρ *certain constituents of the tensor square.*

Note: These series are excellent for benchmarks!

Easy Example: Tensor square, $n = 2$:

$$\begin{aligned} b_1 &= {x_1}^2 \\ b_2 &= -2\,{x_2}^2 + 2\,{x_1}^2 x_2 \\ b_3 &= {x_1}^2 {x_2}^2 \\ b_4 &= {x_2}^4 \end{aligned}$$

Easy Example: Tensor square, $n = 2$:

$$\begin{aligned} b_1 &= {x_1}^2 \\ b_2 &= -2\,{x_2}^2 + 2\,{x_1}^2 x_2 \\ b_3 &= {x_1}^2 {x_2}^2 \\ b_4 &= {x_2}^4 \end{aligned}$$

After elimination:

$${b_1}^2 b_4 - {b_3}^2, \qquad {b_2}^2 b_3 - 4\,b_1 {b_3}^2 + 4\,b_1 b_2 b_4 + 4\,b_3 b_4,$$
$$b_1 {b_2}^2 - 4\,{b_1}^2 b_3 + 4\,b_2 b_3 + 4\,b_1 b_4, \qquad {b_1}^2 b_2 b_4 - b_2 {b_3}^2,$$
$${b_2}^3 b_3 - 4\,b_1 b_2 {b_3}^2 + 16\,{b_3}^3 - 12\,b_2 b_3 b_4 - 16\,b_1 {b_4}^2,$$
$${b_2}^4 - 16\,{b_1}^2 {b_3}^2 + 32\,b_2 {b_3}^2 - 8\,{b_2}^2 b_4 + 16\,{b_4}^2,$$
$${b_1}^2 {b_2}^2 - 4\,{b_1}^3 b_3 + 4\,b_1 b_2 b_3 + 4\,{b_3}^2$$

Outline

Introduction (Finite matrix group recognition)

The field approach

Degree steering

Summary

Abstract problem:

Given: A field K and n variables $x_1, \dots, x_n$ and m polynomials

$$y_i = p_i(x_1, \dots, x_n) \quad \in \quad K[x_1, \dots, x_n] \quad \text{for } i = 1, \dots, m. \quad (1)$$

Aim: Find a presentation for the subring $K[y] := K[y_1, \dots, y_m]$ of $K[x] := K[x_1, \dots, x_n]$.

Abstract problem:

Given: A field K and n variables $x_1, \ldots, x_n$ and m polynomials

$$y_i = p_i(x_1, \ldots, x_n) \quad \in \quad K[x_1, \ldots, x_n] \quad \text{for } i = 1, \ldots, m. \quad (1)$$

Aim: Find a presentation for the subring $K[y] := K[y_1, \ldots, y_m]$ of $K[x] := K[x_1, \ldots, x_n]$.

Invariants: The difference of m and the *transcendence degree* of $K(y) := K(y_1, \ldots, y_m)$ over K will be called the *deficiency* $d = d(y)$ of the tuple y in $K(x)$.

Abstract problem:

Given: A field K and n variables $x_1, \ldots, x_n$ and m polynomials

$$y_i = p_i(x_1, \ldots, x_n) \quad \in \quad K[x_1, \ldots, x_n] \quad \text{for } i = 1, \ldots, m. \quad (1)$$

Aim: Find a presentation for the subring $K[y] := K[y_1, \ldots, y_m]$ of $K[x] := K[x_1, \ldots, x_n]$.

Invariants: The difference of m and the *transcendence degree* of $K(y) := K(y_1, \ldots, y_m)$ over K will be called the *deficiency* $d = d(y)$ of the tuple y in $K(x)$.

Assumption: K perfect, so that the deficiency $d(y)$ can be computed from the rank of the Jacobian matrix $J := (\frac{\partial y_i}{\partial x_j}) \in K(x)^{m \times n}$, viz. $d(y) = m - \mathrm{rank}(J)$.

Elimination via field extension

Technical assumption: $[K(x) : K(y)]$ is a finite field extension.

Elimination via field extension

Technical assumption: $[K(x) : K(y)]$ is a finite field extension.

