http://ijgt.ui.ac.ir

International Journal of Group Theory
ISSN (print): 2251-7650, ISSN (on-line): 2251-7669
Vol. 13 No. 3 (2024), pp. 241-250.

www.ui.ac.ir

COMPUTING GALOIS GROUPS

ANDREAS-STEPHAN ELSENHANS

Dedicated to Daniela Nikolova-Popova on the occasion of her 70th birthday

Abstract. The determination of a Galois group is an important question in computational algebraic number theory. One approach is based on the inspection of resolvents. This article reports on this method and on the performance of the current `magma` [W. Bosma, J. Cannon and C. Playoust, The Magma algebra system. I. The user language, J. Symbolic Comput., 24 (1997) 235–265]. implementation.

1. Introduction

Given a polynomial $f \in \mathbb{Q}[X]$ of degree n, one may ask for its splitting field K and for the Galois group $\mathrm{Gal}(K/\mathbb{Q})$. As the degree of the extension $K/\mathbb{Q}$ may be as large as $n!$, an explicit description of the splitting field might be very complex. However, if we represent the Galois group $\mathrm{Gal}(K/\mathbb{Q})$ as a permutation group of the roots of f, we are simply looking for a subgroup of the symmetric group S_n.

Thus, the challenge is to compute $\mathrm{Gal}(K/\mathbb{Q})$ as a permutation group without constructing K as a number field. One approach for this is to replace K by a local splitting field, i.e. $\mathbb{C}$ or a p-adic field. As this approach results only in approximations of the roots of f, one has to overcome all the difficulties that arise from this. This approach goes at least back to [18]. More recent implementations are described in [12] and [11].

Keywords: Galois group, Resolvent, Numberfield.

MSC(2010): Primary: 12R32; Secondary: 12-08. Communicated by Patrizia Longobardi.

Article Type: 2022 CCGTA IN SOUTH FLA.

Received: 18 January 2023, Accepted: 22 February 2023.

Cite this article: A.-S. Elsenhans, Computing Galois groups, Int. J. Group Theory, **13** no. 3 (2024) 241–250. http://dx.doi.org/10.22108/ijgt.2023.136401.1825 .

The general idea to compute a Galois group is to construct and factorise resolvents. For this, let $r_1, \ldots, r_n$ be the roots of $f \in \mathbb{Z}[X]$. Then there is the homomorphism $\pi\colon \mathrm{Gal}(K/\mathbb{Q}) \to S_n$ given by $\sigma(r_i) = r_{\pi(\sigma)(i)}$, which describes the Galois group of the splitting field K of f as a permutation group.

One has that

$$R(X) := (X - r_1r_2)(X - r_1r_3)\cdots(X - r_{n-1}r_n)$$

is a rational polynomial. It is an example of what is called a resolvent. If the roots of R are pairwise distinct, the factorisation of R in $\mathbb{Q}[X]$ encodes the orbits of $\pi(\mathrm{Gal}(K/\mathbb{Q}))$ on pairs $\{\{i,j\}\colon 1 \le i < j \le n\}$. Using this and similar resolvents, one can determine the Galois group of a polynomial. For polynomials in degree 4 and 7, this is described in detail in [4, Sec. 6.3]. The treatment of the degrees 5 and 6 in [4, Sec. 6.3] requires to check in addition whether some auxiliary expressions are squares.

The main disadvantage of using only absolute resolvents as above is that the information derived from one resolvent is not used in the construction of the next one. To formalise the usage of the information, we have to introduce the notion of a *Galois starting group*, which is an arbitrary subgroup of S_n that contains $\pi(\mathrm{Gal}(K/\mathbb{Q}))$.

The article is structured as follows. First, the concepts of a relative invariant and a relative resolvent are introduced. Next, an outline of the algorithm is given. Then improvements based on the use of subfields are shown. To handle subgroups of large index, the usage of the local Galois group of the splitting field is explained. This requires to introduce the notion of short cosets. As this gives only conditional results, a section follows on the Galois proof algorithm. Finally, a systematic performance test is carried out. The article ends with some open questions.

2. Relative resolvents

Definition 2.1. *Assume that S_n acts on $\mathbb{Z}[X_1, \ldots, X_n]$ by permuting the variables. Let $U \subset G \subset S_n$ be permutation groups. A polynomial $I \in \mathbb{Z}[X_1, \ldots, X_n]$ such that*

$$U = \{\sigma \in G \mid \sigma \circ I = I\}$$

is called a relative invariant for the pair of groups (U, G).

