J. Symbolic Computation (2000) **11**, 1–23

Galois Group Computation for Rational Polynomials

KATHARINA GEISSLER AND JÜRGEN KLÜNERS

Technische Universität Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany
Universität Heidelberg, Im Neuenheimer Feld 368, 69120 Heidelberg, Germany

(Received 25 February 2000)

We describe methods for the computation of Galois groups of univariate polynomials over the rationals which we have implemented up to degree 15. These methods are based on Stauduhar's algorithm. All computations are done in unramified p-adic extensions. For imprimitive groups we give an improvement using subfields. In the primitive case we use known subgroups of the Galois group together with a combination of Stauduhar's method and the absolute resolvent method.

1. Introduction

Let $f \in \mathbb{Z}[x]$ be a monic irreducible polynomial. Algorithms for the computation of the Galois group $\mathrm{Gal}(f)$ of f are an important tool of constructive number theory. Deterministic exponential time algorithms were already used more than 100 years ago (see (Tschebotarew and Schwerdtfeger, 1950)). Nevertheless until today no general polynomial time algorithm is known. In this paper we restrict ourselves to the case of univariate, irreducible polynomials over $\mathbb{Q}$. By applying suitable transformations we assume that we have monic polynomials with integer coefficients.

All practical algorithms use the classification of transitive groups, which is known up to degree 31 (Hulpke, 1996). These algorithms can be divided into the absolute resolvent method (Soicher, 1981; Soicher and McKay, 1985; Mattman and McKay, 1997) and the method of Stauduhar (Stauduhar, 1973). From the coefficients of the given polynomial it is possible to compute so-called absolute resolvents (Casperson and McKay, 1994). The factorization of these resolvents gives lots of information about the Galois group which may be enough to identify it. In general the degrees of these resolvents can be huge compared to the degree of the given polynomial. Therefore for higher degrees (say larger than 11) it is very expensive to compute these factorizations. Another disadvantage of this approach is that we only get the name of the Galois group, but no explicit action on the roots. To know these actions is an important ingredient of the algorithms presented in Klüners and Malle (2000). There are implementations of this method in Maple (Mattman and McKay, 1997) and Gap (Schönert *et al.*, 1997).

The Stauduhar method uses so-called relative resolvents which are computed using approximations of the roots of the given polynomial. It computes the Galois group including the action on the roots. We give a detailed description of this method in the next section. There are implementations of this method in Pari (Eichenlaub and Olivier, 1995) (up to

0747–7171/90/000000 + 00 \$03.00/0 © 2000 Academic Press Limited

degree 11) and KANT (Geißler, 1997) (up to degree 15) which use complex approximations of the roots. The disadvantage of complex approximations is that we need a very high precision to get proven results. This makes this approach inefficient. Yokoyama (1997) suggests using p-adic approximations. There is an implementation of this method in the computer algebra system ASIR up to degree 8.

In this paper we describe Stauduhar's method using p-adic approximations. Looking at degrees 12 to 15 it turns out that the ordinary method is not efficient enough to compute the Galois group. The goal was to solve this defect in order to treat higher degree polynomials within reasonable time. One important improvement is the use of subfields of a stem field of f, that is the field extension of $\mathbb{Q}$ which we get by adjoining a root of f to $\mathbb{Q}$. Klüners and Pohst (1997), Klüners (1998) give efficient algorithms to compute subfields. Using this information we obtain that the Galois group is a subgroup of the intersection of suitable wreath products which can be computed easily. This intersection is a good starting point for our algorithm. In the case of primitive groups this method gives no improvement. Here we present a combination of the method of Stauduhar and the absolute resolvent method to compute the Galois group. As mentioned before we use p-adic approximations of the roots. The Frobenius automorphism of the underlying p-adic field already determines a subgroup of the Galois group, which can be used to speed up the computations dramatically.

The algorithms presented in this paper are implemented in the computer algebra system KANT (Daberkow *et al.*, 1997). We give examples for all transitive groups of degree 12 to 15. In most examples the computing time is only a few seconds.

We remark that in the case that the stem field is normal or even abelian there are efficient algorithms to compute the automorphism group (Acciaro and Klüners, 1999; Klüners, 1997). Since the factorization of polynomials over number fields is in polynomial time (Lenstra *et al.*, 1982; Landau, 1985) the computation of the automorphism group of a normal field is possible in polynomial time. Landau and Miller (1985) show how to decide the question of solvability in polynomial time. To our knowledge there do not exist efficient implementations of these polynomial time algorithms.

2. The method of Stauduhar

The main purpose of this section is to recall the essential components of the method of Stauduhar and to introduce some notation. In general, Stauduhar's method (see Stauduhar (1973)) is based on so-called resolvents, that is, polynomials whose splitting fields are subfields of the splitting field of the given polynomial $f \in \mathbb{Z}[x]$, whose Galois group we would like to calculate. The resolvents used in Stauduhar's algorithm are defined as follows:

Consider the fields $L := \mathbb{Q}(x_1, \ldots, x_n)$ of rational functions and $M := \mathbb{Q}(s_1, \ldots, s_n)$ of elementary symmetric functions in $x_1, \ldots, x_n$ and let $H \leq G \leq S_n$ be permutation groups acting on $\{x_1, \ldots, x_n\}$ by permuting the indices. We denote by L^H the fixed field of L under H. Since L/M is a Galois extension, L^H/L^G is finite and separable. By the theorem of primitive elements, there exists a primitive element $F \in L^H$ with $L^H = L^G(F)$. It is always possible to choose F integral over $\mathbb{Q}[s_1, \ldots, s_n]$. Since the unique factorization domain $\mathbb{Q}[x_1, \ldots, x_n]$ is integrally closed in its quotient field, it follows that F is an element of $\mathbb{Q}[x_1, \ldots, x_n]$. By multiplication with a scalar in $\mathbb{Z}$, F is even an element of $\mathbb{Z}[x_1, \ldots, x_n]$. The primitive element property of F is equivalent to the fact that $\mathrm{Stab}_G(F) = \{\, \sigma \in G \mid \sigma F = F \,\} = H$. The minimal polynomial of F over

L^G is given by $\prod_{\sigma \in G//H}(X - \sigma F)$, where $G//H$ denotes a full system of representatives of left cosets (by left cosets we mean cosets of the form σH). The minimal polynomial is called a *generic relative resolvent*. The following definition and the next theorem will show the importance for the method of Stauduhar of the last two properties.

We introduce the general definition of G- relative H-invariant resolvent polynomials, these are specialized generic relative resolvents.

Definition 2.1. *Let $f \in \mathbb{Z}[x]$ be a polynomial with roots $\alpha_1, \dots, \alpha_n \in \overline{\mathbb{Q}}$ and $H \leq G$ be permutation groups acting on $\{x_1, \dots, x_n\}$. We call $F \in \mathbb{Z}[x_1, \dots, x_n]$ a G-relative H-invariant polynomial if and only if*

1 $\sigma F = F$ for all $\sigma \in H$,
2 $\sigma F \neq F$ for all $\sigma \in G \setminus H$.

In this case

$$R_{G,H,F}(X) := \prod_{\sigma \in G//H} (X - \sigma F(\alpha_1, \dots, \alpha_n))$$

is called a G-relative H-invariant resolvent.

Remark 2.2. For $G = S_n$, we call the G-relative H-invariant resolvent an absolute resolvent.

Theorem 2.3. *Let $f \in \mathbb{Z}[x]$ be a monic, irreducible polynomial of degree n. Moreover, let $H \leq G \leq S_n$ such that $\mathrm{Gal}(f) \leq G$ and let $\sigma \in G$. The polynomial $F \in \mathbb{Z}[x_1, \dots, x_n]$ is assumed to be a G-relative H-invariant polynomial. The roots of f are again denoted by $\alpha_1, \dots, \alpha_n \in \overline{\mathbb{Q}}$. Then:*

1 $R_{(G,H,F)}(X) = \prod_{\sigma \in G//H} (X - \sigma F(\alpha_1, \dots, \alpha_n)) \in \mathbb{Z}[X]$.
2 If $\mathrm{Gal}(f)$ is contained in $\sigma H \sigma^{-1}$, then $(\sigma F)(\alpha_1, \dots, \alpha_n) \in \mathbb{Z}$.
3 If on the other hand $(\sigma F)(\alpha_1, \dots, \alpha_n) \in \mathbb{Z}$ and $(\sigma F)(\alpha_1, \dots, \alpha_n)$ is a simple root of $R_{(G,H,F)}$, then $\mathrm{Gal}(f) \leq \sigma H \sigma^{-1}$. In this case the roots of f can be rearranged according to $\alpha'_j = \alpha_{\sigma(j)}$ such that $\mathrm{Gal}(f) \leq H$.

The main idea of Stauduhar's algorithm is the following: Suppose the Galois group $\mathrm{Gal}(f) \leq G$ with respect to the chosen ordering of the roots of the polynomial f. Initially we know that for $G = S_n$. Using 2 and 3 of Theorem 2.3, we can determine whether $\mathrm{Gal}(f) \leq \sigma H \sigma^{-1}$ for some maximal subgroup H of G and some $\sigma \in G//H$. If $\mathrm{Gal}(f)$ is contained in no maximal subgroup of G, then $\mathrm{Gal}(f) = G$. Otherwise, if $\mathrm{Gal}(f) \leq \sigma H \sigma^{-1}$, we reorder the roots of f according to the permutation σ such that $\mathrm{Gal}(f) \leq H$ and repeat the procedure. Thus, the algorithm traverses the subgroup lattice of transitive permutation groups of degree n from the largest group to the actual Galois group.

Remark 2.4. 1 It is always possible to make the resolvent having no double integral roots by applying a suitable Tschirnhausen transformation to the polynomial f (see (Girstmair, 1983)).

