



Laboratoire d'Arithmétique, de Calcul formel et d'Optimisation
ESA - CNRS 6090

On the Implementation of a p-adic Method for Computing Galois Groups

Marc J. Rybowicz & Boris Lenzinger

Rapport de recherche n° 2002-05

Université de Limoges, 123 avenue Albert Thomas, 87060 Limoges Cedex
Tél. 05 55 45 73 23 - Fax. 05 55 45 73 22 - laco@unilim.fr

<http://www.unilim.fr/laco/>

On the Implementation of a p -adic Method for Computing Galois Groups

Marc J. Rybowicz
Boris Lenzinger

Laboratoire d'Arithmétique, Calcul formel et Optimisation
Université de Limoges, France
UMR CNRS 6090

rybowicz@unilim.fr

Most of this work was done in 1998 while the authors were visitors of the Symbolic Computation Group, University of Waterloo, Ontario.

Abstract

We present an algorithm and a Maple program to compute the Galois groups of univariate polynomials over the rationals. It combines a relative resolvent method using p -adic approximations of the roots with the well-understood absolute resolvent method. This approach is free of the errors inherent to numerical methods and it turns out that it yields acceptable running times for polynomials with small coefficients. We describe our implementation and some experimental results for irreducible polynomials of degree 9.

1 Introduction

The computation of Galois groups of univariate polynomials is an old problem which has received some attention recently [2, 3, 6, 10, 11, 12, 14, 15, 19, 25, 26]. Several implementations are available (see Section 3 and 4). Besides its historical interest and central role in number theory ([22], see also [9]), applications to Computer Algebra includes the resolution of polynomial equations and the determination of properties of the splitting field of univariate polynomials.

Maple V Release 5 does have a `galois` function that computes the Galois group conjugacy class of irreducible univariate polynomials of degree up to 8. The coefficient field may be \mathcal{Q} , the field of rational numbers, or a multivariate rational function field $\mathcal{Q}(T)$. Unlike some other implementations (see Section 4 for details), Maple's code does certify that the result obtained is correct. The goal was to treat higher degree polynomials within a "reasonable" amount of time—that is, small enough for interactive use—without compromising the reliability. We shall see that this objective can be achieved for degree 9 polynomials with small coefficients. When the size of the coefficients grows, some groups offer some resistance.

We extended the method used by Maple to cover degree 9 polynomials and implemented a prototype [18]. It turned out that a couple of groups could not be identified quickly enough because of the factorization costs of certain abso-

lute resolvents (see Section 3). Therefore, we switched to a relative resolvent method using p -adic approximations of the roots, as suggested by Kazuhiro Yokoyama in [26] and described in Section 4 and 5. The combination of the two approaches yields satisfactory results.

Throughout the paper, $f(x)$ will denote a monic irreducible univariate polynomial over a field k of characteristic zero.

There are essentially three methods for computing Galois groups: The absolute resolvent method, the relative resolvent method and the splitting field method.

We eliminate right away the third one since it requires to a priori determine a representation of the splitting field of f . Typically, the splitting field is computed by factorizations over a tower of algebraic extensions [2]. These factorization steps are beyond the capabilities of current algorithms and Computer Algebra Systems, except for very small degree polynomials (say, less than 5 or 6) or for a small number of special cases (normal extensions). The only advantage of this method seems to be that it is not necessary to know in advance the list of possible groups.

We will focus on resolvent methods. They require a table of conjugacy classes of transitive permutation groups. Such tables have been computed and published up to degree 15 [7, 13]. We shall use the "T" notation: nTm , or Tm if the context is clear, is the m -th group of degree n as classified in [7].

In Section 2, we introduce notations and recall classical facts about resolvents and Galois groups. Sections 3 and 4 are devoted to a description of known algorithms and a brief review of their implementations. The following section describes the p -adic approach of the relative resolvent algorithm. In Section 6, we detail some aspects of our implementation which are crucial to obtain a satisfactory efficiency. Finally, we report some experimental results in Section 7.

2 Galois groups and resolvents

The main purpose of this section is to introduce some notations. All the results and definitions below have been collected in the literature (see for instance [9, 14, 22]).

Let k be a field of characteristic zero and $f(x)$ be an irreducible univariate polynomial of degree n in $k[x]$. We denote by $\alpha = \{\alpha_1, \dots, \alpha_n\}$ the roots of f in an algebraic closure \bar{k} of k . The Galois group Γ of f over k is the group of auto-

morphisms of $k(\alpha)$ which fix the elements of k . The group Γ permutes transitively the α_i 's and the action on the indices identifies Γ with a transitive subgroup of the symmetric group S_n . For τ and σ in S_n , we shall use the following notation: $\tau\sigma$ is the permutation such that $\tau\sigma(i) = \tau(\sigma(i))$ for all i in $\{1, \dots, n\}$.

Given a set of indeterminates $X = \{X_1, \dots, X_n\}$, S_n acts on X in a natural way and this action extends to the polynomial ring $k[X]$: If $\tau \in S_n$ and $P(X) \in k[X]$ then we will denote by τP the polynomial $\tau P(X_1, \dots, X_n) = P(X_{\tau(1)}, \dots, X_{\tau(n)})$.

