A combined Gröbner basis and power series approach in inverse Galois theory

Peter Müller

Bonn, March 3, 2015

Mathieu group M_{23} as monodromy group of a polynomial

Properties of M_{23}

- $M_{23} \leq S_{23}$ is 4-transitive on $\{1, 2, \ldots, 23\}$.
- $|M_{23}| = 23 \cdot 22 \cdot 21 \cdot 20 \cdot 48 = 10200960$.
- $[A_{23} : M_{23}] = 1267136462592000$.
- M_{23} is simple.
- M_{23} is self-normalizing in S_{23}.

M_{23} as monodromy group of polynomial

- $\hat{h}(X) \in \mathbb{C}[X]$, such that $\mathrm{Gal}(\hat{h}(X) - t/\mathbb{C}(t)) = M_{23}$,
- (equivalent to) $\mathrm{Mon}(\mathbb{P}^1(\mathbb{C}) \to \mathbb{P}^1(\mathbb{C}), z \mapsto \hat{h}(z)) = M_{23}$.
- Existence: Well known and easy (up to Riemann's existence theorem),
- with unique branching type: $1^7 2^8$, $1^3 2^2 4^4$, 23^1.

Mathieu group M_{23} as monodromy group of a polynomial

Theorem (Elkies 2013)

$K = \mathbb{Q}(\sqrt{23 - 2\sqrt{-23}})$
$\hat{h}(X) = X^{23} +$ *complicated lower order terms* $\in K[X]$:
$\mathrm{Gal}(\hat{h}(X) - t/K(t)) = M_{23}$.

Analytic verification of Galois group

Numerically compute monodromy group of cover $\begin{array}{c} \mathbb{P}^1(\mathbb{C}) \to \mathbb{P}^1(\mathbb{C}) \\ z \mapsto \hat{h}(z) \end{array}$.

Algebraic verification of Galois group, first step

- Pick prime $p > 23$, such that $h(X) = (\hat{h}(X) \bmod p) \in \mathbb{F}_p[X]$.
- Suffices to show: $\mathrm{Gal}(h(X) - t/\mathbb{F}_p(t)) = M_{23}$. (S. Beckmann)
- Easy: $M_{23} \leq \mathrm{Gal}(h(X) - t/\mathbb{F}_p(t))$. (Dedekind)
- Need to decide: $\mathrm{Gal}(h(X) - t/\mathbb{F}_p(t)) = M_{23}$ or A_{23}?

Verification of Galois group

Naive idea, for small $p > 23$:

- M_{23} has two orbits on 5–sets, of lengths 5313 and 28336.
- "Compute" polynomial of degree $\binom{23}{5} = 33649$, whose roots are the 5-sums of roots of $h(X) - t$, and "check" if it has a degree 5313 factor over $\mathbb{F}_p(t)$ ...

Using Weil-bound for points on curves (Elkies)

$$\frac{1}{|G|} = \lim_{p\to\infty} \frac{|\{t_0 \in \mathbb{F}_p \mid h(X) - t_0 \text{ splits into linear factors}\}|}{p}$$

Elkies chose $p = 10^8 + 7$: *The factorization of 10^8 polynomials mod p was a somewhat extravagant computation (two days of CPU time in gp).*

M_{23} and its Steiner system

Steiner system $S = S(4, 7, 23)$

- $\mathfrak{P} = 23$ points, $\mathfrak{B} = 253$ blocks $=$ certain 7-sets from $\mathfrak{P}$
- $|\mathfrak{B}|\binom{7}{4} = \binom{23}{4}$ (any 4-set of points contained in exactly one block)
- $M_{23} = \mathrm{Aut}(S)$ transitive on $\mathfrak{P}$ and $\mathfrak{B}$
- $h(X) - t = \prod_{x\in\mathfrak{P}}(X - x)$. Fix $x_0 \in \mathfrak{P}$, so $t = h(x_0)$.

Associated polynomials

$x \in \mathfrak{P}$ integral over $\mathbb{F}_p[t]$ and $\mathbb{F}_p[x_0]$, hence

$$H(h(x_0), Y) = H(t, Y) = \prod_{B\in\mathfrak{B}} (Y - \sum_{x\in B} x) \in \mathbb{F}_p[t][Y] \quad \text{(degree 253)}$$

$$H_1(x_0, Y) = \prod_{x_0\in B\in\mathfrak{B}} (Y - \sum_{x\in B} x) \in \mathbb{F}_p[x_0][Y] \text{ (degree 77)}$$

$$H_2(x_0, Y) = \prod_{x_0\notin B\in\mathfrak{B}} (Y - \sum_{x\in B} x) \in \mathbb{F}_p[x_0][Y] \text{ (degree 176)}$$

Lemma (essentially)

If $h(X) \in \mathbb{F}_p[X]$ *has degree* 23, *then*

$$\mathrm{Gal}(h(X) - t/\mathbb{F}_p(t)) = M_{23} \iff H(h(X), Y) = H_1(X, Y)H_2(X, Y)$$

for some $H(t, Y) \in \mathbb{F}_p[t][Y]$ *irreducible of degree* 253, *and* $H_1(X, Y), H_2(X, Y) \in \mathbb{F}_p[X][Y]$ *of degrees* 77 *and* 176.