2 We have $\mathrm{Gal}(f) \leq A_n$ if and only if the discriminant of the polynomial f is a rational integral square.
3 If H is a maximal transitive subgroup of G, then for each G-conjugacy class of H we need to consider only one representative.
4 Factorization of the polynomial f into distinct monic irreducible polynomials in $\mathbb{F}_p[x]$ leads to cycle shapes of $\mathrm{Gal}(f)$. For each shape found in this manner, we eliminate all candidate groups which do not exhibit this shape. So it is possible to usually quickly determine if the Galois group of the polynomial f is the symmetric or alternating group by finding shapes unique to these groups and using the discriminant criterion.

According to 2.4.*3* we are left with the case that we have representatives of two conjugacy classes which are maximal in G but which are not G-conjugate to one another. We have computed up to degree 15 that two maximal subgroups of $G \leq S_n$, which are conjugate to one another in S_n are already conjugate to one another in

$$N_{S_n}(G) := \{\, \sigma \in S_n \mid \sigma G \sigma^{-1} = G \,\}$$

the normalizer of G in S_n. Degree 16 is the first degree, where this does not hold any more. For example the group $16T^{+}_{640}$ has two maximal subgroups of transitive group type $16T^{+}_{412}$, which are not conjugate to one another in $N_{S_{16}}(16T^{+}_{640})$. For two maximal subgroups H_1, H_2 of G, lying in the same $N_{S_n}(G)$-conjugacy class, the following holds (see (Eichenlaub and Olivier, 1995)):

THEOREM 2.5. *Let $H_2 = \tau H_1 \tau^{-1}, \tau \in N_{S_n}(G)$ and F be a G-relative H_1-invariant polynomial. Then τF is a G-relative H_2-invariant polynomial and*

$$R_{(G,H_2,\tau F)}(X) = \prod_{\sigma \in G//H_1} (X - \tau\sigma F(\alpha_1, \ldots, \alpha_n))$$

is a G-relative H_2-invariant resolvent. In particular, if $\tau \in G$, then $R_{(G,H_2,F)}(X) = R_{(G,H_1,\tau F)}(X)$.

We will close this section giving, for each degree, an overview of the necessary data which must be computed for this method. Given a list $\mathfrak{T}$ of representatives for the S_n-conjugacy classes of transitive subgroups the following tasks have to be completed for all $G \in \mathfrak{T}$:

1 Find all $T \in \mathfrak{T}$ for which there exists a permutation $\rho \in S_n$ such that $\rho T \rho^{-1}$ is maximal in G. Then we define $\mathfrak{T}_G := \{(T_1, \rho_1), \ldots, (T_k, \rho_k)\}$.
2 For each $T_i \in \mathfrak{T}_G$ let $H_i := \rho_i T_i \rho_i^{-1} \leq G$. Then $\mathfrak{H}(G, H_i) := \{\sigma H_i \sigma^{-1} \mid \sigma \in S_n \text{ and } \sigma H_i \sigma^{-1} \leq G\}$ is the set of subgroups of G of the same transitive group type as H_i.
3 $N_{S_n}(G)$ operates by conjugation on $\mathfrak{H}(G, H_i)$. Compute a G-relative H_i-invariant polynomial $F_{i,j}$ for each orbit $B_{i,j}$ under this action. Since for $n \leq 15$ there is always exactly one orbit, $j = 1$, and we simply write F_i instead of $F_{i,j}$.
4 Compute coset representatives $\sigma_i \in G//H_i$ and $\tau_j \in N_{S_N}(G)//G$. The permutations $\tau_j \sigma_i$ constitute a complete system of representatives for $N_{S_n}(G)//H_i$.

In our current implementation the subgroup lattice, the $\rho_i' s$ and the $\tau_j' s$ are precomputed and stored. The coset representatives $\sigma_i \in G//H_i$ and most of the invariant polynomials are computed during the running time.

2.1. The computation of G-relative H-invariant polynomials

It is well known that G-relative H-invariant polynomials always exist:

Lemma 2.6. *For $H \leq G \leq S_n$ and $\tilde{F}(x_1, \ldots, x_n) = x_1^1 x_2^2 \cdots x_{n-1}^{n-1}$ let*

$$F(x_1, \ldots, x_n) := \sum_{\sigma \in H} \sigma \tilde{F}.$$

Then $\mathrm{Stab}_G(F) = H$.

In practice it is not very efficient to use this polynomial. Our aim is to find an invariant of small total degree. Let $R := \mathbb{Q}[x_1, \ldots, x_n]$. We can decompose

$$R = \bigoplus_{d=0}^{\infty} R_d,$$

where R_d denotes the homogeneous components of degree d. Clearly this gives a decomposition of the invariant ring

$$R^H = \bigoplus_{d=0}^{\infty} R_d^H.$$

R_d is a $\mathbb{Q}$-vector space of dimension $\binom{n+d-1}{n-1}$.

Definition 2.7. *Let $S := R^H$. The Hilbert Series of S is the formal power series*

$$\mathrm{h}(S, t) := \sum_{d=0}^{\infty} \dim_{\mathbb{Q}}(S_d) \cdot t^d \in \mathbb{Z}[[t]].$$

Choosing a G-relative H-invariant polynomial with smallest total degree d among all invariants has major effects on the efficiency of the program: multiplications are very expensive, so we can speed up computations extremely by minimizing the number of multiplications. On the other hand we also gain time during the lifting procedure (see Theorem 2.17) by using an invariant whose resolvent has smaller absolute value roots. Since H is a maximal subgroup of G, d equals the smallest index such that the corresponding coefficients of $\mathrm{h}(R^H, t)$ and $\mathrm{h}(R^G, t)$ are different.

Algorithm 2.8. (Computation of G-relative H-invariant polynomials)

Input: *A permutation group $G \leq S_n$, $(n \geq 4)$ and a maximal transitive subgroup H of G.*

Output: *A homogeneous polynomial F of minimal degree $d \leq \frac{n(n-1)}{2}$ with* $\mathrm{Stab}_G(F) =$ *H.*

Step 1: *Compute the Hilbert series $\mathrm{h}(R^H, t)$ and $\mathrm{h}(R^G, t)$ and compute the smallest index d such that the corresponding coefficients are different.*

Step 2: *Compute all homogeneous invariants of H of total degree d.*

Step 3: *Remove the invariants which are G-relative.*

Step 4: *Return an invariant with the smallest number of monomials.*

For Steps 1 and 2 we use the algorithms implemented in Magma (Kemper and Steel, 1999). Step 2 is the most expensive one of our algorithm. In the sequel we give three lemmata (see Eichenlaub (1996)), which are useful for obtaining computationally better invariant polynomials.

Let us start with a result about wreath products.

LEMMA 2.9. *Suppose $G \leq G' \leq S_\Lambda$ and $H \leq H' \leq S_\Gamma$ are transitive permutation groups acting on $\Lambda := \{1, \ldots, l\}$ resp. $\Gamma := \{1, \ldots, m\}$. Let $y_j := \sum_{\lambda=1}^{l} x_{\lambda,j}$ and $F_j := F(x_{1,j}, \ldots, x_{l,j})$ for $j = 1, \ldots, m$, where F is a G'-relative G-invariant polynomial. Furthermore let E be a H'-relative H-invariant polynomial. Then*

$$F_1 + F_2 + \cdots + F_m + E(y_1, \ldots, y_m)$$

is a $G' \wr_\Gamma H'$-relative $G \wr_\Gamma H$-invariant polynomial.

REMARK 2.10. If we have $G = G'$ in the last lemma, then $E(y_1, \ldots, y_m)$ yields a $G' \wr_\Gamma H'$-relative $G \wr_\Gamma H$-invariant polynomial. Analogously $F_1 + \cdots + F_m$ is sufficient for $H = H'$.

We come to a statement about subgroups of index 2. Essentially we construct new invariants for other subgroups of G of index 2 from known G-relative H-invariant polynomials F with $[G : H] = 2$. Thereby we try to change the known invariant polynomials F, such that the corresponding resolvent is of the form $X^2 - F^2(\alpha_1, \ldots, \alpha_n)$, where the $\alpha_i' s, (1 \leq i \leq n)$ again denote the roots of the polynomial f.

LEMMA 2.11. *Let G be a permutation group with subgroups H_1 and H_2 of index 2. Let $F_i, (i = 1, 2)$ be G-relative H_i-invariant polynomials with $\sigma_i F_i = -F_i$, $(\sigma_i \in G \backslash H_i)$. Then $H_1 + H_2 := (H_1 \cap H_2) \cup ((G \backslash H_1) \cap (G \backslash H_2)) \leq G$ and $F_1 F_2$ is a G-relative $H_1 + H_2$-invariant polynomial.*

REMARK 2.12. The condition $\sigma_i F_i = -F_i$, $(\sigma_i \in G \backslash H_i)$ is no restriction. It can always be obtained by replacing F_i by $F_i' = F_i - \sigma_i F_i$, $\sigma_i \in G \backslash H_i$.

The last lemma deals with wreath products of the form $G = S_l \wr S_m$. We classify subgroups of G by consideration of stabilizers of symmetric polynomials: Define

$$d_k := \prod_{1 \leq i < j \leq l} (x_{i,k} - x_{j,k}), \ \ (1 \leq k \leq m) \ \text{ and } \ D := \prod_{1 \leq i < j \leq m} (y_i - y_j)$$

with $y_j' s$ as in Lemma 2.9 and denote by $s_k, (1 \leq k \leq m)$ the k's elementary symmetric function. Then we have the following

LEMMA 2.13. *The group $S_l \wr_\Gamma S_m$ with $\Gamma := \{1, \ldots, m\}$ has at least three subgroups of index 2: The stabilizers of $s_m(d_1, \ldots, d_m)$, $D(y_1, \ldots, y_m)$ (that is $S_l \wr_\Gamma A_m$), and $D(y_1, \ldots, y_m) s_m(d_1, \ldots, d_m)$. Furthermore $S_l \wr_\Gamma S_m$ has a subgroup of index 2^{m-1} and a subgroup of index 2^m, $(A_l \wr_\Gamma S_m)$, which are the stabilizers of $s_2(d_1, \ldots, d_m)$ resp. $s_1(d_1, \ldots, d_m)$.*

Definition 2.14. *Let G be a transitive permutation group acting on a finite set Ω. A subset $\emptyset \neq \Delta \subset \Omega$ is called a block, if $\Delta \cap \Delta^\sigma \in \{\emptyset, \Delta\}$ for all $\sigma \in G$. The orbit of a block Δ under G is called a block system. A group is called primitive, if it only has blocks of size 1 or $|\Omega|$. Otherwise it is called imprimitive.*

Finally we give an example with combines the three lemmata to show the effect on the performance.