Algorithm

Input: Equations (1).
Output: Field presentation for $K(y)$

Elimination via field extension

Technical assumption: $[K(x) : K(y)]$ is a finite field extension.

Algorithm

Input: Equations (1).
Output: Field presentation for $K(y)$
Algorithm: Step 0: Choose maximal algebraically independent subset of the y_i*, e. g.* $\{y_1, \ldots, y_n\}$*, and define*
$K_0 := K(y_1, \ldots, y_n)$.

Elimination via field extension

Technical assumption: $[K(x) : K(y)]$ is a finite field extension.

Algorithm

Input: Equations (1).
Output: Field presentation for $K(y)$
Algorithm: Step 0: Choose maximal algebraically independent subset of the y_i, e. g. $\{y_1, \ldots, y_n\}$, and define $K_0 := K(y_1, \ldots, y_n)$.
Step i: Find a presentation for $K_i := K_{i-1}(y_{n+i})$ by computing the minimal polynomial of y_{n+i} over K_{i-1}.

Elimination via field extension

Technical assumption: $[K(x) : K(y)]$ is a finite field extension.

Algorithm

Input: Equations (1).
Output: Field presentation for $K(y)$
Algorithm: Step 0: Choose maximal algebraically independent subset of the y_i, e. g. $\{y_1, \ldots, y_n\}$, and define $K_0 := K(y_1, \ldots, y_n)$.
Step i: Find a presentation for $K_i := K_{i-1}(y_{n+i})$ by computing the minimal polynomial of y_{n+i} over K_{i-1}.

Note: We can now check any relation among the y_i, can even generate relations, but have no K-algebra presentation of $K[y]$.

From field to ring presentation

First idea: Define an ascending chain of ideals

$$I_0 \subsetneq I_1 \subsetneq \ldots \subsetneq I_f \trianglelefteq K[Y_1, \ldots, Y_m]$$

such that I_0 is gerated by the numerators of the relators for the presentation of $K(y)$ and $K[Y_1, \ldots, Y_m]/I_f \cong K[y]$ as follows:

From field to ring presentation

<u>First idea</u>: Define an ascending chain of ideals

$$I_0 \subsetneq I_1 \subsetneq \ldots \subsetneq I_f \trianglelefteq K[Y_1, \ldots, Y_m]$$

such that I_0 is gerated by the numerators of the relators for the presentation of $K(y)$ and $K[Y_1, \ldots, Y_m]/I_f \cong K[y]$ as follows:

Run the Janet-Algorithm twice for I_i,

- over K to obtain $K[Y]/I_i$
- and over $K(y_1, \ldots, y_n)$ to see which denominators $d \in K[Y_1, \ldots, Y_m]$ turn up
- enlarge I_i to I_{i+1} by the kernel of the multiplication with d on $K[Y_1, \ldots, Y_m]/I_i$, in case it is not injective.

From field to ring presentation

<u>First idea</u>: Define an ascending chain of ideals

$$I_0 \subsetneq I_1 \subsetneq \ldots \subsetneq I_f \trianglelefteq K[Y_1, \ldots, Y_m]$$

such that I_0 is gerated by the numerators of the relators for the presentation of $K(y)$ and $K[Y_1, \ldots, Y_m]/I_f \cong K[y]$ as follows:

Run the Janet-Algorithm twice for I_i,

- over K to obtain $K[Y]/I_i$
- and over $K(y_1, \ldots, y_n)$ to see which denominators $d \in K[Y_1, \ldots, Y_m]$ turn up
- enlarge I_i to I_{i+1} by the kernel of the multiplication with d on $K[Y_1, \ldots, Y_m]/I_i$, in case it is not injective.

Stop, when all kernels are trivial.

From field to ring presentation (discussion)

- The problem is a very special case of a primary decomposition.

From field to ring presentation (discussion)

- The problem is a very special case of a primary decomposition.
- For big examples, i. e. $n > 2$ it is too slow.

From field to ring presentation (discussion)

- The problem is a very special case of a primary decomposition.
- For big examples, i. e. $n > 2$ it is too slow.
- The method and some variations of it can be used to find relators, which can be used to speed up other approaches.