Let G and H be two subgroups of S_n such that $H \subset G$. We shall say that a polynomial $P(X) \in k[X]$ is a *primitive invariant of H relative to G* , or a (G, H) -invariant for short, if $\text{Stab}_G(P) = H$. In other words, a (G, H) -invariant is a primitive element of the extension $k(X)^H/k(X)^G$ where $k(X)^H$ is the subfield of $k(X)$ formed by the elements of $k(X)$ fixed by H (similarly for $k(X)^G$).

For each pair (G, H) , we choose a primitive invariant P and we define:

$$R_{G,H}(T, X) = \prod_{\sigma \in G/H} (T - \sigma P(X))$$

where G/H denotes a set of representatives of the left cosets of H in G (that is, G/H). This polynomial will be called the *generic relative resolvent* for G and H associated with P . Its coefficients are invariants of G . The specialized polynomial $R_{G,H}(T, \alpha)$ is the *relative resolvent* of f associated with G , H , and P . It will be abbreviated by $R_{G,H}(T)$.

If G happens to be S_n , then $R_{G,H}(T, X)$ and $R_{G,H}(T)$ are called *absolute resolvents* and will be denoted by $R_H(T, X)$ and $R_H(T)$. The coefficients of an absolute resolvent are symmetric functions of the roots of f and are therefore elements of k . Similarly, the coefficients of a generic absolute resolvent are symmetric functions of the X_i 's.

From now on, we assume that $\Gamma \subset G$. Then, Γ acts on the set of left cosets G/H . If $R_{G,H}(T)$ is squarefree, Γ also acts on the roots of $R_{G,H}(T)$ and any ordering of the roots of $R_{G,H}(T)$ in \bar{k} , gives an homomorphism ϕ from Γ to S_m where $m = (G : H) = \deg_T R_{G,H}(T)$.

Resolvent based methods stem from the following theorem ([9]) :

Theorem 1 *If $R_{G,H}(T)$ is squarefree, then:*

1. *the action of Γ on G/H is equivalent to the action of Γ on the roots of $R_{G,H}(T)$,*
2. *the Galois group of $R_{G,H}(T)$ is isomorphic to $\phi(\Gamma)$.*

Consequently, provided that $R_{G,H}(T)$ is squarefree, the degree of the irreducible factors of $R_{G,H}(T)$ as well as their Galois group over k depend only on Γ ; some information about Γ can therefore be obtained from the factorization of the resolvents.

For instance, the following method appeared in Camille Jordan's "Traité des substitutions algébriques" [17] : If $G = S_n$, then Γ is included in a conjugate of $\sigma H \sigma^{-1}$ of H if and only if the orbit of σH under the action of Γ contains only one element. By Theorem 1 this is true if and only if a squarefree absolute resolvent for H has a rational root. Hence, it is sufficient to know an absolute resolvent for each candidate group H —up to conjugacy—and to search for rational roots of these resolvents.

This method has no practical interest because the degrees of the resolvents involved are far too large. Nonetheless, the more recent methods surveyed in the next two sections can be viewed as refinements this nineteenth century technique.

3 Absolute resolvents methods

In this section, the group G of Section 2 is set to S_n . From Theorem 1, the degrees of the irreducible factors over k of an absolute resolvent $R_H(T)$ are in one to one correspondence with the orbit lengths of the action of Γ on G/H . For each possible value of Γ and each group H , the degrees of the irreducible factors of an absolute resolvent R_H can be tabulated in advance. Such a table is called a *partition table* by Arnaudis and Valibouze. The absolute resolvent method consists in building a partition table for a set of well-chosen test groups H . The Galois group of f will be identified by factorization of absolute resolvents associated with the groups H and inspection of the partition table. Some methods have been published for computing families of particular absolute resolvents [20]. Published tables [4, 5, 3, 14] allow to identify in theory all groups up to degree 11, but for any degree n , it is always possible to find a set of test groups H that will identify Γ . Unfortunately, the factorization stage of this algorithm becomes quickly too expensive. The degree of the resolvent to factor is the index of H in S_n . When this degree exceeds 100, the complete factorization may already be out of the scope of current algorithms. Indeed, resolvents are polynomials which are particularly hard to factor. Over \mathcal{Q} , for instance, the Galois groups of $R_H(T)$ will contain only cycles of length less than n , so that resolvents have many factors modulo primes numbers. Thus, the exponential time combination stage of the factorization algorithm is extremely expensive.

To alleviate this shortcoming, some authors [14, 19, 25] suggested to choose H so that only small degree factors of the resolvent need to be considered. They also proposed to take advantage of additional information about the resolvent factors, such as the parity of their Galois groups, or even their Galois groups themselves. These refinements have led to the Maple implementation for $n = 8$, $k = \mathcal{Q}$ or $k = \mathcal{Q}(t)$.

