Block Systems of a Galois Group

Alexander Hulpke

CONTENTS

Supported by the Graduiertenkolleg "Analyse und Konstruktion in der Mathematik".

Keywords: block systems, ideal decomposition, subfields

AMS Subject Classification. Primary: 11Y40; Secondary: 20B40, 12F10, 12Y05, 12-04.

We describe an algorithm to compute subfields of an algebraic number field as block systems of its Galois group. It relies only on symbolic computations and avoids numerical approximations.

1. INTRODUCTION

We are given an irreducible polynomial $f \in K[x]$ (where K is perfect), with root α. We describe an algorithm for finding all subfields of $K(\alpha)$ that contain K.

By the main theorem of Galois theory, the subfields of $K(\alpha)$ correspond to the subgroups of the Galois group G of f that contain the stabilizer $\operatorname{Stab}_G \alpha$, and thus to the block systems of the operation of G on the cosets of $\operatorname{Stab}_G \alpha$. Let σ be a primitive element of a subfield S. We find polynomials $g, h \in K[x]$ such that $\sigma = h(\alpha)$ and $g(\sigma) = 0$. Accordingly, we have a "decomposition" $f | (g \circ h)$, sometimes called an ideal decomposition [Casperson et al.]; here $\circ$ denotes composition of functions.

Applications of this procedure include constructive Galois theory, the denesting of radical expressions [Zippel 1985], algebraic geometry [Lazard and Valibouze 1993], and the expression of the roots of solvable polynomials in terms of radicals [Landau and Miller 1985].

There have been other approaches to this problem [Casperson and McKay 1992; Casperson et al.; Dixon 1990; Landau 1993; Lazard and Valibouze 1993]. However, the algorithm presented here has some advantages over others in the literature. It avoids unnecessarily hard computations (even though its worst case complexity may be the same) by embedding the algebraic extension in an appropriate p-adic field and building blocks from

1058-6458/96 \$0.50 per page

stabilizer orbits as suggested in [Schönert and Seress 1994]. It also avoids numerical approximations and relies on exact algebraic computations only.

Section 6 below compares the performance of previous algorithms with the one introduced here. The comparison includes a polynomial reduction algorithm [Cohen and Diaz y Diaz 1991] that runs very fast in general and sometimes, but not always, yields subfields.

2. A THEOREM ABOUT BLOCKS

In the sequel, G will always denote a group operating transitively on a set Ω. A G-invariant partition $\mathcal{B} = \{B_1, \dots, B_m\}$ of Ω with $1 < m < |G|$ is called a (nontrivial) *block system* for G. Since G operates transitively, every block system $\mathcal{B}$ is already determined uniquely by one block $B \in \mathcal{B}$.

Our first aim is to give a different characterization for a partition $\mathcal{B}$ to be a block system, based on one set in $\mathcal{B}$. We show that orbits and coset representatives of a point stabilizer determine all block systems.

If $\mathcal{B}$ is a block system, $B \in \mathcal{B}$ and $\beta \in B$, then $\operatorname{Stab}_G \beta$ must fix B setwise. Therefore B is the union of orbits of $\operatorname{Stab}_G \beta$.

We shall construct all blocks B containing β by examining unions of orbits of $\operatorname{Stab}_G \beta$. If $\Gamma \subset \Omega$ is such an union of orbits, the following lemma gives a sufficient condition for Γ^G to be a block system. By the above remarks, this condition is necessary as well.

Lemma 2.1. *Let $\Gamma \subset \Omega$ and $\beta \in \Gamma$ be such that, for all $g \in G$, the condition $\beta^g \in \Gamma$ implies $\Gamma^g = \Gamma$; that is, $g \in \operatorname{Stab}_G \Gamma$, where*

$$\operatorname{Stab}_G \Gamma = \{g \in G \mid \Gamma^g = \Gamma\}$$

denotes the setwise stabilizer. Then the orbit of Γ under the operation of G is a block system for G.

Proof. We show first that we may replace β by an arbitrary point in Γ. Let $\gamma \in \Gamma$ and $g \in G$, with $\gamma^g \in \Gamma$. Since G operates transitively, there is an element $h \in G$ with $\beta^h = \gamma \in \Gamma$. We thus also have $\beta^{hg} \in \Gamma$. By the hypothesis this implies $h, hg \in \operatorname{Stab}_G \Gamma$, and accordingly also $g \in \operatorname{Stab}_G \Gamma$.

Now assume that there is $x \in G$ with $\Gamma \cap \Gamma^x \neq \varnothing$. Then there is $\gamma \in \Gamma$ with $\gamma^x \in \Gamma$. Accordingly we have $x \in \operatorname{Stab}_G \Gamma$ and $\Gamma^x = \Gamma$. This finally implies that two images Γ^x and Γ^y of Γ either intersect trivially or are identical. Therefore the set of images Γ^G is a G-invariant partition of Ω, that is, a block system for G. □

We now apply this lemma to the situation that interests us, using the fact that the Galois group of an irreducible polynomial operates transitively on its roots.