Example 2.15. Consider the group pair $G = 12T_{260}$ and $H = 12T'_{235}$. In this example all $'$-groups result from the groups in Conway *et al.* (1996) by conjugation with $(2, 10, 12, 7)(3, 4, 11, 6, 8)$. Using algorithm 2.8 we obtain an invariant which needs $11 \cdot 1152$ multiplications for this descent. By testing several subgroups of index two, we get $T'_{235} = T'_{241} + T'^{+}_{236}$. Both groups $T'_{241} = S_2 \wr F_{36}(6)$ and $T_{260} = S_2 \wr F_{36}(6) : 2 = S_2 \wr (S_3 \wr S_2)$ are wreath products, that means we can use Theorem 2.9. Remark 2.10 shows, that it is sufficient to find an $S_3 \wr S_2$-relative $F_{36}(6)$-invariant polynomial. Theorem 2.13 gives $\mathrm{Stab}_{S_3 \wr S_2}(Ds_2) = F_{36}(6)$ for $n = 6$. The groups T_{260} and T'_{235} both have a block system $\mathfrak{B} = \{\{1, 7\}, \{2, 8\}, \{3, 9\}, \{4, 10\}, \{5, 11\}, \{6, 12\}\}$ according to the generators used in Conway *et al.* (1996). Thus, we get $y_j = (x_j + x_{j+6})$, $d_j = (x_j - x_{j+6})$, $j = 1, \ldots, 6$ and

$$Ds_2 = \prod_{1 \leq i < j \leq 6} (y_i - y_j) \sum_{1 \leq i < j \leq 6} d_i d_j.$$

Now we are left with the task to construct a T_{260}-relative T'^{+}_{236}-invariant polynomial. Since T'^{+}_{236} is an even permutation group, the polynomial $s_6 = d_1 d_2 d_3 d_4 d_5 d_6$ is stabilized by all permutations from T'^{+}_{236} and permutations from $T_{260} \backslash T'^{+}_{236}$ will change the sign of s_6. Both polynomials, Ds_2 and s_6 satisfy the assumptions of Theorem 2.11. Thus, we obtain as a T_{260}-relative T'_{235}-invariant polynomial

$$Ds_2 s_6,$$

whose evaluation needs less than 40 multiplications.

We have not said anything yet about the decision step of Stauduhar's algorithm. There are several possibilities for performing this step. Stauduhar proposed using high-precision approximations to the roots of f. Since the resolvent has integer coefficients he approximated the roots to sufficient precision so that the resulting error in the absolute value of the coefficient of $R_{G,H,F}(X)$ is less than $\frac{1}{2}$. The required precision using numerical approximations can be very large and therefore leads to bad performances. Another approach is to use p-adic approximations of the roots of the polynomial f as suggested by Yokoyama (1997). We decided to use p-adic approximations, because the advantages are guaranteed results combined with competitive times.

2.2. The p-adic method

In this section we will describe the p-adic decision step in the algorithm of Stauduhar for irreducible monic polynomials $f \in \mathbb{Z}[x]$. Let p denote a prime integer such that f is square-free modulo p. Denote the ring of p-adic integers by $\mathbb{Z}_p$, the field of fractions of $\mathbb{Z}_p$ by $\mathbb{Q}_p$, and an algebraic closure of $\mathbb{Q}_p$ by $\bar{\mathbb{Q}}_p$. In order to compute approximations of the roots $\alpha_1, \ldots, \alpha_n \in \overline{\mathbb{Q}_p}$, we use the following lemma. The proof of it is straightforward. Klüners (1998) describes the p-adic arithmetic in much more detail.

LEMMA 2.16. *Let $l \in \mathbb{Z}$ be minimal such that $f(t) \bmod p$ has n (distinct) roots in $\mathbb{F}_{p^l}$. Let $g(t) \in \mathbb{Z}[t]$ be monic of degree l such that $\mathbb{F}_{p^l}$ is generated by a root of $g(t) \bmod p$ over $\mathbb{F}_p$. Then $g(t)$ is irreducible over $\mathbb{Q}_p$. Furthermore, let $N_p := \mathbb{Q}_p(\omega)$ and $N := \mathbb{Q}(\omega)$ with $g(\omega) = 0$. N_p is the unique unramified extension of $\mathbb{Q}_p$ of degree l and is also the splitting field of $f(t)$ over $\mathbb{Q}_p$. The prime p is inert in $N/\mathbb{Q}$, $p\mathfrak{o}_N = \mathfrak{p}$, and the $\mathfrak{p}$-adic completion of N equals N_p.*

Let $v_\mathfrak{p}$ be the discrete valuation of $N_p/\mathbb{Q}_p$. For all $\beta \in N_p$ and $k \in \mathbb{Z}$ there is an approximation $\beta^{(k)} \in N$ such that $v_\mathfrak{p}\left(\beta - \beta^{(k)}\right) \geq k$ holds. Using Newton lifting we are able to compute approximations $\alpha_1^{(k)}, \ldots, \alpha_n^{(k)} \in N$ of $\alpha_1, \ldots, \alpha_n \in N_p$. For $y \in \mathbb{Z}$ denote by $\lfloor y \rfloor_{p^k}$ the unique representative of $y \bmod p^k$ in $[-(p^k-1)/2, p^k/2]$. We have chosen the symmetric residue system to get small numbers modulo p^k. Denote by $\beta_\sigma \in N_p$ the root of $R_{(G,H,F)}(X)$ belonging to $\sigma \in G//H$.

Darmon and Ford (1989) used the following theorem to verify the Galois groups of polynomials having the Mathieu groups M_{11} and M_{12} as Galois groups.

THEOREM 2.17. *Let $M \in \mathbb{R}$ be an upper bound for the absolute values of the complex roots of $R_{(G,H,F)}(X)$. Let $k \in \mathbb{Z}$ be such that $p^k > (2M)^{[G:H]}$. If $\beta_\sigma \in N_p$ is a root of $R_{(G,H,F)}(X)$ subject to*

1 $\beta_\sigma^{(k)} \in \mathbb{Z}$,

2 $|\lfloor \beta_\sigma^{(k)} \rfloor_{p^k}| < M$,

3 $\beta_\sigma^{(k)} \not\equiv \beta_{\tilde\sigma}^{(k)} \bmod \mathfrak{p}^k$ for all $\tilde\sigma \in G//H$ with $\tilde\sigma \neq \sigma$.

Then $\beta_\sigma = \lfloor \beta_\sigma^{(k)} \rfloor_{p^k} \in \mathbb{Z}$ is a simple root of $R_{(G,H,F)}(X)$.

PROOF. 1 β_σ is a root of $R_{G,H,F}(X)$. Thus,

$$\begin{aligned} R_{G,H,F}(\beta_\sigma^{(k)}) &\equiv R_{G,H,F}(\beta_\sigma) && mod\ \mathfrak{p}^k \\ \Leftrightarrow\quad R_{G,H,F}(\beta_\sigma^{(k)}) &\equiv 0 && mod\ \mathfrak{p}^k. \end{aligned}$$

Since the left side is an element in $\mathbb{Z}$ and $\mathfrak{p} = p\,o_N$ it follows:

$$\Leftrightarrow\quad R_{G,H,F}(\beta_\sigma^{(k)}) \equiv 0 \qquad mod\ p^k.$$

2 Because $|\lfloor \beta_\sigma^{(k)} \rfloor_{p^k}| < M$, we may assume without loss of generality that $|\beta_\sigma^{(k)}| < M$. From $|\sigma F(\alpha_1, \ldots, \alpha_n)| < M$ (for complex α_i) it follows that

$$\begin{aligned} |R_{G,H,F}(\beta_\sigma^{(k)})| &= \prod_{\sigma \in G//H} |\beta_\sigma^k - \sigma F(\alpha_1, \ldots, \alpha_n)| \\ &\leq \prod_{\sigma \in G//H} (2M) \\ &\leq (2M)^{[G:H]}. \end{aligned}$$

Since $p^k \mid R_{G,H,F}(\beta_\sigma^{(k)})$ and $p^k > (2M)^{[G:H]}$ we have $R_{G,H,F}(\beta_\sigma^{(k)}) = 0$. Thus, $\beta_\sigma^{(k)} = \beta_\sigma$. From assumption (iii) we get that β_σ is a simple root of $R_{G,H,F}(X)$. □

Some remarks are in order here.

REMARK 2.18. In our implementation we first lift the approximations up to the heuristic bound $p^{k'}$ with $k' = min\,\{\,3\,log_p(2M), [G:H]\,log_p(2M)\,\}$. Approximations $\beta_\sigma^{(k')}\,mod\,p \notin \mathbb{F}_p$ cannot correspond to an integer root if $l > 1$, since this implies that $\beta_\sigma \notin \mathbb{Q}_p$. In a second loop we lift the remaining roots up to the bound k. If the absolute value of the representative of $\beta_\sigma^{(j)}\,mod\,p^j$ is bigger than M for $j \geq k$, than either $\beta_\sigma^{(j)}$ is not an element of $\mathbb{Z}$ or $|\,\lfloor \beta_\sigma^{(j)} \rfloor_{p^k}\,| > M$. Therefore β_σ can also be removed from the candidate list.