From field to ring presentation (discussion)

- The problem is a very special case of a primary decomposition.
- For big examples, i. e. $n > 2$ it is too slow.
- The method and some variations of it can be used to find relators, which can be used to speed up other approaches.
- Specialization techniques can be used to find good choices for the maximally algebraically independent y_i.

Example: Degrees for $3 \otimes 3$-problem

For the $3 \otimes 3$-problem one has (in the end) Krull dimension $n = 5$ and $m = 9$.

Example: Degrees for $3 \otimes 3$-problem

For the $3 \otimes 3$-problem one has (in the end) Krull dimension $n = 5$ and $m = 9$.

Any 5-element subset $S \subset \{y_1, \ldots, y_9\}$ is algebraically independent.

Example: Degrees for $3 \otimes 3$-problem

For the $3 \otimes 3$-problem one has (in the end) Krull dimension $n = 5$ and $m = 9$.

Any 5-element subset $S \subset \{y_1, \ldots, y_9\}$ is algebraically independent.

By specialization one gets rather quickly the following degrees $[K(y) : K(y_i | y_i \in S)]$:

$$6, 9, 10, 11(2\times), 12(13\times), \ldots, 54, \ldots, 108(4\times), 126(5\times).$$

Outline

Introduction (Finite matrix group recognition)

The field approach

Degree steering

Summary

Degree steering: basics

The most powerful method is similar to Groebner walks and is based on the following easy to prove lemma.

Lemma

Let $J \subseteq K[X_1, \dots, X_n, Y_1, \dots, Y_m]$ *be a Janet basis with respect to some term ordering. For any* $0 \neq p \in K[X_1, \dots, X_n, Y_1, \dots, Y_m]$ *let* $\lambda(p)$ *be its leading monomial. If*

$$J \cap K[Y_1, \dots, Y_m] = \{\, p \in J \mid \lambda(p) \in K[Y_1, \dots, Y_m] \,\},$$

then $J \cap K[Y_1, \dots, Y_m]$ *generates* $\langle J \rangle \cap K[Y_1, \dots, Y_m]$*.*

Degree steering: algorithm

Algorithm

Input: A non-empty finite subset $N \subseteq K[X_1, \ldots, X_n, Y_1, \ldots, Y_m]$.

Degree steering: algorithm

Algorithm

Input: A non-empty finite subset $N \subseteq K[X_1, \ldots, X_n, Y_1, \ldots, Y_m]$*.*
Output: A subset $M \subseteq K[Y_1, \ldots, Y_m]$ *generating* $\langle N \rangle \cap K[Y_1, \ldots, Y_m]$*.*

Degree steering: algorithm

Algorithm

Input: A non-empty finite subset $N \subseteq K[X_1, \ldots, X_n, Y_1, \ldots, Y_m]$.
Output: A subset $M \subseteq K[Y_1, \ldots, Y_m]$ generating $\langle N \rangle \cap K[Y_1, \ldots, Y_m]$.
Algorithm: Run Janet's algorithm for N over K with respect to some degree lexicographical term ordering.
Keep replacing N by this Janet basis and changing the term ordering by increasing the degrees of all the X_i until the criterion of the lemma is satisfied.

Degree steering: algorithm

Algorithm

Input: A non-empty finite subset $N \subseteq K[X_1, \ldots, X_n, Y_1, \ldots, Y_m]$.
Output: A subset $M \subseteq K[Y_1, \ldots, Y_m]$ generating $\langle N \rangle \cap K[Y_1, \ldots, Y_m]$.
Algorithm: Run Janet's algorithm for N over K with respect to some degree lexicographical term ordering.
Keep replacing N by this Janet basis and changing the term ordering by increasing the degrees of all the X_i until the criterion of the lemma is satisfied.
Take $M := N \cap K[Y_1, \ldots, Y_m]$.

Degree steering: discussion

- Degree steering tries to approximate the elimination block order slowly.

Degree steering: discussion

- Degree steering tries to approximate the elimination block order slowly.
- For big examples eliminate only one Y_i at a time.