Remark 2.2. (1) *For each pair of groups $U \subset G \subset S_n$ a relative invariant exists. For example, the orbit sum*

$$I := \sum_{\sigma \in U} \sigma \circ \left(X_2X_3^2 \cdots X_n^{n-1}\right)$$

is a relative invariant, independently of what G is.

(2) *A list of the methods used in order to construct simpler invariants in a large variety of special cases is given in* [8].

http://dx.doi.org/10.22108/ijgt.2023.136401.1825

Definition 2.3. *Let f be a polynomial with roots $r_1, \ldots, r_n$ and $G \subset S_n$ be an overgroup of the Galois group of f. Further, let $U \subset G$ be a subgroup and $I \in \mathbb{Z}[X_1, \ldots, X_n]$ be a relative invariant for the pair (U, G). Then*

$$R_{G,U,I} := \prod_{\sigma \in G/U} (X - (\sigma \circ I)(r_1, \ldots, r_n))$$

is called a relative resolvent of f with respect to the pair (U, G). A relative resolvent is called non-degenerate if it has no multiple roots. In the case that $G = S_n$, the resolvent is called absolute.

Remark 2.4.
(1) *Because all invariants have integer coefficients, all relative resolvents are integral, if f is a monic and integral.*
(2) *If the non-degenerate relative resolvent is reducible then the Galois group is a proper subgroup of G.*
(3) *Assume the relative resolvent to be non-degenerate. Then the Galois group is contained in $\sigma U \sigma^{-1}$ if and only if the relative resolvent $R_{G,U,I}$ has the rational root $(\sigma \circ I)(r_1, \ldots, r_n)$.*
(4) *In general, the factors of a non-degenerate resolvent encode the decomposition of G/U into its $\pi(Gal(K/\mathbb{Q}))$-orbits.*

Remark 2.5. *The computation of Galois groups for irreducible polynomials up to degree 7 as described in* [4, Sec. 6.3] *uses resolvents with respect to the following pairs of groups:*
(1) *$A_n \subset S_n$.*
(2) *$C_4 \subset S_4$ and $D_{2\cdot 4} \subset S_4$.*
(3) *$M_{20} \subset S_5$, the affine group of order 20 and $C_5 \subset D_{2\cdot 5}$.*
(4) *$\mathrm{PGL}_2(\mathbb{Z}/5\mathbb{Z}) \subset S_6$, $(S_3 \times S_3) \rtimes S_2 \subset S_6$.*
(5) *The stabiliser $S_3 \times S_4 \subset S_7$ of $\{1, 2, 3\}$.*

In higher degree, one has to deal with a much larger number of subgroups of S_n and the advantage of using relative resolvents gets more visible.

2.1. **Outline of the Algorithm.** The general idea to compute a Galois group of a monic degree n polynomial $f \in \mathbb{Z}[X]$, as described in [18], is as follows:

Input: A monic polynomial $f \in \mathbb{Z}[X]$ of degree n.

Output: p-adic or complex roots $r_1, \ldots, r_n$ and $\pi(\mathrm{Gal}(K/\mathbb{Q})) \subset S_n$.

Steps:

(1) Choose a p-adic splitting field or the complex numbers.
(2) Compute the roots $r_1, \ldots, r_n$ of f in the field chosen.
(3) Set $G := S_n$ as a Galois starting group.
(4) For each conjugacy class of maximal subgroups of G, do the following:
   (a) Pick a representative U of the conjugacy class.
   (b) Compute a relative invariant I for the pair (U, G).

http://dx.doi.org/10.22108/ijgt.2023.136401.1825

(c) Compute all the roots of the relative resolvent $R_{G,U,I}$.

(d) If a root has multiplicity larger than 1, restart with a modified invariant.

(e) If $\sigma \circ I(r_1, \ldots, r_n)$ is a simple rational root of the relative resolvent, then replace G by $\sigma U \sigma^{-1}$ and restart step 4.

(5) If G is no longer replaced by a smaller group then return G as the Galois group of f acting on the roots $r_1, \ldots, r_n$.