Maple's absolute resolvent method has been implemented by Ron Sommeling, Thomas Mattman and John McKay for irreducible polynomials of degree up to 8. The implementation for degree up to 7 is based on [20] while the treatment of the more difficult degree 8 case is described in [19]. This implementation is reasonably fast: For instance, the computation of the group of any degree 8 polynomial in the Maple test suite takes less than 40 seconds¹. This test suite is made of polynomials with small coefficients from [19, 23] and covers each possible group. Note that this is an upper bound: the program is significantly faster for most of these polynomials, partly because some heuristics, such as factorization modulo some primes, are used to quickly eliminate some candidates. This Maple implementation also supports polynomials over $\mathcal{Q}(T)$, where T is a set of indeterminates. The program is part of the standard Maple V Release 5 library.

For degree 9, the tables aforementioned show that some

¹Timings are given for a DEC Alpha 3000/800 S with 256 Mb of RAM running Digital Unix 3.2. This machine is roughly 1.5 times slower than a Pentium Pro 200 PC running Windows NT. The version of Maple is Maple V Release 5.

groups will be particularly difficult to separate. In [4], the authors suggest to search for a degree 28 factor of a resolvent of degree 280 to distinguish T_{33} from its subgroups T_{32} and T_{27} : If there is no such factor, the group is T_{33} , otherwise it is T_{32} , or T_{27} . Assuming a maximum number of modular factors of degree 9, the resolvent will have at least 32 modular factors (31 of degree 9 and one of degree 1). Proving that a factor of degree 28 does not exist will require at least $\binom{31}{3} = 4495$ combinations. This is a gross underestimate since in general there will be many more modular factors. Other cases cause similar concerns.

An implementation of the absolute resolvent method in Gap 3.4 covers degrees up to 15. However, the timings are disappointing and the code is consistently slower than Maple's in degree 8. Gap needs up to 3 hours for some examples that Maple resolves in a couple of seconds. In degree 9, some tests chosen in [15] and corresponding to T_{32} and T_{27} do not seem to terminate (process killed after 60000 sec.). It is not clear whether these overall humble performances are due to the lack of heuristics, weaknesses of the factorization code or inappropriate choices of resolvents.

From these considerations, it seems that a method based only on absolute resolvents cannot be reliably "fast" for all groups of degree 9.

Nonetheless, some small degree absolute resolvents shall prove very useful. In particular absolute resolvents associated with the group $S_r \times S_{n-r}$ (with $1 < r < n$) are commonly used and bear some advantages: They are easy to compute and have reasonably low degrees for small values of n and r : in degree 9, a 2-set resolvent has degree 36 and a 3-set resolvent has degree 72. Following [20] we shall call them *r-set resolvents*.

4 Relative resolvents methods

From Theorem 1, it is clear that Γ is included in a conjugate of H under G if and only if $R_{G,H}(T)$ has a rational root. The permutation τ associated to the root gives the conjugate, that is $\Gamma \subset \tau H \tau^{-1}$. The idea is to start from $G = S_n$ (or $G = A_n$ if Γ is even) and to traverse the lattice of transitive subgroups from the largest groups to the smallest by testing the inclusion of Γ in the maximal subgroups of the current group G . Let H be a representative for a conjugacy class of maximal subgroups of G .

Following [15] we introduce the notation:

$$\mathcal{H}(G, H) = \{\sigma H \sigma^{-1} \mid \sigma \in S_n \text{ and } \sigma H \sigma^{-1} \subset G\}.$$

At each stage, one needs to test the inclusion of Γ in an element of $\mathcal{H}(G, H)$. In [15], it is explained that it is sufficient to compute only one relative resolvent for each orbit of $\mathcal{H}(G, H)$ under the action of G .

The relative resolvent method was presented by Stauduhar in [24] for polynomial of degree up to 7 and extended to degree 11 by Eichenlaub and Olivier [15]. Lattices of transitive subgroups, relative resolvents, test polynomials, as well as timings have been published in [15] for degrees 9 to 11. The corresponding C implementation by Eichenlaub for $k = \mathcal{Q}$ relies on numerical approximations of the roots of f to estimate the coefficients and the roots of the resolvents.

The program performs well for degree 8 and 9 (less than 30 sec.)², degree 10 (less than 110 sec.) but there are some

²The timings and the test suite are from [15]. The test suite includes parameterized polynomials specialized at random small integer values

difficult cases in degree 11 (more than 50 min.).

Another implementation of this method is available with Kant 1.7. The running times in degree 8 and 9 are consistent with those obtained by Eichenlaub's program. Kant 1.8 does cover groups of degree 12. The latter implementation also makes use of the computation of imprimitivity block systems in order to eliminate some imprimitive groups from the candidate list [8]. Unlike the absolute method, the relative method as implemented in Kant and by Eichenlaub also provides the action of the group on the roots.

Unfortunately, the efficiency of both implementations relies on heuristic bounds for the numerical approximations of the roots. Therefore, the group returned by these programs is not *provably* the Galois group. Moreover, round-off errors add to the uncertainty of the result. The precision required to guarantee an exact result has been studied in [14, 16], but leads to worse performances, since the number of necessary digits is high.

Two authors proposed algorithms to avoid numerical computations.

Colin in [10] presented a formal method to compute relative resolvents based on fairly heavy invariant theory computations. Although this approach sounds attractive, it remains to be proved that this method is competitive from a practical point of view. We are not aware of any implementation. Improvements have been published in [11].