Degree steering: discussion

- Degree steering tries to approximate the elimination block order slowly.
- For big examples eliminate only one Y_i at a time.
- Degree steering can be applied in more general situations as described in (1).

Degree steering: discussion

- Degree steering tries to approximate the elimination block order slowly.
- For big examples eliminate only one Y_i at a time.
- Degree steering can be applied in more general situations as described in (1).
- Degree steering can be accelerated, if one knows already some relations among the y_i.

Degree steering: discussion

- Degree steering tries to approximate the elimination block order slowly.
- For big examples eliminate only one Y_i at a time.
- Degree steering can be applied in more general situations as described in (1).
- Degree steering can be accelerated, if one knows already some relations among the y_i.
- Degree steering can be used to verify a presentation for the y_i or to complete it, if necessary.

Degree steering: example

Critical run for the $2 \otimes 3$-problem:
variables with degrees:

$y_5 : 10, \quad y_4 : 8, \quad y_3 : 6, \quad y_2 : 4, \quad y_1 : 2, \quad x_2 : 2$

Degree steering: example

Critical run for the $2 \otimes 3$-problem:
variables with degrees:

$y_5 : 10, \quad y_4 : 8, \quad y_3 : 6, \quad y_2 : 4, \quad y_1 : 2, \quad x_2 : 2$

Eliminate x_2 (in less than 2 minutes with GINV):

Degree steering: example

Critical run for the $2 \otimes 3$-problem:
variables with degrees:

$y_5 : 10, \quad y_4 : 8, \quad y_3 : 6, \quad y_2 : 4, \quad y_1 : 2, \quad x_2 : 2$

Eliminate x_2 (in less than 2 minutes with GINV):

Notation: J involutive Basis,

$$J_{\lambda,y} := \{\, p \in J \mid \lambda(p) \in K[Y_1, \ldots, Y_m] \,\},$$

Degree steering: example

Critical run for the $2 \otimes 3$-problem:
variables with degrees:
$y_5 : 10, \quad y_4 : 8, \quad y_3 : 6, \quad y_2 : 4, \quad y_1 : 2, \quad x_2 : 2$

Eliminate x_2 (in less than 2 minutes with GINV):

Notation: J involutive Basis,

$$J_{\lambda,y} := \{\, p \in J \mid \lambda(p) \in K[Y_1, \ldots, Y_m] \,\},$$

degree (x_2)	$\lvert J \cap K[Y] \rvert$	$\lvert J_{\lambda,y} \rvert$	$\lvert J \rvert$
2	0	15	25
11	0	18	109
21	6	19	148
29	21	21	164

Outline

Introduction (Finite matrix group recognition)

The field approach

Degree steering

Summary

Summary:

- Series of test examples for elimination originating from the matrix group recognition project were defined.

Summary:

- Series of test examples for elimination originating from the matrix group recognition project were defined.
- Two approaches to carry out the elimination were described.

Summary:

- Series of test examples for elimination originating from the matrix group recognition project were defined.
- Two approaches to carry out the elimination were described.
- The first approach (elimination via field extensions) generates some relations quickly and gives an estimate of the difficulty of the problem.

Summary:

- Series of test examples for elimination originating from the matrix group recognition project were defined.
- Two approaches to carry out the elimination were described.
- The first approach (elimination via field extensions) generates some relations quickly and gives an estimate of the difficulty of the problem.
- The second approach (degree steering) is a powerful tool for general elimination.

Summary:

- Series of test examples for elimination originating from the matrix group recognition project were defined.
- Two approaches to carry out the elimination were described.
- The first approach (elimination via field extensions) generates some relations quickly and gives an estimate of the difficulty of the problem.
- The second approach (degree steering) is a powerful tool for general elimination.
- Both methods build on involutive division and have been tested using GINV.

Summary:

- Series of test examples for elimination originating from the matrix group recognition project were defined.
- Two approaches to carry out the elimination were described.
- The first approach (elimination via field extensions) generates some relations quickly and gives an estimate of the difficulty of the problem.
- The second approach (degree steering) is a powerful tool for general elimination.
- Both methods build on involutive division and have been tested using GINV.
- The first few problems in each series were solved using GINV.