2.3. Main problems

The main problem of the relative resolvent method is that for growing n the first descent from S_n resp. A_n becomes very large. For example, in degrees $n = 13, 14$ and 15 we have the following indices of maximal transitive subgroups in S_n and A_n:

$$
\begin{array}{ll}
[S_{13} : 13T_6] = 39916800 & [A_{13} : 13T_7^+] = 554400 \\
 & [A_{13} : 13T_5^+] = 39916800 \\
 & \\
[S_{14} : 14T_{61}] = 1716 & [A_{14} : 14T_{59}^+] = 3432 \\
[S_{14} : 14T_{57}] = 135135 & [A_{14} : 14T_{55}^+] = 270270 \\
[S_{14} : 14T_{39}] = 39916800 & [A_{14} : 14T_{30}^+] = 39916800 \\
 & \\
[S_{15} : 15T_{102}] = 126126 & [A_{15} : 15T_{99}^+] = 126126 \\
[S_{15} : 15T_{93}] = 1401400 & [A_{15} : 15T_{89}^+] = 1401400 \\
 & [A_{15} : 15T_{72}^+] = 32432400
\end{array}
$$

These indices increase exponentially in n, e.g. for n even we have

$$[S_n : (S_{\frac{n}{2}} \wr S_2)] = \frac{n!}{2(\frac{n}{2})!(\frac{n}{2})!} \text{ and } [S_n : (S_2 \wr S_{\frac{n}{2}})] = \frac{n!}{2^{\frac{n}{2}}(\frac{n}{2})!}.$$

For p prime we have $\mathrm{PSL}_2(\mathrm{p}) \leq A_{p+1}$, where $[A_{p+1} : \mathrm{PSL}_2(p)] = (p-2)!$. For $p \neq 2, 3, 11, 23$ we get that $\mathrm{PSL}_2(\mathrm{p})$ is a maximal subgroup of A_{p+1}.

One problem which occurs is that the coset computation takes a lot of time, and the inclusion test, too. Another problem is the verification of the result. To verify the Galois group we must lift the approximations to a bound k such that

$$p^k > (2\,M)^{[G:H]}.$$

And there the index comes in. Both points seem to be extremely time consuming for large degrees n, thus our goal is to give improvements especially on these two points.

3. Extension of the relative resolvent method using subfields

In this section we develop an extension of the relative resolvent method. Previous investigations have shown that the first descent from S_n resp. A_n, is particularly time consuming. Thus it would be desirable to skip this first step by means of computing suitable additional information. Using this information, we would like to change the starting point of the algorithm in the subgroup lattice, to get as close as possible to the actual Galois group. In order for the method to work, we must be guaranteed that the Galois group $\mathrm{Gal}(f) \leq G$ chosen as the starting point. That means the Galois group considered as a permutation group must be a subgroup of G with respect to the chosen ordering of the roots of f. Such an extension can be realized for imprimitive transitive permutation groups. By Krasner's and Kaloujnine's theorem (see (Meldrum, 1995)) a transitive, imprimitive permutation group with a block system, which consists of m blocks of length l, can be embedded in a wreath product of the form $S_l \wr S_m$. If the imprimitive permutation group has different block systems, then it lies in the intersection of these wreath products.

How do we arrive at this information for a given polynomial f? Let α be a root of f. In the computer algebra system KANT there is a fast algorithm for computing subfields of algebraic number fields $\mathbb{Q}(\alpha)$ (Klüners and Pohst, 1997; Klüners, 1998). The subfields of $\mathbb{Q}(\alpha)$ of degree m are in bijection with the blocks B of length $l := \frac{n}{m}$ of $\mathrm{Gal}(f)$ which contain α. Each subfield can be represented by a pair of polynomials $(g, h) \in \mathbb{Z}[x] \times \mathbb{Q}[x]$, where g is the minimal polynomial of a primitive element β of a subfield and $h(\alpha) = \beta$. We call h the embedding polynomial. To specialize this fact with respect to the application we have in mind, we use the following

THEOREM 3.1. *Let $E_1 = \mathbb{Q}(\beta)$, $E_2 = \mathbb{Q}(\alpha)$ be algebraic number fields with $\mathbb{Q} \subseteq E_1 \subseteq E_2$ and $g, f \in \mathbb{Z}[x]$ be the minimal polynomials of β and α, respectively. Let $h \in \mathbb{Q}[x]$ be the embedding polynomial with $h(\alpha) = \beta$. Denote the conjugates of α and β in some algebraic closure with $\alpha_1, \ldots, \alpha_n$ and $\beta_1, \ldots, \beta_m$, respectively. Defining $B_i = \{\alpha_j | h(\alpha_j) = \beta_i\}$ it follows:*

1 $B_1, \ldots, B_m$ form a block system of $\mathrm{Gal}(f)$. Furthermore, $n = |B_i| m$.
2 $\mathrm{Gal}(g)$ is isomorphic to the permutation representation of $\mathrm{Gal}(f)$ with respect to $B_1, \ldots, B_m$ under the mapping $\theta : \beta_i \mapsto B_i$.

PROOF. *(1)* Let $\sigma \in G$ and $\gamma \in B_i$ with $\sigma(\beta_i) = \beta_k$. Then the following equivalences hold:

$$\begin{aligned} \gamma \in B_i &\Leftrightarrow h(\gamma) = \beta_i \\ &\Leftrightarrow \sigma(h(\gamma)) = h(\sigma(\gamma)) = \beta_k \\ &\Leftrightarrow \sigma(\gamma) \in B_k. \end{aligned}$$

From the above equivalence and the transitivity of G it follows $n = \mid B_i \mid m$ for $1 \le i \le m$.

(2) $\mathrm{Gal}(g)$ is equivalent to the permutation representation of G according to the B_i under the mapping $\theta : \beta_i \longmapsto B_i$ because $\mathbb{Q}(\beta_i) = \mathbb{Q}(\alpha_1, \ldots, \alpha_n)^{\mathrm{Stab}_{\mathrm{Gal}(f)}(B_i)}$. □

Because of Theorem 3.1 *2* one knows, that the operation of the Galois group of f on the blocks B_i of length $l, 1 \le i \le m$, is equivalent to the operation of the Galois group of the minimal polynomial of the subfield on their roots. It follows that one can embed the Galois group in $S_l \wr \mathrm{Gal}(g)$.

ALGORITHM 3.2. (Galois group computation using subfields)

Input: *Monic, irreducible polynomial f of degree n with rational integer coefficients, roots $\alpha_1, \ldots, \alpha_n$ given in some p-adic completion.*

Output: *Permutation group $T \in \mathfrak{T}$ and root ordering such that $\mathrm{Gal}(f) \le T$.*

Step 1: *(Initialization) Compute roots of f and choose an arbitrary root ordering.*

Step 2: *(Discriminant?) If $\mathrm{disc}(f)$ is a square in $\mathbb{Z}$, then $G \leftarrow A_n$, else $G \leftarrow S_n$.*

Step 3: *(Subfields) Compute minimal polynomials $g_1, \ldots, g_s$ of all subfields of $\mathbb{Q}(\alpha)$, (α a root of f), and embedding polynomials $h_1, \ldots, h_s$ by using the subfield algorithm.*

Step 4: *(Primitivity?) If $s = 0$, then $\mathrm{Gal}(f)$ is a primitive permutation group. Output of $T \leftarrow G$ and root ordering $\alpha_1, \ldots, \alpha_n$ and terminate. Otherwise set $i \leftarrow 1$.*

Step 5: *(Roots in blocks) Set $m_i \leftarrow deg(g_i)$ and $l_i \leftarrow n/m_i$. The Galois group has a block system $\mathfrak{B}_i = \{B_1, \ldots, B_{m_i}\}$ with blocks of length l_i. Compute the root partitioning of f with respect to the blocks $B_1, \ldots, B_{m_i}$ using the embedding polynomial h_i (Theorem 3.1).*

Step 6: *(Wreath product) Let $K_i = S_{l_i} \wr S_{m_i}$ and determine the permutation $\sigma \in S_n$ which maps the block system of K_i onto the block system $\mathfrak{B}_i$.*

Step 7: *(Conjugate wreath product) Set $K_i \leftarrow \sigma K_i \sigma^{-1}$. Now $\mathrm{Gal}(f) \leq K_i$.*

Step 8: *(Next g_i?) If $i < s$, then $i \leftarrow i + 1$ and repeat from step 5.*

Step 9: *(Intersection) Set $G \leftarrow G \cap (\bigcap_{i=1}^{s} K_i)$.*

Step 10: *(Identification) Identify G with $T \in \mathfrak{T}$ and determine permutation σ such that $G = \sigma T \sigma^{-1}$.*

Step 11: *(Adjust root ordering) Set $\alpha_i \leftarrow \alpha_{\sigma(i)}$. Now $\mathrm{Gal}(f) \leq T$. Output of T and root ordering $\alpha_1, \ldots, \alpha_n$.*

Remark 3.3. 1 If we compute the Galois group $\mathrm{Gal}(g_i)$ acting on $\beta_1, \ldots, \beta_{m_i}$ in step 5 of the above algorithm, we can use the isomorphism θ of Theorem 3.1 to improve the above algorithm. After reordering the B_i according to θ we can use $K_i = S_{l_i} \wr \mathrm{Gal}(g_i)$ in step 6. The group T may become smaller, but we need some computing time to compute $\mathrm{Gal}(g_i)$.

2 A similar improvement can be done if we are able to compute the relative Galois group G of m_α over $\mathbb{Q}(\beta)$, where m_α denotes the minimal polynomial of α over $\mathbb{Q}(\beta)$. In this case we can use $K_i = G \wr S_{m_i}$.

4. Short coset systems

The previous section gave an improvement of Stauduhar's method for imprimitive groups. The primitive groups remain. In the sequel we give independent solutions for the problems of large coset representative systems and high lifting bounds. In general, these methods apply to both, imprimitive and primitive groups. For large degrees (≥ 11) the best results are obtained by combining the techniques of section 4 and section 5.