In [26], Yokoyama replaced numerical approximations by p-adic approximations and reported competitive times of a Risa/Asir implementation in easy cases (degrees up to 7 with small coefficients). His paper motivated our implementation of a similar algorithm for degree higher than 8.

5 The p-adic method

In this section, the ground field k is the field of rational numbers \mathcal{Q} . The ring of integers is denoted by \mathcal{Z} . At each stage of the relative resolvent method it has been proved that, for a given ordering of the roots of f , the group Γ is included in a group G . The initial inclusion is $\Gamma \subset S_n$. Let H be a maximal transitive subgroup of G and $R_{G,H}(T)$ a relative resolvent.

The problem is to distinguish between the three situations below:

1. $R_{G,H}$ has a simple rational root associated with a permutation σ . In this case, $\Gamma \subset \sigma H \sigma^{-1}$ and the permutation σ will be returned in order to permute the roots of f and proceed recursively with maximal subgroups of $\sigma H \sigma^{-1}$.
2. $R_{G,H}$ has no rational root. In this case, Γ is included in no conjugate of H under G .
3. $R_{G,H}$ has multiple rational roots. Theorem 1 is not conclusive. A random Tschirnhaus transformation is applied to f and resolvents corresponding to the new polynomial are considered. The transformation is repeated until one of the two situations above occur. This loop is known to terminate.

If $\mathcal{H}(G, H)$ has several orbits under the action of G , the test must be repeated for each orbit. For degree less than 12, there is only one orbit for most pairs (G, H) .

To perform this decision step, Yokoyama [26] replaced numerical approximations of the roots of f originally used in [24] by p-adic approximations.

Without loss of generality, we assume from now on that f is monic and has coefficients in \mathcal{Z} . As a consequence, the roots of $R_{G,H}$ are integral elements of $k(\alpha)$ and rational roots must be in \mathcal{Z} .

In order to describe the p -adic method, let us introduce some notations: Let p denote a prime integer such that f is squarefree modulo p . We denote by \mathcal{Z}_p the ring of p -adic integers and by \mathcal{Q}_p the fraction field of \mathcal{Z}_p . Let d denote an integer such that f splits into linear factors in $GF(p^d)$. There is a unique non-ramified extension of \mathcal{Q}_p of degree d and we call it K_p (see for instance [21]). The polynomial f has n distinct roots in K_p and we denote them by $\beta = \{\beta_1, \dots, \beta_n\}$. If θ is an integral element of K_p with the following expansion:

$$\theta = \theta_0 + \theta_1 p + \theta_2 p^2 + \dots,$$

then for each positive integer i , we define:

$$\theta^{(i)} = \theta_0 + \theta_1 p + \theta_2 p^2 + \dots + \theta_{i-1} p^{i-1}.$$

The approximations $\beta^{(i)} = \{\beta_j^{(i)}\}_{1 \leq j \leq n}$ of the β_j 's satisfy:

$$f(\beta_j^{(i)}) \equiv 0 \pmod{p^i}.$$

For $\theta \in \mathcal{Z}_p$, denote by $\phi_i(\theta)$ the integer such that:

$$\begin{aligned} \phi_i(\theta) &\equiv \theta^{(i)} \pmod{p^i} \\ \phi_i(\theta) &\in [-(p^i - 1)/2, p^i/2] \end{aligned}$$

The decision procedure reads as follow:

Inclusion test.

- Compute an upper bound ρ for the modulus of the roots of $R_{G,H}$ in \mathcal{C} . Choose a real number M such that $M > \max\{2(G : H), \rho\}$.
- Choose an integer e such that:

$$p^e > (2M)^{(G:H)}. \quad (1)$$

- Compute $\beta^{(e)}$.
- Compute approximations $\{\gamma_\sigma^{(e)}\}_{\sigma \in G//H}$ of the roots of $R_{G,H}$ in K_p using $\beta^{(e)}$, $G//H$ and P .
- If there is a τ such that $\gamma_\tau^{(e)}$ satisfies:
 1. $\gamma_\tau^{(e)}$ is in \mathcal{Z}_p ,
 2. $|\phi_e(\gamma_\tau)| < M$,
 3. $\gamma_\tau^{(e)} \neq \gamma_\sigma^{(e)}$ if $\tau \neq \sigma$,

then γ is a simple root of $R_{G,H}$ in \mathcal{Z} . Permute β and its approximations according to τ and terminate.

- If there is a root which satisfies conditions 1 and 2 above but not condition 3, $R_{G,H}$ has multiple roots in \mathcal{Z} and a Tschirnhaus transformation is required.

The correctness of this inclusion test is proved in [26].

The initial approximations $\beta^{(1)}$ are the roots of f in $GF(p^d)$. Typically, $\beta^{(e)}$ is computed from the best approximations already known using a lifting procedure, such as the

following quadratic double Newton iteration which appears in [1]:

$$\begin{aligned} \xi_{j+1} &\equiv \xi_j - \omega_j f(\xi_j) \pmod{p^{2^{j+1}}} \\ \omega_{j+1} &\equiv \omega_j [2 - \omega_j f'(\xi_j)] \pmod{p^{2^{j+1}}} \end{aligned}$$

where ξ_0 is a root of f modulo p and

$$\omega_0 \equiv 1/f'(\xi_0) \pmod{p}.$$

In [26] it is proved³ that one can chose:

$$M = \max\{2(G : H), CN \|f\|^v\} \quad (2)$$

where $\|f\|$ is the 2-norm of f , C is a bound for the coefficients of P , $v = \max\{\deg_{X_i} P\}$ and N is the number of monomials of P .