Remark 2.6. (1) *Using the product formula of valuation theory, one can prove that a p-adic root approximation relates to a rational root of a resolvent* [12, Satz 3.33]. *This requires to use a p-adic precision that is proportional to the index of the inspected subgroup.*

(2) *In the case of a subgroup of large index, one can restrict to an inspection of all the conjugates of U that contain the p-adic Galois group. The corresponding cosets in G/U are sometimes called short cosets. Some details on this are given in Sec. 4. This approach does not guarantee that all the resolvent roots are distinct.*

(3) *The use of a low precision and the inspection of only some of the cosets does no longer result in a proven Galois group. However, the result is correct with a high probability an can be verified by another method.*

3. Block systems of the Galois group

3.1. **Introduction.** The polynomial f is irreducible if and only if its Galois group is transitive on its roots. A transitive group may have block systems. They correspond to the intermediate groups of $\mathrm{Stab}_G(1) \subset G$. Under the Galois correspondence, $\mathrm{Stab}_G(1)$ corresponds to the field $\mathbb{Q}[X]/(f)$. Thus, the intermediate fields of $\mathbb{Q} \subset \mathbb{Q}[X]/(f)$ are in one to one correspondence with the block systems of the Galois group.

3.2. **Computing subfields.** There are two different approaches to compute subfields of a number field. One is combinatorial. It first enumerates all potential block systems and then checks which of them are indeed block systems [16]. The running time depends heavily on the example considered.

A second approach is based on an LLL-computation that guesses a generator of a subfield. This approach determines a generating set of the subfield lattice in polynomial time [14]. In practice, we use a combination of the two approaches [10].

3.3. **Usage of subfields for the Galois group computation.** After a determination of the subfields of $\mathbb{Q}[X]/(f)$, we know all the block systems of the Galois group. Therefore, the Galois group is contained in all the wreath products of symmetric groups that correspond to these block systems and we can use their intersection as a Galois starting group.

http://dx.doi.org/10.22108/ijgt.2023.136401.1825

One can refine this starting group further, by testing, whether any of the discriminants of the subfields are squares or any product of these discriminants is a square. See [10] for a detailed description of the entire starting group algorithm.

4. **Short Cosets**

Remark 4.1. *Assume that an element $g \neq \mathrm{id}$ of a permutation group $G \subset S_n$ is known. Further, let a subgroup $U \subset G$ be given. Then the probability of $g \in U$ can be estimated by $[G : U]^{-1}$. Thus, we expect the set*

$$S_{G,U,g} := \{\overline{h} \in G/U \mid g \in hUh^{-1}\} = \{\overline{h} \in G/U \mid h^{-1}gh \in U\}$$

to be small even when $[G : U]$ is very large.

Definition 4.2. *The set $S_{G,U,g}$ is called the set of short cosets of G/U with respect to the element g.*

Remark 4.3. (1) *In practice, one uses this as follows. Assume that the roots of f are given in an unramified p-adic extension $K_p/\mathbb{Q}_p$. Then the lift of the Frobenius automorphism generates $Gal(K_p/\mathbb{Q}_p)$. The latter group is known to be a subgroup of the Galois group.*

Thus, when checking whether one of the conjugates $gUg^{-1} \subset G$ contains the Galois group $Gal(K/\mathbb{Q})$ to be computed, it suffices to check the conjugation for $\overline{g} \in S_{G,U,\pi(Frob(K_p/\mathbb{Q}_p))}$.

(2) *Methods to compute short cosets efficiently are described in* [12, Sec. 5.2] *and* [9]. *In* `magma`, *they can be called via* `ShortCosets`.

5. **Proving the result**

5.1. **Introduction.** When we run Algorithm 2.1 in practice, we might have to use a highly likely, but unproven group as an overgroup of the Galois group. This leads to an unproven result G. Thus, it has to be confirmed that the Galois group is contained in G. The possibility that the Galois group is a proper subgroup of G is already excluded. This can be done by verifying that resolvent polynomials factor as predicted by G. In detail, we do the following:

Algorithm 1. *(Proof of Galois group)*
Input: *Polynomial $f \in \mathbb{Z}[X]$ with roots $r_1, \ldots, r_n$ and a permutation group $G \subsetneq S_n$ as unproven Galois group.*
Output: `Galois group is proven` *or* `Galois group is disproven`.
Steps:

(1) *Set $H := S_n$.*
(2) *Search for a subgroup of small index $U \subset H$ such that G does not act transitively on H/U. We denote the G-orbits by $O_1, \ldots, O_k$.*
(3) *Compute a relative invariant I and a non-degenerate resolvent $R_{H,U,I} \in \mathbb{Z}[X]$ of f with respect to the pair of groups (U, H).*

http://dx.doi.org/10.22108/ijgt.2023.136401.1825

(4) *Assuming that G coincides with $\pi(Gal(K/\mathbb{Q}))$, the irreducible factors of $R_{H,U,I}$ in $\mathbb{Z}[X]$ are in one-to-one correspondence with the orbits $O_1, \ldots, O_k$. Therefore, compute p-adic approximations of the predicted factors*

$$T_i := \prod_{\sigma \in O_i} (X - (\sigma I)(r_1, \ldots, r_n)) \text{ for } i = 1, \ldots, k$$

of $R_{H,U,I}$ and reconstruct them in $\mathbb{Z}[X]$.

(5) *Check whether the reconstructed polynomials are in fact divisors of $R_{H,U,I}$. If not then return* `Galois group is disproven`.

(6) *If the factorisation of $R_{H,U,I}$ is confirmed then replace H by $Stab_H(O_1, \ldots, O_k)$, as this stabiliser is now proven to contain $\pi(Gal(K/\mathbb{Q}))$.*

(7) *If H coincides with G then return* `Galois group is proven`.

(8) *Restart at step 2 with the new group H.*

Remark 5.1. *To pick U efficiently, we use the following strategy:*

(1) *In the first iteration of the proof, we have $H = S_n$. Therefore, in most cases, an intransitive group of the shape $U_k := S_k \times S_{n-k} \subset S_n$ for $k \in \{2, 3, 4\}$ is the best possible choice. The resolvent for the pair (U_k, S_n) is sometimes called the k-set resolvent. It can be computed symbolically* [2] *by using* `MSet` *in* `magma`.

The choice $k = 2$ will work for all imprimitive groups and all block systems of G will be confirmed.

(2) *Assume H has the block system $\{B_1, \ldots, B_k\}$. Then we have an intermediate field M with $(\mathbb{Q}[X]/(f)) \supsetneq M \supsetneq \mathbb{Q}$. We first prove $Gal(M/\mathbb{Q})$ and $Gal((\mathbb{Q}[X]/(f)) / M)$. For this, we choose U as follows:*

Proof of Gal$(M/\mathbb{Q})$: *Write $\phi\colon H \to S_k$ for the action on the blocks. If $\phi(H)$ and $\phi(G)$ do not coincide then pick a low index subgroup $V \subset \phi(H)$ such that $\phi(G)$ is not transitive on $\phi(G)/V$. Then use $U := \phi^{-1}(V)$.*

Proof of Gal$((\mathbb{Q}[X]/(f)) / M)$: *Set $S := Stab_H(B_1)$ and write $\phi\colon S \to Sym(B_1)$ for the action of S on B_1. If $\phi(S)$ and $\phi(S \cap G)$ do not coincide then pick a low index subgroup $V \subset \phi(S)$ such that $\phi(S \cap G)$ is not transitive on $\phi(S)/V$. Then use $U := \phi^{-1}(V)$.*

(3) *If the above strategies do not apply, a direct search for U within all the low index subgroups of H works usually in a reasonable amount of time.*

6. Practical performance test

6.1. **Introduction.** The running time in practical examples depends at least on the following:

(1) The size of the coefficients of the input polynomial.

(2) The complexity of the relative invariants used. I.e., its degree and the number of multiplications necessary for an evaluation.

http://dx.doi.org/10.22108/ijgt.2023.136401.1825

(3) The p-adic precision and the degree of the p-adic splitting field.
(4) The number of short cosets that have to be inspected.

6.2. **Performance in degree 20.** To get an impression, we pick one polynomial for each of the 1117 transitive groups in degree 20 of the database [17] and compute the Galois group in each case. It takes 455 seconds in total on a current PC, using only one core, to determine all the unproven Galois groups. The slowest example takes 6 seconds. This is explained by its large coefficients. The average running time of the proof is smaller than the running time of the computation of the group.

An atypically slow example with small coefficients is

$$P_{20}^{231} = X^{20} + 4X^{15} + 3X^{10} - 2X^5 + 2\,.$$

The computation takes 1 second. In detail, we have:

- 6 times, G is replaced by a smaller group (descents). Once this is unproven.
- 14 times, an invariant has to be modified as the resolvent has multiple roots.
- 12 conjugacy classes of subgroup are ruled out.
- 0.5 seconds are taken for constructing invariants.

Running the proof afterwards takes 0.3 seconds.