How to compute ...

$$H(t, Y) = \prod_{B \in \mathfrak{B}} (Y - \sum_{x \in B} x)$$

from $h(X)$? Certainly not as a degree 253 factor of the degree $\binom{23}{7} = 245157$ polynomial

$$\prod_{C \in \binom{\mathfrak{P}}{7}} (Y - \sum_{x \in C} x).$$

Laurent series

Computation of ...

$$H(t,Y) = \prod_{B\in\mathfrak{B}} (Y - \sum_{x\in B} x) \in \mathbb{F}_p[t][Y] = \mathbb{F}_p[\tau^{23}][Y]$$

by explicit determination of $\mathfrak{P}$ and $\mathfrak{B}$:

- $h(X) - t = h(X) - \tau^{23} = 0$ has a root

$$L(\tau) = \tau + a_0 + a_1\tau^{-1} + a_2\tau^{-2} + \cdots \in \mathbb{F}_p((1/\tau)).$$

- $\mathfrak{P} = \{L(w\tau) \mid w \in W\}$ where $W \leq \bar{\mathbb{F}}_p^\star$ with $|W| = 23$.
- W acts regularly on $\mathfrak{P}$.
- There are only two continuations of W to an action of M_{23}, so there are only two candidates for $\mathfrak{B}$. One of them works!
- Suffices to work with truncated Laurent series.
- No need to factor $H(h(X),Y)$ to obtain $H_1(X,Y)$ and $H_2(X,Y)$. Work in $\bar{\mathbb{F}}_p((1/x_0))$!

Laurent series

More general case

Want to upper bound $G = \mathrm{Gal}(P(X) - tQ(X)/k(t)) \leq S_n$. Method works best,

- if there is a set B with $2 \leq |B| \leq n-2$, and $[G : G_B]$ small,
- there is an inertia generator with few cycles (hence few potential candidates for B), and
- k is a finite field (otherwise the coefficients of Laurent series explode).

For instance, it works well for Granboulan's M_{24}-polynomial.

Reverting the technique to find polynomials

Using $H_1(x_0, Y)$

$$\mathrm{Gal}(h(X) - t/\mathbb{F}_p(t)) = M_{23} \Longrightarrow \prod_{x_0 \in B \in \mathfrak{B}} (Y - \sum_{x \in B} x) \in \mathbb{F}_p[x_0][Y],$$

hence

$$S_k = \sum_{x_0 \in B \in \mathfrak{B}} (\sum_{x \in B} x)^k \in \mathbb{F}_p[x_0] \text{ for all } k \geq 0.$$

On the other hand, with $x_0 = 1/z$, ω a 23-rd root of unity, and $m > 0$, the roots x_i of $h(X) - t = h(X) - h(1/z)$ are

$$x_i = \frac{\omega^i}{z} + \text{higher order terms} = A_i(z) + O(z^m) \in \mathbb{F}_p((z)), \text{ hence}$$

$$\mathbb{F}_p[1/z] \ni S_k = \sum_{x_0 \in B \in \mathfrak{B}} (\sum_{x_i \in B} A_i(z))^k + O(z^{m+1-k}).$$

Reverting the technique to find polynomials

Strategy:

- Set $h(X) = a_1 X + a_2 X^2 + \cdots + a_{21} X^{21} + X^{23} \in \mathbb{F}_p[\mathbf{a}][X]$.
- For $m > 0$ compute $x_i = A_i(z) + O(z^m) \in \mathbb{F}_p[\mathbf{a}]((z))$.
- For $k = 1, 2, \ldots, m-1$ collect the coefficients of z^j with $j \geq 1$ in $\sum_{x_0 \in B \in \mathfrak{B}} (\sum_{x \in B} A_i(z))^k$. They all have to vanish!
- Solve this system of polynomial equations for the unknowns $\mathbf{a}$.

Results:

- For $p = 47$ get Elkies' polynomial within a few seconds (compared to 46 CPU hours by refined standard approach).
- One can also compute the Laurent series and Gröbner bases over $\mathbb{Q}$ instead of $\mathbb{F}_p$. Then a naive Sage implementation takes a few minutes to get the degree 4 number field over which the polynomial is defined.