6 Implementation

From now on, we will be concerned with degree $n = 9$, although many of our comments apply to larger degrees. There are 34 possible conjugacy classes for degree 9. Our implementation is based on the following remarks.

Representation of elements of K_p . The field $GF(p^d)$ will be represented⁴ as $GF(p^d) = GF(p)[x]/(F(x))$ where F is an irreducible polynomial of degree d . The approximations $\beta^{(e)}$ can be viewed as elements of $\mathcal{Z}[x]/(p^e, F)$ and represented by univariate polynomials of degree less than d over $\mathcal{Z}/(p^e)$. Arithmetic operations are coded using Maple's fast internal `modp1` package. The prime p is searched from 60013. This ensures that in practice, elements of $GF(p^d)$ will be internally represented by arrays of machine integers.

Cost estimates. The cost of a straightforward implementation of this algorithm is dominated by:

- The lifting of the roots of f . Using classical operations for integer arithmetic and polynomial division, the running time of a lifting up to order e is proportional to:

$$(n+1)d^2 e^2 \quad (3)$$

From (1), we get:

$$(n+1)d^2 (G : H)^2 \log(2M)^2 \quad (4)$$

- The computation of the roots of the resolvents. If m is the number of multiplications in $\mathcal{Z}[x]/(p^e, F)$ required to evaluate P , a gross estimate gives a cost proportional to:

$$m d^2 e^2 (G : H) \quad (5)$$

If (1) is taken into account, we get:

$$m d^2 (G : H)^3 \log(2M)^2 \quad (6)$$

³The bound given [26] is actually more precise, but it will not make any significant difference in practice.

⁴Note that Yokoyama [26] covers a more general situation: the splitting field of f over $GF(p)$ is represented as the quotient of a multivariate polynomial ring by a Gröbner basis. Since it is not clear that there will be a practical gain for degree 9, we have simplified his approach.

- The computation of initial approximations for the roots. An application of Cantor-Zassenhaus' algorithm yields:

$$dn^3 \log p \quad (7)$$

Note that this step can be bounded independently of f if the cost of the initial reduction modulo p is not taken into account.

Choice of p . It is clear from (4), (6) and (7) that the degree of the splitting field should be kept as low as possible by an appropriate choice of p . Inspection of the tables in [7] giving the number of permutations of a given cycle type shows, by application of Chebotarev's density theorem, that the probability for d to be no greater than 4 is asymptotically greater than 0.10. It is therefore reasonable to search for a prime p such that $p \leq 4$. Successive primes are tried⁵ and if the search terminates after less than 15 trials, another is started to attempt to decrease d . It is tempting to try to decrease d further, but for T_{33} and T_{34} , the probability to obtain $d = 3$ is less than 0.05. For $d = 2$, the probability is about 0.01 in the cases T_{17} , T_{33} and T_{34} . Note that if the group is T_{33} or T_{34} chances are that it will be identified by factorization modulo primes. If this heuristic is used by the program, then it is worth trying to reach $d = 3$, or even $d = 2$.

Evaluation of the invariants. It is important to keep the number of multiplications as low as possible when the primitive invariant P is evaluated. Many of the primitive invariants in [15] are generated by a monomial $\mathcal{M}(X)$: In this case, P is the sum of the orbit of $\mathcal{M}(X)$ under the action of H . Since all the $\{\tau P(X)\}_{\tau \in G/H}$ will eventually have to be evaluated, each monomial of the orbit may have to be evaluated many times. For instance, the monomial $\mathcal{M}(X)$ proposed for determining if Γ is included in the maximal subgroup T_{26} of T_{34} is $X_7 X_8 X_9$ which leads to an invariant P having 12 monomials. Since $(T_{34} : T_{26}) = 840$, a brute-force evaluation of the roots of $R_{T_{34}, T_{26}}$ may require 20160 multiplications. But one may notice that there are only 84 distinct monomials of the form $X_i X_j X_k$. Therefore, tabulating the values of the monomials yields a considerable speed up.

Early detection of irrational roots. To prove that Γ is *not* included in a conjugate of H , it is in general not necessary to lift the roots up to the bound e . First of all, if $d > 1$, a computation modulo p will identify many roots of $R_{G,H}(T)$ modulo p that are not in $GF(p)$ and therefore cannot possibly correspond to an integer root. The larger d is, the more candidates are eliminated by this check. Secondly, consider a root γ_τ of $R_{G,H}$ in K_p and suppose that $\gamma_\tau^{(i)}$ is in \mathcal{Z}_p for some i . If $|\phi_i(\gamma_\tau)| > M$, it is obvious that for $j > i$, either $\gamma_\tau^{(j)}$ is not in \mathcal{Z}_p , or $|\phi_j(\gamma_\tau)| > M$. Therefore, γ_τ can also be removed from the candidate list. In our program, we first lift approximations up to the heuristic bound $p^{\tilde{e}}$ with $\tilde{e} = \max\{2 \log_p M, 10 \log_p(10M)\}$. It turns out that it is enough in practice to prove that Γ is not included in H ; the cost (4) drops to a value proportional to $\log^2(G : H)$ (see (2)) while (6) becomes a function of $(G : H) \log^2(G : H)$. If

⁵Here we assume some uniformity of the factorization of f modulo primes.

it is not conclusive, either Γ is included in H with a very high probability⁶ or $R_{G,H}(T)$ has multiple roots.