An atypically slow example for the Galois proof algorithm is

$$\begin{aligned} &X^{20} - 8X^{19} + 5510X^{16} + 12160X^{15} + 29450X^{14} + 41800X^{13} + 752305X^{12} \\ &+ 6263920X^{11} + 22326330X^{10} + 59864440X^9 + 519659025X^8 \\ &+ 1540067800X^7 + 3729981200X^6 + 5914614880X^5 + 7255193320X^4 \\ &+ 4136148000X^3 + 364876000X^2 - 1916943360X + 425342480 \end{aligned}$$

with group $T_{20}^{272} = \mathrm{PSL}_2(\mathbb{F}_{19})$. Here, the group is determined in a about half a second. The proof takes about 40 seconds, as it has to inspect the 4-set resolvent of degree 4845.

6.3. **A test based on the database of transitive groups.** In earlier implementations, the complexity of the invariants was the main bottleneck. Using a database of groups, one can compute relative invariants for all the maximal subgroups of the groups in the database and compare the invariants by size. Doing this for the transitive group database up to degree 30 and all the transitive maximal subgroups results in an inspection of 253085 pairs of groups and takes about 35 minutes of CPU time.

In only 34 cases, the invariant involves more than 250 multiplications. The average degree is about 7.5. The largest degree of an invariant is 435. This is the smallest possible degree of a relative invariant for the pair $A_{30} \subset S_{30}$ [5, Ex. 2.6.5]. Examples with an exceptionally complex invariant are given in the table below.

http://dx.doi.org/10.22108/ijgt.2023.136401.1825

Groups	Index	$\deg(I)$	Evaluation costs
$\mathrm{P\Gamma L}_2(\mathbb{F}_8) \subset A_9$	120	6	387 Multiplications
$\mathrm{M}_{24} \subset A_{24}$	1267136462592000	6	759 Powers
$\mathrm{T}_{30}^{1153} \subset \mathrm{T}_{30}^{4863}$	20160	8	5221 Multiplications

6.4. **Invariants for M_{24}.** The Mathieu group M_{24} corresponds to the $(5, 8, 24)$ Steiner system. This Steiner system consists of 759 subsets of size 8 in $\{1, \ldots, 24\}$. For simplicity, we assume that one of these subsets is $\{1, \ldots, 8\}$. Its stabiliser is an index 759 subgroup $U_{759} \subset M_{24}$. Using this, we get the following relative invariants for $M_{24} \subset A_{24}$

$$I_s = \sum_{\sigma \in M_{24}/U_{759}} \sigma \circ (X_1 + \cdots + X_8)^6 ,$$

$$I_p = \sum_{\sigma \in M_{24}/U_{759}} \sigma \circ X_1X_2X_3X_4X_5X_6X_7X_8 .$$

As the implementation also supports polynomials with coefficients in $\mathbb{F}_p(t)$, the following example uses I_p. Note that I_s degenerates in small characteristic.

```
> F<t> := FunctionField(GF(2));
> R<x> := PolynomialRing(F);
> f := x^24 + x + t;
> time G,r,S := GaloisGroup(f);
Time: 3.090
> IsPrimitive(G);
true
> #G;
244823040
```

In this example, about one second of CPU time is used for the evaluation of all invariants used.

6.5. **A degree 30 polynomial.** The list above indicates that a test of a degree 30 polynomial having Galois group T_{30}^{1153} is worth a trial. As this group is isomorphic to S_8, one can build an example in `magma` as follows:

```
// Construct a degree 30 polynomial with group S_8:
> r<x> := PolynomialRing(Rationals());
> f8 := x^8+x+2;
> ram := PrimeDivisors(Integers()!Discriminant(f8));
> G,r,S := GaloisGroup(f8);
> G eq Sym(8);
true
> u30 := LowIndexSubgroups(G,<30,30>);
```

http://dx.doi.org/10.22108/ijgt.2023.136401.1825

```
> f30 := GaloisSubgroup(S,u30[1]);

// Use the maximal order to find a nice degree 30 polynomial:
> ord := LLL(MaximalOrder(f30:Ramification := ram));
> f30_red := MinimalPolynomial(ord.2);

// Compute the Galois group in degree 30:
> G2,r2,S2 := GaloisGroup(f30_red);
> GaloisProof(f30_red,S2);
true
> TransitiveGroupIdentification(G2);
1153 30
```

The computation of the Galois group takes about 4.4 seconds on a current standard PC. The most expensive steps are:

- 1.0 seconds for the determination of subfields and block systems.
- 1.1 seconds to compute p-adic root approximations with the required precision.
- 0.25 seconds to determine all the invariants used.
- 1.7 seconds to evaluate the invariants.