Let us start by introducing short coset systems. Let $f \in \mathbb{Z}[x]$ be monic and irreducible, $\alpha_1, \ldots, \alpha_n \in \bar{\mathbb{Q}}$ be the roots of f and set $E := \mathbb{Q}(\alpha_1, \ldots, \alpha_n)$. We look at $\mathrm{Gal}(f)$ as a permutation group on the roots of f and assume that we know a group $G \leq S_n$ such that $\mathrm{Gal}(f) \leq G$ holds. For a maximal transitive subgroup H of G the method of Stauduhar needs to check whether $\mathrm{Gal}(f) \leq \sigma H \sigma^{-1}$ for some $\sigma \in G//H$.

Improvement: If we additionally know a permutation group $K \leq \mathrm{Gal}(f)$, we can restrict to those $\sigma \in G//H$ with $K \leq \sigma H \sigma^{-1}$.

Definition 4.1. *Let $H \leq G \leq S_n$ and K be a subgroup of the Galois group of f, viewed as a permutation group with respect to the chosen ordering of the roots of f. Then we call the set*

$$(G/H)_K := \{\ \sigma H \in G/H \mid K \leq \sigma H \sigma^{-1}\ \}$$

short cosets. We denote by $(G//H)_K$ a full system of representatives of $(G/H)_K$.

Explicit permutation subgroups $K \leq \mathrm{Gal}(f)$ can be obtained as follows:

Complex case: For $\alpha_1, \ldots, \alpha_n \in \mathbb{C}$ we may take the cyclic subgroup K generated by the complex conjugation. Complex conjugation is an automorphism of any subfield of the complex numbers and induces an element in $\mathrm{Gal}(f)$ of cycle type $(2^{r_2}, 1^{r_1})$, where r_1 denotes the number of real zeros and r_2 is the number of complex conjugate pairs of roots of f.

p-adic case: For $\alpha_1, \ldots, \alpha_n \in \bar{\mathbb{Q}}_p$ we may take the cyclic subgroup K generated by the Frobenius automorphism. Assuming $p \nmid \mathrm{disc}(f)$, all α_i are different modulo p. Therefore the Frobenius automorphism τ can be computed using the congruence $\tau(\alpha_i) \equiv \alpha_i^p \bmod p$. The Frobenius automorphism is an element of cycle type $(\deg(f_1), \ldots, \deg(f_r))$, where $f \equiv f_1 \cdots f_r \bmod p$ is the factorization of f modulo p.

Even if the group K is of small order, this shortens the set of coset representatives extremely as the following example shows:

Example 4.2. Let H be the group $\mathrm{PSL}_2(p)$ which is maximal in $G := A_{p+1}$ for $p \neq 2, 3, 11, 23$. It has index $[G : H] = (p-2)!$. Let K be generated by an element of order p. Then we get $|(G//H)_K| = 1$.

Here we see another advantage of the p-adic computation. If we have chosen a prime number p for which we cannot reduce the coset system, we are able to take another prime number. In the complex case there is no such possibility for totally real polynomials.

Theorem 4.3. *Let $f \in \mathbb{Z}[x]$ be an irreducible monic polynomial and denote by E the splitting field of f over $\mathbb{Q}$. Let $\mathrm{Gal}(f) \leq G$ be a permutation groups acting on $\{\alpha_1, \ldots, \alpha_n\}$ and H be a maximal subgroup of G. Furthermore let $F(x_1, \ldots, x_n)$ be a G-relative H-invariant polynomial. If $|(G//H)_K| \geq 2$ and if the shortened resolvent*

$$\prod_{\sigma \in (G//H)_K} (X - \sigma F(\alpha_1, \ldots, \alpha_n)) \in E[X]$$

has a simple root $a \in \mathbb{Z}$, then we must have $\mathrm{Gal}(f) \lneqq G$.

Proof. Supposing $\mathrm{Gal}(f) = G$ we get that $\gamma := F(\alpha_1, \ldots, \alpha_n)$ is an element of E^H since $\mathrm{Stab}_G(F) = \{\sigma \in G \mid \sigma F = F\} = H$. Therefore we have for the characteristic polynomial $\mu_\gamma(X)$ of γ in $E^H/\mathbb{Q}$:

$$\begin{aligned} \mu_\gamma(X) &= \prod_{\sigma \in G//H} (X - \sigma F(\alpha_1, \ldots, \alpha_n)) \\ &= R_{(G,H,F)}(X). \end{aligned}$$

On the other hand we have

$$\mu_\gamma(X) = (m_\gamma(X))^k \text{ for some } k \in \mathbb{N},$$

where $m_\gamma(X)$ denotes the minimal polynomial of γ over $\mathbb{Q}$. Since $(X-a) \mid \mu_\gamma(X) = (m_\gamma(X))^k$ in $\mathbb{Z}[X]$ it follows that $\mu_\gamma(X) = (X-a)^{[G:H]}$ which is a contradiction to the fact that there is a root $b \neq a$ of $R_{(G,H,F)}(X)$. Thus $\mathrm{Gal}(f) \lneqq G$. $\Box$

REMARK 4.4. 1 In Theorem 4.3 it is enough to consider $\sigma_1, \sigma_2 \in (G//H)$ with $\sigma_1 \neq \sigma_2$ and $\sigma_1 F(\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}$ and $\sigma_1 F(\alpha_1, \ldots, \alpha_n) \neq \sigma_2 F(\alpha_1, \ldots, \alpha_n)$.
2 In the situation of Theorem 4.3 it does not follow that $\mathrm{Gal}(f) \leq H$.

APPLICATION 4.5. *Consider all maximal subgroups of the group G with short coset systems. If there is only one possible descent left, this descent is proven. Especially for primitive groups of degree $11 \leq n \leq 15$ in the most cases there is only one group which is maximal in S_n resp. A_n.*

In the following we assume that $K = \langle \tau \rangle \leq \mathrm{Gal}(f)$. A straight forward, but quite impracticable and time consuming method to compute a short coset system would be to first compute all coset representatives $\sigma \in G//H$ and then filter out the ones for which $\tau \in \sigma H \sigma^{-1}$ hold. We are looking for other possibilities to make the program more efficient. The next algorithm is a big improvement to the straight forward method for large indices. For this we have to use some basic group theory. For a permutation group G and a permutation τ denote by $C_G(\tau) := \{\, \sigma \in G \mid \sigma\tau = \tau\sigma \,\}$ the centralizer of τ in G.

ALGORITHM 4.6. (Computation of a short coset system)

Input: $K \leq H \leq G \leq S_n$ *with* $K = \langle \tau \rangle$.

Output: $(G//H)_K$.

Step 1: *Compute the set $\mathcal{C}$ of H-conjugacy classes of H which have the same cycle type as τ.*

Step 2: *For each $C \in \mathcal{C}$ compute a $\sigma \in G$ such that $\sigma^{-1}\tau\sigma \in C$, if σ exists. The set of these σ is denoted by $\mathcal{G}$.*

Step 3: *For each $\sigma \in \mathcal{G}$ compute the set $A_\sigma := (C_G(\tau)//C_{\sigma H \sigma^{-1}}(\tau))$.*

Step 4: *Output of $\{\, a\sigma \mid \sigma \in \mathcal{G},\ a \in A_\sigma \,\} = (G//H)_K$.*

PROOF. Correctness of the algorithm:

1 For $\sigma \in G$ we have $\langle \tau \rangle \leq \sigma H \sigma^{-1}$ is equivalent to $\sigma^{-1}\tau\sigma \in H$. Therefore $\sigma^{-1}\tau\sigma \in H$ lies in one $C \in \mathcal{C}$.
2 Let $\sigma \in \mathcal{G}$ with $\sigma^{-1}\tau\sigma \in C$. For $\tilde\sigma \in G$ it follows that
$$\tilde\sigma^{-1}\tau\tilde\sigma \in C \Longleftrightarrow \text{ it exists } \rho \in H : \tilde\sigma^{-1}\tau\tilde\sigma = \rho^{-1}\sigma^{-1}\tau\sigma\rho \Longleftrightarrow \tilde\sigma \in C_G(\tau)\sigma H.$$
Then $\{\, \sigma \in G \mid \sigma^{-1}\tau\sigma \in H \,\} = \dot\bigcup_{\sigma \in \mathcal{G}} C_G(\tau)\sigma H$ with $\mathcal{G}$ such as in Algorithm 4.6.
3 Since $C_G(\tau) = \dot\bigcup_{a \in A_\sigma} a C_{\sigma H \sigma^{-1}}(\tau)$ for every $\sigma \in \mathcal{G}$ and $C_{\sigma H \sigma^{-1}}(\tau)\sigma H = \sigma H$ we obtain $\dot\bigcup_{\sigma \in \mathcal{G}} C_G(\tau)\sigma H = \dot\bigcup_{\sigma \in \mathcal{G}} (\dot\bigcup_{a \in A_\sigma} a\sigma H)$. The last union is disjoint, because:
$$\begin{aligned} a_1\sigma H = a_2\sigma H &\Longleftrightarrow a_1 a_2^{-1} \in C_G(\tau) \cap \sigma H \sigma^{-1} \\ &\Longleftrightarrow a_1 a_2^{-1} \in C_{\sigma H \sigma^{-1}}(\tau) \end{aligned}$$

which is not possible according to the choice of A_σ. $\square$

In this section we have solved one of the two main problems, namely that the number of cosets is too large. In Remark 2.18 we explained that it may happen that we can detect cosets which do not correspond to integral roots of the resolvent using a small p-adic precision. The practice shows that in most cases we are left with at most one coset which may correspond to an integral solution of the resolvent. If $[G : H]$ is large the remaining problem is to prove that this coset indeed corresponds to an integral solution. Suppose that we have the additional information that $\mathrm{Gal}(f) \leq \sigma H \sigma^{-1}$ for some σ. For instance this can be the case when the polynomial was constructed in a special way. Then we know that the last remaining coset must correspond to an integral solution of the resolvent and we do not need to apply the method of the next section.