Early detection of multiple roots. If after a lifting to order \tilde{e} multiple roots are detected, a Tschirnhaus transformation is applied. Since these roots do not provably correspond to multiple roots of the resolvent in R , this test is used at most 4 times. Experimentally, this avoids almost always unnecessary lifting of the roots.

Parity. As usual, we first determine whether the discriminant of f is a square in \mathcal{Z} . If this is the case, then Γ is included in A_9 .

Absolute resolvent of small degree. Partition tables show that irreducible factors of a 2-set resolvent provide very useful information at low cost, since they have degree 36 for $n = 9$. First of all, they are sufficient to identify T_9 and T_{16} . More importantly, they separate imprimitive groups from primitive groups. Since primitive groups and imprimitive groups can be treated independently, provided the imprimitive ones are considered first, this information allows to immediately choose the appropriate sub-lattice in [15]. Finally, they separate T_4 and T_8 from the other odd groups and T_1, T_2, T_3 and T_5 from the remaining even groups. In the next paragraph, we shall also need to separate T_1 and T_3 from T_2 and T_5 . It turns out that it is sufficient to search for two degree 3 factors of a 3-set resolvent.

Certificate of inclusion. At this point, we are left with a small number of candidates for the rational roots of $R_{G,H}$ and there is a high probability that these roots are indeed integers. The order e is particularly large when the index $(G : H)$ is large. An inspection of the subgroup lattices in [15] shows that the indices are large when $G = S_9 (= T_{34})$ or $G = A_9 (= T_{33})$ and no greater than 4 otherwise. In the latter case, we lift the roots up to the bound e to prove or disprove inclusion.

For the maximal transitive subgroups of T_{34} and T_{33} we have: $(T_{34} : T_{26}) = (T_{33} : T_{23}) = 840$, $(T_{34} : T_{31}) = (T_{33} : T_{30}) = 280$, $(T_{33} : T_{32}) = 120$. Note that the groups T_{32} , T_{26} , and T_{23} are primitive, T_{31} and T_{30} are not.

If Γ is odd and imprimitive, it is included in T_{31} . If Γ is even and imprimitive, it must be included in T_{30} . Therefore, from the remark of the previous paragraph, 2-set resolvents suffice to decide inclusion in these cases.

Similarly, if Γ is an odd primitive group, Γ is included in T_{26} if and only if a 3-set resolvent (degree 84) has an irreducible factor of degree 12. For even primitive groups, if $\Gamma \not\subset T_{32}$, then the same criterion will decide whether Γ is included in T_{23} or not. We must say that we have not found a similar trick to characterize the inclusion in T_{32} .

In case we have proved that Γ is included in a maximal subgroup H of S_9 or A_9 using an absolute resolvent, one important issue remains: In order to proceed with the descent in the subgroup lattice, one needs to know in which conjugate of H the group Γ lies. This information is not provided by absolute resolvents, but can be obtained as follows: Since we know that the relative resolvent $R_{G,H}(T)$ has at least one rational root, one can lift the roots of $R_{G,H}(T)$

⁶This is an empirical observation that should be made more precise.

until the candidates are distinct and the number of remaining candidates is equal to the number of expected rational roots. For inclusions in T_{26} and in T_{23} , this number is always one. For inclusions in T_{31} and T_{30} , it depends on Γ , but it may be precomputed⁷ by Theorem 1. It turns out that if Γ is T_2 or T_5 , the number of rational roots is 4, if Γ is T_4 or T_8 , the number of rational roots is 2. In all other relevant cases, it is one. By the remarks of the previous paragraph, the expected number of rational roots can therefore be pre-determined using 2-set and 3-set resolvents.

Factorization of absolute resolvents. We have modified Maple's factorization code so that only modular factor combinations of the relevant degree are tried. For the degree 12 factor of the 3-set resolvent, it achieves a significant speed-up. It is unfortunate that factorization programs of Computer Algebra Systems do not offer this feature.

Tschirnhaus transformations. The random Tschirnhaus transformations that we apply to f tend to enlarge the size of the coefficients and considerably slow down the computation. So, we start with small degree and small coefficient transformations and progressively increase the size of the coefficients and the degree.

A remark about the lifting bound. One can prove that the bound (1) can be refined:

Proposition 1 *Let D be an upper bound for the degrees of the irreducible factors of $R_{G,H}$. If $p^e > 2(2M)^D$, then the inclusion test of the previous section will correctly detect integer roots and multiple integer roots.*

If $\Gamma = G$, the largest degree is $(G : H)$ since the action of G on G/H is transitive. Apparently, there is no improvement. However, if we know in advance that $\Gamma \neq G$ (see above), this proposition may provide a valuable help to decide between several maximal subgroups of G . In degree 9 though, the proposition is not very useful. For inclusions in maximal subgroups of S_9 , once the primitivity or imprimitivity of Γ has been determined, there is only one subgroup class left. For inclusions in maximal subgroups of A_9 , there are two classes (T_{32} and T_{23}) if Γ is primitive, but we unfortunately do not know how to a priori eliminate T_{33} . For other inclusions, $(G : H)$ is small and the proposition can only have a limited impact.