Thus, even in this case, the complexity of the invariant does not turn into a bottleneck.

7. **Further work**

The current implementation of the Galois group algorithm performs well in a large variety of examples. However, there are still open questions.

Question 7.1. *Is the computation of Galois groups using only resolvents a polynomial time algorithm?*

Question 7.2. *Given a polynomial $f \in \mathbb{Z}[X]$ such that the Galois group is a primitive permutation group of affine type (i.e., a primitive group with abelian socle). Is there an algorithmic approach for this special case? For instance, can an LLL-type computation (or any other approach) be used the determine a primitive affine permutation group $\mathrm{AGL}_n(\mathbb{Z}/p\mathbb{Z}) \subset S_{p^n}$ that (most likely) contains the Galois group?*

Currently, it seems that polynomials with a primitive affine Galois groups such as

`PrimitiveGroup(49,22)` or `PrimitiveGroup(64,42)`

are atypically hard to treat. For both groups, the algorithm runs into an irreducible 2-set resolvent and the number of short cosets is atypically large. Thus, a new approach to compute a Galois starting group in these cases would be helpful.

http://dx.doi.org/10.22108/ijgt.2023.136401.1825

References

[1] W. Bosma, J. Cannon and C. Playoust, The Magma algebra system. I. The user language, *J. Symbolic Comput.*, **24** (1997) 235 – 265.

[2] A. Bostan, P. Flajolet, B. Salvy and E. Schost, Fast computation of special resultants, *J. Symbolic Comput.*, **41** (2006) 1 – 29.

[3] J. Cannon and D. Holt, Computing maximal subgroups of finite groups, *J. Symbolic Comput.*, **37** (2004) 589 – 609.

[4] H. Cohen, *A course in computational algebraic number theory*, Springer, Berlin 1993.

[5] H. Derksen and G. Kemper, *Computational invariant theory*, Encyclopaedia of Mathematical Sciences, **130**, Springer, Berlin, 2002.

[6] J. Dixon and B. Mortimer, *Permutation groups*, Graduate Texts in Mathematics, **163**, Springer, New York, 1996.

[7] A.-S. Elsenhans, Invariants for the computation of intransitive and transitive Galois groups, *J. Symbolic Comput.*, **47** (2012) 315 – 326.

[8] A.-S. Elsenhans, Improved methods for the construction of relative invariants for permutation groups, *J. Symbolic Comput.*, **79** (2017) 211 – 231.

[9] A.-S. Elsenhans, A note on short cosets, *Exp. Math.*, **23** (2014) 411 – 413.

[10] A.-S. Elsenhans and J. Klüners, Computing subfields of number fields and applications to Galois group computations, *J. Symbolic Comput.*, **93** (2018) 1 – 20.

[11] C. Fieker and J. Klüners, Computation of Galois groups of rational polynomials, *LMS J. Comput. Math.*, **17** (2014) 141 – 158.

[12] K. Geißler, *Berechnung von Galoisgruppen über Zahl- und Funktionenkörpern*, Dissertation, Berlin, 2003.

[13] W. Hart, M. van Hoeij and A. Novocin, *Practical polynomial factoring in polynomial time*, ISSAC 2011–Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation, New York, 2011 163 – 170.

[14] M. van Hoeij, J. Klüners and A. Novocin, Generating subfields, *J. Symbolic Comput.*, **52** (2013) 17 – 34.

[15] B. Huppert, *Endliche Gruppen. I*, Springer, Berlin 1967.

[16] J. Klüners, On computing subfields. A detailed description of the algorithm, *J. Théor. Nombres Bordeaux*, **10** (1998) 243 – 271.

[17] J. Klüners and G. Malle, A database for field extensions of the rationals, *LMS J. Comput. Math.*, **4** (2001) 182 – 196.

[18] R. Stauduhar, The determination of Galois groups, *Math. Comp.*, **27** (1973) 981 – 996.

Andreas-Stephan Elsenhans

Department of Mathematics, University of Würzburg, Würzburg, Germany

Email: stephan.elsenhans@mathematik.uni-wuerzburg.de

http://dx.doi.org/10.22108/ijgt.2023.136401.1825