5. Verification of Stauduhar steps with large index

Up to now, we have solved the problem of large coset representative systems by means of introducing short coset systems. In order to obtain verifiable results we have to lift the p-adic approximations of the roots of up to a bound k, which strongly depends on the index $[G : H]$. For running time reasons it would be desirable to avoid the lifting procedure for the $G : H$ step. Roughly speaking, this can be done in the following way: First, compute the Galois group with the method of Stauduhar using short coset systems and a lower lifting bound for the first descent. This yields an unproven result. Secondly, verify the Galois group by using absolute resolvent methods.

The absolute resolvent method uses mainly resolvents associated to intransitive permutation groups of the form $H = S_r \times S_{n-r}$, $(1 < r < n)$. For this kind of groups there exist very simple S_n-relative $S_r \times S_{n-r}$-invariant polynomials F. For instance, one can choose

$$F(x_1, \ldots, x_n) = x_1\, x_2 \cdot \ldots \cdot x_r \quad \text{or} \quad F(x_1, \ldots, x_n) = x_1 + x_2 + \ldots + x_r.$$

Therefore absolute resolvents corresponding to groups of the form above are often called r-set resolvents. These r-set resolvents are easy to compute, because for the computation over fields of characteristic zero only the coefficients of the polynomial f are needed (see Casperson and McKay (1994)). Provided that the absolute resolvent is square-free, it is well known (see Soicher (1981), Soicher and McKay (1985)) that the degrees of the irreducible factors of the resolvent in $\mathbb{Z}[x]$ correspond to the lengths of the $\mathrm{Gal}(f)$-orbits of $S_n//H$. For each possible Galois group $\mathrm{Gal}(f)$ and each group H the degrees of the irreducible factors can be tabulated in advance. Such a table is called a partition table. For small degrees the Galois group can be identified by comparing the irreducible factors of the absolute resolvent belonging to the group H with the partition table. For higher degrees n not all possible Galois groups can be distinguished using r-set resolvents and unfortunately, these resolvents are particularly hard to factor.

Since the method of Stauduhar also provides the action of the group on the roots, we can work in reverse: instead of factoring the r-set resolvent, we can write down the factors and then test if the factors divide the r-set resolvent. In our current implementation, we use this method for degrees $n > 9$. Instead of taking k as in Theorem 2.17, we have chosen a heuristic bound for the first step to be $k' = \min\{10 \log_p(2M), [G : H] \log_p(2M)\}$. In the sequel we describe the verification step.

ALGORITHM 5.1. (Verification of Stauduhar steps with large index.)

Input: *A monic irreducible polynomial* $f \in \mathbb{Z}[x]$, $H \leq \mathrm{Gal}(f) \leq G$ *as permutation groups on the roots* $\alpha_1, \ldots, \alpha_n \in \bar{\mathbb{Q}}_p$ *of* f, $r \in \mathbb{N}$ *such that the orbits of the* r*-sets under* H *and* G *are different.*

Output: $H \neq \mathrm{Gal}(f)$ *or* $G \neq \mathrm{Gal}(f)$.

Step 1: $S := \{A \subseteq \{\alpha_1, \ldots, \alpha_n\} \mid |A| = r\}$.

Step 2: *Compute an* H*-orbit* O *of* S *which is not a* G*-orbit.*

Step 3: *Compute the* r*-set resolvent polynomial* $F \in \mathbb{Z}[x]$.

Step 4:

$$f_1 := \sum_{A \in O} \prod_{\alpha \in A} \alpha \bmod p.$$

Step 5: *Compute* $f_2 \in \mathbb{Z}[x]$ *such that* $f \equiv f_1 f_2 \bmod p$.

Step 6: *Check if* f_1 *and* f_2 *are coprime modulo* p. *If not, compute a suitable Tschirnhausen transformation for* f *and go to Step 3.*

Step 7: *Compute a bound* M *for the size of the coefficients of the factors of* F *and* $k \in \mathbb{N}$ *such that* $p^k > 2M$.

Step 8: *Lift* $F \equiv f_1 f_2 \bmod p$ *to* $F \equiv F_1 F_2 \bmod p^k$.

Step 9: *Check, if* F_1 *correspond to a true factor of* F. *In this case return that* $\mathrm{Gal}(f) \neq G$. *Otherwise return that* $\mathrm{Gal}(f) \neq H$.

In Step 7 of the above algorithm we use well known bounds of factorization algorithms (see e.g. (von zur Gathen and Gerhard, 1999)). For the transformations in Step 6 we choose random $\lambda_1, \ldots, \lambda_n \in \mathbb{Z}$ in such a way that $\sum_{j=1}^{n} \lambda_j \alpha_i$ is a primitive element and replace α_i by $\sum_{j=1}^{n} \lambda_j \alpha_i$, $1 \leq i \leq n$ (see also (Girstmair, 1983)).

EXAMPLE 5.2.

1 Let $H = 12T_{295}^{+} = M_{12}$ and $G = 12T_{300}^{+} = A_{12}$. Looking at the following table we have to take $r = 6$ to distinguish H and G. In this case H is a maximal subgroup of G. Therefore the output of the algorithm that $\mathrm{Gal}(f) \neq G$ implies $\mathrm{Gal}(f) = H$.

2 Let $H = 15T_{20}^{+}$ and $G = 15T_{103}^{+}$. From the following table we get that $r = 2$ suffices to distinguish H and G. In this case H is not a maximal subgroup. We have the following situation: $15T_{20}^{+} < 15T_{28}^{+} < 15T_{47}^{+} < 15T_{72}^{+} < 15T_{103}^{+}$. The only unproven step in the algorithm is the step from $15T_{103}^{+}$ to $15T_{72}^{+}$. The other steps are proved using Stauduhar's method provided the first step was correct. If the algorithm outputs that $\mathrm{Gal}(f) \neq G = 15T_{103}^{+}$ this proves that $H = \mathrm{Gal}(f)$. If we only use the absolute resolvent method we have to use $r = 4$ to distinguish $15T_{20}^{+}$ and $15T_{28}^{+}$.

In the following we give a partition table for the primitive groups of degree 12 to 15 used for the verification step. For the transitive groups of degree 9 to 11 tables can be found for instance in Eichenlaub (1996). In the following table $110^3, 132^2, 330$ means that there are three factors of degree 110, two factors of degree 132, and one factor of degree 330.

Degree 12

Gal(f)	2-set	3-set	4-set	5-set	6-set
$12T_{301}$	66	220	495	792	924
$12T_{300}^{+}$	66	220	495	792	924
$12T_{295}^{+}$	66	220	495	792	$132, 792$
$12T_{272}^{+}$	66	220	$165, 330$	$132, 660$	$22, 110, 792$
$12T_{218}$	66	220	$165, 330$	$132, 660$	$110, 220, 264, 330$
$12T_{179}^{+}$	66	220	$165, 330$	$132, 660$	$110^3, 132^2, 330$

Degree 13

Gal(f)	2-set	3-set	4-set	5-set	6-set
$13T_9$	78	286	715	1287	1716
$13T_8^{+}$	78	286	715	1287	1716
$13T_7^{+}$	78	$52, 234$	$13, 234, 468$	$117, 468, 702$	$78, 234, 468, 936$
$13T_6$	78	$52, 78, 156$	$39, 52, 78^2, 156^3$	$39, 78^2, 156^7$	$26, 52, 78^3, 156^9$
$13T_5^{+}$	39^2	$26^2, 39^2, 78^2$	$26^2, 39^5, 78^6$	$39^5, 78^{14}$	$13^2, 26^2, 39^6, 78^{18}$
$13T_4$	26^3	$26^3, 52^4$	$13^3, 26^6, 52^{10}$	$13^3, 26^6, 52^{21}$	$26^{10}, 52^{28}$
$13T_3^{+}$	39^2	$13^4, 39^6$	$13^4, 39^{17}$	39^{33}	$13^6, 39^{42}$
$13T_2^{+}$	13^6	$13^6, 26^8$	$13^{15}, 26^{20}$	$13^{15}, 26^{42}$	$13^{20}, 26^{56}$
$13T_1^{+}$	13^6	13^{22}	13^{55}	13^{99}	13^{132}

Degree 14

Gal(f)	2-set	3-set	4-set	5-set	6-set	7-set
$14T_{63}$	91	364	1001	2002	3003	3432
$14T_{62}^{+}$	91	364	1001	2002	3003	3432
$14T_{39}$	91	364	$182, 273, 546$	$364, 546, 1092$	$91, 182, 546, 1092^2$	$156, 364, 728, 1092^2$
$14T_{30}^{+}$	91	182^2	$91^2, 273, 546$	$182^2, 546^3$	$91^3, 546^3, 1092$	$78^2, 182^2, 364^2, 546^4$

Degree 15

Gal(f)	2-set	3-set	4-set	5-set	6-set	7-set
$15T_{104}$	105	455	1365	3003	5005	6435
$15T_{103}^{+}$	105	455	1365	3003	5005	6435
$15T_{72}^{+}$	105	$35, 420$	$105, 420, 840$	$168, 315, 840, 1680$	$105, 280, 420, 1680, 2520$	$15, 120, 420, 840, 2520^2$
$15T_{47}^{+}$	105	$35, 420$	$105, 210, 420, 630$	$42, 126, 315, 420, 840, 1260$	$70, 105, 210, 420^2, 1260, 2520$	$15, 120, 420, 630^2, 840, 1260^3$
$15T_{28}^{+}$	$45, 60$	$15, 20, 60, 180^2$	$30, 45, 60^2, 90, 180^2, 360^2$	$6, 45, 60, 72, 90^2, 120, 180^2, 360^6$	$10, 15, 60^3, 90^2, 120, 180^3, 360^9, 720$	$15, 60, 90^2, 120^2, 180^9, 360^6, 720^3$
$15T_{20}^{+}$	$45, 60$	$15, 20, 60, 180^2$	$30, 45, 60^2, 90, 180^4, 360$	$6, 36^2, 45, 60, 90^2, 120, 180^6, 360^4$	$10, 15, 60^5, 90^2, 180^7, 360^9$	$15, 60, 90^2, 120^2, 180^{15}, 360^9$

6. The entire algorithm

In this section we give a brief survey about the whole algorithm. One critical point is the prime p chosen for the p-adic completion. Let $f \in \mathbb{Z}[x]$ be a monic polynomial and p be a prime not dividing $\mathrm{disc}(f)$. Factorize $f \equiv f_1 \cdots f_r \bmod p$ and define $d_p := \mathrm{lcm}(\deg(f_1), \ldots, \deg(f_r))$. Let $\mathfrak{T}_{A_n}$ be the set of all transitive subgroups of A_n up to conjugation in S_n. Analogously, let $\mathfrak{T}_{S_n}$ be the set of all transitive subgroups of S_n not contained in A_n up to conjugation in S_n. When we say that a group is contained in such a set we mean that there is a group in the set which is conjugated (in S_n) to our given group. When we have fixed a prime $p \nmid \mathrm{disc}(f)$, we have no multiple roots modulo p. Therefore it is sufficient to compute the roots in the p-adic completion modulo p to distinguish them. When we need more precision Newton lifting can be used to lift the roots to the desired precision.