7 Experimental results

In [15], two lists of degree 9 test polynomials are presented. The first test suite, that we shall call \mathcal{S} , is a list of 34 monic polynomials f_i with small coefficients in \mathcal{Z} such that f_i has Galois group T_i . The coefficients have at most four digits, and one or two digits in most cases. The second test suite contains 34 polynomials $P_i(t, x)$ in $\mathcal{Q}(t)[x]$ which covers all cases over $\mathcal{Q}(t)$. By specializing $P_i(t, x)$, one obtains a polynomial whose group over \mathcal{Q} is very often the group of P_i over $\mathcal{Q}(t)$. We build two test suites out of this parametrized families: In the first one, t is replaced by a random one digit integer and denoted \mathcal{S}_1 . In the second one, called \mathcal{S}_2 , t is evaluated at a two digits integer.

⁷We have used the Maple group package, which is not very powerful but good enough for this purpose.

After specialization at a small integer n , $|n| < 100$, the P_i 's may sometimes have rather large coefficients. Moreover, some of these polynomials do not have coefficients in \mathcal{Z} but in \mathcal{Q} . Since our implementation supports only monic polynomials, a brute-force transformation $f(x) \rightarrow f(x/\text{lcoeff}(f(x)))$ is applied. This transformation may multiply the size of the coefficients by 8. Since the algorithm is quadratic in $\|f\|$ (see (2) and the subsequent estimates), this growth of the coefficient dramatically slows down the program.

For the sake of transparency, our timings below correspond to an implementation which *do not use factorization modulo p* to remove groups from the candidate list. It is clear that a final version of our program should do so. Most distributed implementations do use such heuristics.

Running times are given in seconds and rounded to the upper value. The machine is a DEC alpha 3000/800 S with 256 Mb of RAM and running Digital Unix. This machine appears to be 1.5 slower than a Pentium Pro 200 PC.

For \mathcal{S} , the average running time is 16 sec., the largest running times is 40 sec. Not surprisingly, it is obtained for a primitive group. Then come T_{14} , T_{27} and T_{32} , which are also primitive groups. The smallest running time is 1 sec., obtain for T_9 , which is identified by mere factorization of a 2-set resolvent.

The computation of the roots of f in $GF(p^d)$ accounts for a significant part of the running time when the d is 3 or 4 (up to 30% for T_{34} or T_{33}).

For \mathcal{S}_1 and \mathcal{S}_2 , the number of digits of the coefficients has also been specified (column "Size").

The average running time for \mathcal{S}_1 is 29 sec. and jumps to 134 sec. for \mathcal{S}_2 .

For T_{27} , the excessive running time is due to the lack of an efficient method to test the inclusion in T_{32} . The roots need to be lifted up to order $e = p^{2639}$ (in fact p^{4096} since we use quadratic lifting) and the bound M has 105 digits. This example demonstrates the kind of running times that our program frequently yielded before the improvements explained in the previous section were implemented, even when the input had "small" coefficients.

8 Conclusion

We have shown that, the absolute and the relative methods are complementary. The absolute method allows to reach at a reasonable cost the maximal subgroups of A_9 and S_9 , except maybe for T_{32} . From there, the relative method can be applied at a reasonable cost. For small coefficient input, the method seems to be sufficiently fast for interactive use, without compromising the reliability with heuristic bounds.

Further refinements, such as a special representation for p -adic expansions, faster computation of the splitting field modulo p and mixed quadratic/linear lifting are under investigation. A method that reduces the coefficient size of the input could prove essential. We do not know at this point if the algorithm given in [9] for this task could provide a significant speed-up.

A final implementation should also take into account the complete factorization of the 2-set resolvent and, of course, heuristics such as factorization modulo several primes to quickly eliminate some candidates.

T_i	S	S_1	Size	S_2	Size
T_1	7	11	16	20	34
T_2	12	16	5	18	7
T_3	10	15	6	42	14
T_4	14	8	2	9	4
T_5	17	91	27	722	74
T_6	9	19	8	39	11
T_7	12	18	5	81	17
T_8	14	20	7	23	11
T_9	2	7	23	11	47
T_{10}	12	18	2	7	2
T_{11}	8	25	15	67	17
T_{12}	11	18	4	9	5
T_{13}	20	46	13	93	16
T_{14}	40	45	6	153	47
T_{15}	16	22	6	85	55
T_{16}	3	3	4	3	6
T_{17}	14	12	3	23	3
T_{18}	13	9	3	14	3
T_{19}	17	22	13	22	12
T_{20}	13	18	3	18	3
T_{21}	22	44	15	18	17
T_{22}	16	23	8	91	32
T_{23}	40	76	26	141	51
T_{24}	15	15	3	13	4
T_{25}	14	12	3	16	7
T_{26}	14	15	3	17	7
T_{27}	38	115	4	2574	52
T_{28}	13	16	8	15	7
T_{29}	12	22	3	24	11
T_{30}	13	13	3	26	4
T_{31}	10	16	3	11	4
T_{32}	36	113	6	91	6
T_{33}	29	34	2	39	5
T_{34}	20	16	2	19	3

Figure 1: Running times and size of coefficients

Acknowledgment. We would like to thank John McKay and Thomas Mattman for a stimulating and informative conversation. Yves Eichenlaub provided electronic versions of his test suite. Alexander Hulpke kindly suggested several useful absolute resolvents for separating groups in degree 9.

References

- [1] ACCIARO, V., AND KLÜNERS, J. Computing Automorphisms of Abelian Number Fields. Preprint.
- [2] ANAI, H., AND YOKOYAMA, K. Computation of the Splitting Fields and the Galois Groups of Polynomials. In *Proceedings of MEGA 94* (1994), Birkhäuser.
- [3] ARNAUDIÈS, J.-M., AND VALIBOUZE, A. Groupes de Galois de polynômes de degré 10 ou 11. Tech. Rep. 94.50, LITP, Université Paris 6, 1994.
- [4] ARNAUDIÈS, J.-M., AND VALIBOUZE, A. Groupes de Galois de polynômes en degré 8. Tech. Rep. 94.25, LITP, Université Paris 6, 1994.
- [5] ARNAUDIÈS, J.-M., AND VALIBOUZE, A. Groupes de Galois de polynômes en degré 9. Tech. Rep. 94.30, LITP, Université Paris 6, 1994.
- [6] ARNAUDIÈS, J.-M., AND VALIBOUZE, A. Lagrange Resolvents. *J. of Pure and Applied Algebra* (1996). To appear.
- [7] BUTLER, G., AND MCKAY, J. The Transitive Groups of Degree up to Eleven. *Comm. in Algebra 11* (1983), 863–911.
- [8] CASPERSON, D., FORD, D., AND MCKAY, J. Ideal Decomposition and Subfields. *J. Symb. Comp.* 21, 2 (1996), 133–137.
- [9] COHEN, H. *A Course in Computational Algebraic Number Theory*. Springer-Verlag, 1993.
- [10] COLIN, A. Formal Computation of Galois Groups with Relative Resolvents. In *Proceedings AAEECC '95* (1995), vol. 948 of *LNCS*, Springer-Verlag, pp. 169–182.
- [11] COLIN, A. Relative Resolvents and Partition Tables in Galois Group Computation. In *Proceedings ISSAC '97* (1997), ACM Press, pp. 78–84.
- [12] COLIN, A. *Théorie des invariants effective. Applications à la théorie de Galois et à la résolution des systèmes algébriques. Implantation en Axiom*. PhD thesis, École Polytechnique, France, 1997.
- [13] CONWAY, J. H., HULPKE, A., AND MCKAY, J. On Transitive Permutation Groups. *London Mathematical Soc. J. of Comp. and Math.* (1996). electronic publication.
- [14] EICHENLAUB, Y. *Problèmes Effectifs de Théorie de Galois en degrés 8 à 11*. PhD thesis, Université de Bordeaux, 1996.
- [15] EICHENLAUB, Y., AND OLIVIER, M. Computation of Galois Groups for Polynomials of Degree up to Eleven. *Preprint, Université de Bordeaux* (1995).
- [16] FORD, D. J., AND MCKAY, J. Computation of Galois Groups from Polynomials over the Rationals. In *Computer Algebra* (1989), vol. 113 of *L. N. Pure Appl. Math.*, Springer-Verlag, pp. 145–150.
- [17] JORDAN, C. *Traité des substitutions et des équations algébriques*. Gauthier-Villars, Paris, 1870.
- [18] LENZINGER, B. Computation of the Galois Group of an Irreducible Polynomial over the Rationals up to Degree 9. Work term report, Symbolic Computation Group, University of Waterloo.
- [19] MATTMAN, T., AND MCKAY, J. Computation of Galois Groups over Function Fields. *Math. Comp.*. To appear.
- [20] MCKAY, J., AND SOICHER, L. Computing Galois Groups over the Rationals. *J. Number Theory* 20 (1985), 273–281.
- [21] NARKIEWICZ, W. *Elementary and Analytic Theory of Algebraic Number*. Springer-Verlag, 1990.

- [22] POHST, M., AND ZASSENHAUS, H. *Algorithmic Algebraic Number Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1989.
- [23] SMITH, G. W. Some Polynomials over $\mathbb{Q}(t)$ and their Galois Groups. *Math. Comp.*. To appear.
- [24] STAUDUHAR, R. P. The Determination of Galois Groups. *Math. Comp.* 27 (1973), 981–996.
- [25] VALIBOUZE, A. Computation of the Galois Group of the Resolvent Factors for the Direct and Inverse Galois Problem. In *Proceedings of AAEECC-11* (1995), vol. 948 of *LNCS*, Springer-Verlag.
- [26] YOKOYAMA, K. A Modular Method for Computing the Galois Group of Polynomials. *J. of Pure and Applied Algebra* (1996). To appear.