Algorithm 6.1. (Computation of Galois groups)

Input: *Monic, irreducible polynomial f of degree n with rational integer coefficients.*

Output: *The Galois group of f including the action on the roots.*

Step 1: *(Discriminant?) If $\mathrm{disc}(f)$ is a square in $\mathbb{Z}$ set $T \leftarrow \mathfrak{T}_{A_n}$. Otherwise set $T \leftarrow \mathfrak{T}_{S_n}$ (Remark 2.4).*

Step 2: *(Factorization mod p) Factorize f modulo some primes $p \nmid \mathrm{disc}(f)$ (Remark 2.4). Remove all groups from T which do not contain an element of the given cycle shape.*

Step 3: *(Galois group found?) If* $|T| = 1$ *then return* $\mathrm{Gal}(f)$ *and an arbitrary ordering of the roots of* f.

Step 4: *(Subfields) Compute the subfields of the stem field* K *of* f.

Step 5: *If there are non-trivial subfields then go to Step 5.1, else go to Step 5.2.*

Step 5.1 *(Galois group imprimitive) Remove all groups from* T *which do not have block systems of the computed shape. Choose a prime* p *such that* d_p *is small. Compute the roots* $\alpha_1, \ldots, \alpha_n \bmod p$. *Apply Algorithm 3.2 to compute* G *such that* $\mathrm{Gal}(f) \leq G$.

Step 5.2 *(Galois group primitive) Remove all imprimitive groups from* T. *Suppose that* $\mathrm{Gal}(f)$ *is the smallest group contained in* T *and find out, if there is a step* $H < G$ *with a huge group index. In this case compute the* r*-set polynomial* F *needed for the proof of the critical step (Algorithm 5.1). Choose a prime* p *with the following properties:*

1 $F \bmod p$ *is square-free.*
2 d_p *is small.*
3 $[C_G(\tau) : C_H(\tau)]$ *is small, where* τ *is the corresponding Frobenius automorphism.*

Compute the roots $\alpha_1, \ldots, \alpha_n \bmod p$ *and set* $G \leftarrow S_n$ *or* $G \leftarrow A_n$ *depending on Step 1.*

Step 6: *(Traverse subgroup lattice) For all maximal subgroups* H *of* G *contained in* T *apply the* p*-adic version of Stauduhar's algorithm (Section 2.2). If* $[G : H] >$ 2000 *use an unproven precision (say* $k = 10 \log_p(2M)$*, compare Theorem 2.17). If* $\mathrm{Gal}(f) \leq H$ *then set* $G \leftarrow H$ *and go to Step 6.*

Step 7: *(Result unproven?) If there was an unproven step, apply Algorithm 5.1 to prove this step. In this case output* G *and the roots* $\alpha_1, \ldots, \alpha_n$. *If the unproven step* $\tilde{H} < \tilde{G}$ *was wrong, then remove* $\tilde{H}$ *from* T, *set* $G \leftarrow \tilde{G}$, *and set* $\alpha_1, \ldots, \alpha_n$ *to the ordering before the critical step.*

We remark that the ordering of the roots is changed in Steps 5 and 6. It may happen that the $r - set$ polynomial F computed in Step 5.2 is not square-free. In this case we have to apply a suitable Tschirnhausen transformation (see Algorithm 5.1). In Step 5.2 *2*, *3* we have to find a good compromise between the degree of the corresponding p-adic field and the number of short cosets. Frobenius automorphisms of large degree usually give smaller short coset systems.

7. Examples

We tested about 70000 polynomials from degree 3 to 15. The running time of the algorithm is dependent on the size of the coefficients and the Galois group. Furthermore it is dependent on the number of Tschirnhausen transformations which usually increase the size of the coefficients. We use the examples from degree 12 to 15 given in Klüners and Malle (2000). The given running times include all necessary computations to get a proven result. All computations were done on a 500MHz Intel Pentium III processor running under SuSE Linux 6.1 and are given in seconds.

Degree 12

Group	Time	Group	Time	Group	Time	Group	Time	Group	Time
1	0.8	45	0.7	89	1.5	133	2.5	177	3.8
2	0.8	46	3.9	90	1.5	134	0.7	178	3.1
3	0.4	47	4.4	91	1.0	135	0.7	179	39.0
4	0.5	48	0.8	92	1.2	136	0.9	180	5.2
5	0.7	49	5.4	93	1.7	137	0.8	181	12.0
6	1.3	50	0.7	94	4.9	138	1.4	182	4.5
7	0.9	51	0.9	95	0.7	139	1.3	183	5.3
8	0.3	52	6.2	96	1.9	140	1.0	184	0.7
9	0.7	53	0.7	97	1.6	141	1.3	185	1.6
10	0.7	54	5.4	98	2.3	142	1.0	186	1.1
11	0.9	55	1.4	99	7.0	143	1.7	187	0.8
12	0.6	56	1.3	100	2.5	144	1.4	188	0.8
13	0.8	57	1.4	101	1.2	145	6.1	189	7.8
14	0.7	58	0.8	102	6.4	146	2.0	190	1.6
15	0.4	59	1.1	103	1.1	147	2.5	191	1.0
16	0.5	60	1.6	104	6.1	148	8.6	192	12.0
17	0.7	61	1.2	105	0.8	149	2.8	193	0.4
18	1.3	62	1.2	106	0.7	150	1.2	194	2.2
19	1.6	63	1.1	107	1.1	151	7.6	195	3.3
20	0.8	64	2.6	108	1.2	152	4.9	196	2.2
21	0.4	65	1.5	109	0.9	153	5.9	197	0.9
22	1.2	66	2.5	110	1.6	154	5.4	198	0.7
23	0.9	67	1.6	111	1.5	155	0.7	199	1.9
24	0.8	68	1.3	112	1.4	156	2.4	200	1.7
25	0.7	69	0.7	113	0.8	157	11.0	201	1.7
26	1.7	70	7.6	114	1.9	158	4.3	202	3.0
27	13.0	71	7.4	115	1.9	159	2.7	203	0.7
28	0.3	72	2.7	116	2.4	160	1.6	204	2.3
29	1.0	73	2.6	117	7.0	161	2.8	205	2.1
30	1.1	74	1.8	118	2.6	162	1.7	206	2.8
31	1.4	75	5.1	119	2.5	163	1.4	207	3.4
32	1.1	76	1.1	120	2.7	164	2.8	208	0.8
33	1.2	77	0.4	121	2.3	165	2.3	209	3.4
34	1.8	78	1.0	122	3.5	166	5.2	210	4.7
35	0.5	79	0.7	123	0.9	167	2.6	211	1.8
36	1.2	80	0.9	124	1.9	168	7.0	212	7.2
37	1.3	81	1.2	125	0.4	169	2.5	213	1.7
38	1.3	82	1.6	126	1.6	170	2.3	214	3.1
39	1.3	83	0.4	127	2.2	171	4.1	215	3.2
40	1.0	84	5.2	128	2.7	172	4.0	216	3.6
41	1.6	85	2.9	129	2.1	173	3.7	217	2.0
42	1.3	86	0.7	130	7.4	174	4.7	218	10.0
43	0.3	87	1.5	131	2.5	175	2.0	219	0.3
44	0.7	88	1.0	132	2.7	176	5.2	220	7.1

Degree 12

Group	Time	Group	Time	Group	Time	Group	Time	Group	Time
221	1.4	238	1.0	255	1.0	272	16.0	289	0.3
222	1.8	239	2.4	256	1.8	273	0.9	290	0.4
223	7.8	240	0.9	257	1.7	274	0.5	291	1.7
224	1.2	241	0.7	258	1.7	275	1.7	292	0.9
225	4.4	242	4.9	259	14.0	276	1.0	293	0.3
226	0.3	243	3.3	260	0.4	277	1.0	294	0.4
227	0.7	244	4.1	261	0.6	278	3.9	295	337.0
228	5.4	245	6.7	262	1.2	279	3.6	296	1.7
229	2.7	246	26.0	263	1.3	280	1.6	297	0.4
230	0.9	247	8.8	264	1.0	281	0.7	298	2.0
231	1.9	248	1.6	265	1.9	282	1.2	299	1.4
232	5.5	249	3.4	266	0.4	283	1.1	300	0.1
233	2.8	250	1.6	267	0.9	284	1.3	301	0.0
234	5.2	251	2.2	268	1.9	285	0.3		
235	0.9	252	2.7	269	1.4	286	1.1		
236	0.6	253	2.0	270	2.3	287	0.8		
237	2.5	254	4.3	271	1.5	288	3.1		

Degree 13

Group	Time	Group	Time	Group	Time	Group	Time	Group	Time
1	8.2	3	2.1	5	1.4	7	2.7	9	0.0
2	6.3	4	14.0	6	3.6	8	0.2		

Degree 14

Group	Time	Group	Time	Group	Time	Group	Time	Group	Time
1	1.5	14	5.4	27	6.7	40	5.5	53	1.1
2	1.1	15	5.1	28	4.8	41	5.2	54	1.7
3	1.9	16	4.0	29	4.7	42	4.7	55	0.8
4	1.4	17	2.8	30	5.9	43	2.2	56	1.0
5	1.4	18	2.8	31	2.7	44	3.5	57	1.1
6	2.3	19	1.6	32	2.2	45	2.0	58	1.3
7	1.4	20	3.4	33	4.2	46	1.1	59	0.5
8	3.9	21	2.1	34	2.0	47	1.2	60	1.6
9	4.1	22	5.4	35	2.0	48	5.5	61	0.4
10	2.0	23	2.8	36	3.2	49	0.5	62	0.0
11	1.8	24	6.7	37	2.4	50	1.8	63	0.0
12	2.4	25	7.4	38	3.1	51	2.1		
13	3.1	26	4.9	39	9.1	52	4.2		

Degree 15

Group	Time	Group	Time	Group	Time	Group	Time	Group	Time
1	1.4	22	0.7	43	4.1	64	7.5	85	5.6
2	1.4	23	1.0	44	4.7	65	45.0	86	3.7
3	1.4	24	1.6	45	3.9	66	26.0	87	9.7
4	1.5	25	4.7	46	3.1	67	11.0	88	1.4
5	3.1	26	3.6	47	15.0	68	5.2	89	0.6
6	1.1	27	5.5	48	7.4	69	1.5	90	1.4
7	1.2	28	3.0	49	6.0	70	2.4	91	1.4
8	1.3	29	0.4	50	3.0	71	2.9	92	1.9
9	5.5	30	3.9	51	3.1	72	9.8	93	1.0
10	5.1	31	6.7	52	7.7	73	4.8	94	1.5
11	1.1	32	5.1	53	1.6	74	11.0	95	1.7
12	4.1	33	4.8	54	4.7	75	5.8	96	1.9
13	3.0	34	2.6	55	4.2	76	1.6	97	2.1
14	5.5	35	4.0	56	4.3	77	1.6	98	1.3
15	3.6	36	3.0	57	20.0	78	2.0	99	0.5
16	1.3	37	43.0	58	28.0	79	4.2	100	1.6
17	14.0	38	8.1	59	5.8	80	2.7	101	1.2
18	4.7	39	8.1	60	5.3	81	5.0	102	0.6
19	5.2	40	9.9	61	2.1	82	5.6	103	0.1
20	7.1	41	5.0	62	1.4	83	1.5	104	0.1
21	7.4	42	5.1	63	2.5	84	4.9		

For all primitive groups of degree 14 and 15 (excepting $A_{14}, S_{14}, A_{15}, S_{15}$), and all examples with more than ten seconds running time we give more details. In the following table Subfield denotes the running time for Algorithm 3.2, which includes subfield computation and group theoretic computations. For primitive groups we give the running time needed for the computation of the resolvent including the necessary transformations. Factor gives the running time for finding the factors of the computed resolvents. In Stauduhar we give the computing time for the Stauduhar steps. The column "All" gives the complete running time rounded to seconds. Looking at the primitive groups we see that the resolvent part is not critical. The worst case is $M_{12} = 12T^{+}_{295}$ since we need an invariant of degree 924. We remark that the coefficients of the polynomials for $15T_{65}$ and $15T_{66}$ are huge compared to the other ones.

Group	Subfield	Resolvent	Stauduhar	Factor	All
$12T_{27}$	0.2		12.8		13
$12T^{+}_{157}$	0.7		9.3		10
$12T^{+}_{179}$	0.0	5.2	10.8	23.0	39
$12T^{+}_{181}$	0.3		11.6		12
$12T_{192}$	0.2		11.8		12
$12T_{218}$	0.0	2.1	2.4	5.2	10
$12T_{246}$	0.6		24.6		26

Group	Subfield	Resolvent	Stauduhar	Factor	All
$12T_{259}^+$	0.3		13.5		14
$12T_{272}^+$	0.0	3.7	6.4	5.6	16
$12T_{295}^+$	0.0	130.2	6.6	200.5	337
$13T_4$	0.0	0.1	11.8	0.1	12
$14T_{30}^+$	0.0	0.1	5.6	0.1	6
$14T_{39}$	0.0	0.8	2.8	4.9	9
$15T_{17}^+$	0.5		13.5		14
$15T_{20}^+$	0.1	0.1	6.6	0.2	7
$15T_{28}^+$	0.1	0.1	2.5	0.2	3
$15T_{37}^+$	0.4		42.6		43
$15T_{47}^+$	0.1	3.6	6.1	4.7	15
$15T_{57}^+$	0.9		18.7		20
$15T_{58}^+$	0.4		27.5		28
$15T_{65}$	2.0		42.9		45
$15T_{66}$	1.8		24.2		26
$15T_{67}^+$	0.6		10.4		11
$15T_{72}^+$	0.0	2.9	2.5	4.4	10
$15T_{74}$	0.6		10.4		11

These examples show the efficiency of our algorithm. For the groups $13T_6$, $13T_5^+$, $14T_{39}$, and $14T_{30}^+$ the index $[G : H]$ is 39916800. Without using short cosets it was impossible to apply Stauduhar's method to these cases. One advantage of the p-adic version of Stauduhar's method is that the algorithm is in polynomial time in the size of the coefficients. The example polynomial f for the group $15T_{65}$ has huge coefficients and our algorithm needs 45s to compute the Galois group. We applied the same algorithm to f (including the use of subfields) but using complex approximations. The following table give the running times and the computed result depending on the used precision:

Precision	Result	Time
100	82	12
200	82	32
300	82	64
400	65	1118

From this table we see another problem of the complex version of Stauduhar's algorithm. When we want to get proven results we have to think about estimations for the used precision. Using a precision which will give proven results the running time will be worse.

8. Acknowledgement

We would like to thank John McKay for suggesting the use of short coset systems. John Cannon and Alexander Hulpke provided maximal transitive subgroups of large transitive groups, and Gunter Malle kindly supplied most of the test polynomials.

References

Acciaro, V., Klüners, J. (1999). Computing automorphisms of abelian number fields. *Math. Comput.*, **68**:1179–1186.

Casperson, D., McKay, J. (1994). Symmetric functions, m-sets, and Galois groups. *Math. Comput.*, **63**:749–757.

Conway, J., Hulpke, A., McKay, J. (1996). On transitive permutation groups. *J. of Comp. and Math.*.

Daberkow, M., Fieker, C., Klüners, J., Pohst, M., Roegner, K., Wildanger, K. (1997). KANT V4. *J. Symb. Comput.*, **24**(3):267–283.

Darmon, H., Ford, D. (1989). Computational verification of M_{11} and M_{12} as Galois groups over $\mathbb{Q}$. *Communications in Algebra*, **17**(12):2941–2943.

Eichenlaub, Y. (1996). *Problèmes effectifs de théorie de Galois en degrés 8 à 11.* Thèse, Université Bordeaux 1.

Eichenlaub, Y., Olivier, M. (1995). Computation of Galois groups for polynomials with degree up to eleven. Preprint, Université Bordeaux I.

Geißler, K. (1997). Zur Berechnung von Galoisgruppen. Diplomarbeit, Technische Universität Berlin.

Girstmair, K. (1983). On the computation of resolvents and Galois groups. *Manuscripta Math.*, **43**:289–307.

Hulpke, A. (1996). *Konstruktion transitiver Permutationsgruppen.* Dissertation, Rheinisch-Westfälische Technische Hochschule, Aachen, Germany.

Kemper, G., Steel, A. (1999). Some algorithms in invariant theory of finite groups. In Dräxler, P., Michler, G., Ringel, C. M., editors, *Proceedings of the Euroconference on Computational Methods for Representations of Groups and Algebras*, Progress in Mathematics, Basel. Birkhäuser.

Klüners, J. (1997). *Über die Berechnung von Automorphismen und Teilkörpern algebraischer Zahlkörper.* Dissertation, Technische Universität Berlin.

Klüners, J. (1998). On computing subfields - a detailed description of the algorithm. *Journal de Théorie des Nombres de Bordeaux*, **10**:243–271.

Klüners, J., Malle, G. (2000). Explicit Galois realization of transitive groups of degree up to 15. *J.Symb.Comput.*. same issue.

Klüners, J., Pohst, M. (1997). On computing subfields. *J. Symb. Comput.*, **24**(3):385–397.

Landau, S. (1985). Factoring polynomials over algebraic number fields. *SIAM J.Comput.*, **14**:184–195.

Landau, S., Miller, G. (1985). Solvability by radicals is in polynomial time. *J. of Computer and System Sciences*, **30**:179–208.

Lenstra, A. K., Lenstra Jr., H. W., Lovász, L. (1982). Factoring polynomials with rational coefficients. *Math.Ann.*, **261**:515–534.

Mattman, T., McKay, J. (1997). Computation of Galois groups over function fields. *Math.Comput.*, **66**:823–831.

Meldrum, J. (1995). *Wreath Products Of Groups And Semigroups.* Pitman Monographs and Surveys in Pure and Applied Mathematics 74.

Schönert, M. *et al.*(1997). GAP 3.4, patchlevel 4. School of Mathematical and Computational Sciences, University of St.Andrews, Scotland.

Soicher, L. (1981). The computation of Galois groups. Master's thesis, Concordia University, Montreal.

Soicher, L., McKay, J. (1985). Computing Galois groups over the rationals. *J. Number Theory*, **20**:273–281.

Stauduhar, R. P. (1973). The determination of Galois groups. *Math. Comput.*, **27**:981–996.

Tschebotarew, N., Schwerdtfeger, H. (1950). *Grundzüge der Galoisschen Theorie.* Nordhoff, Groningen–Djakarta.

von zur Gathen, J., Gerhard, J. (1999). *Modern Computer Algebra.* Cambridge University Press.

Yokoyama, K. (1997). A modular method for computing the Galois group of polynomials. In Cohen, A., Roy, M.-F., editors, *MEGA '96*, volume 117-118 of J.Pure Appl. Algebra, pages 617–636.