We denote the Galois group of f over K by G, its splitting field by L and the roots of f by $\alpha = \alpha_1, \alpha_2, \dots, \alpha_n$. By the Galois correspondence, the field $K(\alpha)$ corresponds to the point stabilizer $\operatorname{Stab}_G \alpha$. Each irreducible factor of f over $K(\alpha)$ is a polynomial whose roots form one orbit of $\operatorname{Stab}_G \alpha$. To search for blocks containing α, we form sets F of irreducible factors f_i of f over $K(\alpha)$, such that $x - \alpha \in F$, the latter obviously being a factor of f over $K(\alpha)$. We denote by B_F the set of roots of polynomials in F, and by G_F the set of automorphisms $\varphi \in G$ for which $\alpha^\varphi \in B_F$. Instead of applying the automorphisms $\varphi \in G$ to the roots α_{ij} of each $K(\alpha)$-irreducible polynomial f_i, we apply them (formally, their extension to the polynomial ring $L[x]$ by action on the coefficients) to the polynomials f_i. The situation of the preceding lemma then becomes:

Theorem 2.2. *Let F be a set of factors f_i of f over $K(\alpha)$, such that $x - \alpha \in F$. Then the corresponding root set B_F is a block of G if and only if, for every automorphism $\varphi \in G_F$ and every $f_i \in F$, the set of roots of f_i^φ remains in B_F.*

Proof. If B_F is a block, every automorphism $\varphi \in G_F$ is contained in the block stabilizer. Consequently, every f_i in F (which is a product of linear factors corresponding to roots in B_F) is mapped by φ to a product of linear factors corresponding to roots in B_F.

To see the converse, take $\varphi \in G_F$. By the assumption, φ maps each root in B_F to another root in B_F. Thus φ is in the stabilizer of B_F. By the preceding lemma, B_F is a block. $\square$

Remark. To apply an automorphism $\varphi \in G$ to one of the factors f_i whose coefficients are polynomials in α, we need to know only the image of α under φ, not the operation of G on the other roots of f.

If we regard $h(x, \alpha) = \prod_{f_i \in F} f_i$ as a polynomial in two indeterminates x and α, the condition given in the theorem is equivalent to

$$h(x, \alpha) - h(x, y) \equiv 0 \mod h(y, \alpha). \tag{2.1}$$

Testing for this condition involves computations in the polynomial ring $K(\alpha)[x, y]$. Experiments (Section 6) show that these computations tend to be significantly harder than those in our approach.

3. A MODULAR APPROACH

From now on, for concreteness, we will take $K = \mathbb{Q}$. We will use p-adic approximations to compute with the roots of f. Any approximation susceptible to computation could be used instead, so the same basic method is applicable to any field for which methods for factoring polynomials and approximating roots exist.

Theorem 3.1 [Tschebotareff 1925]. *Let $f \in \mathbb{Q}[x]$ be an irreducible polynomial with integer coefficients, and let p be a prime that divides neither the discriminant of f nor the leading coefficient of f. Then the distribution of the degrees of the irreducible factors of f modulo p corresponds to the cycle structure (the set of cycle lengths) of the action of an element of* $\operatorname{Gal} f$ *on the roots of f. If the frequency of these distributions is averaged over all primes $p \leq P$, this converges, as P tends to infinity, to the frequency of cycle structures in* $\operatorname{Gal} f$.

Since $\operatorname{Gal} f$ contains the identity, we can choose a prime p such that f splits into different linear factors modulo p (there are infinitely many such p). By Hensel's lemma, f splits over the corresponding p-adic field $\mathbb{Q}_p$ in linear factors corresponding to roots $\varrho_i \in \mathbb{Q}_p$. We thus obtain n different embeddings $\mu_i\colon \mathbb{Q}(\alpha) \to \mathbb{Q}_p$, with $\mu_i(\alpha) = \varrho_i$. Denoting the splitting field of f over $\mathbb{Q}$ by L, we may extend μ_1 to a mapping $\mu\colon L \to \mathbb{Q}_p$. Without loss of generality we may suppose that μ maps α_i to ϱ_i. If φ_i is an automorphism that maps α to α_i, we obtain the following commutative diagram:

$$\begin{tikzcd}
\mathbb{Q}(\alpha) \arrow[rr, "\varphi_i"] \arrow[dr, "\mu_i"'] & & \mathbb{Q}(\alpha_1) \arrow[dl, "\mu"] \\
& \mathbb{Q}_p &
\end{tikzcd}$$

For every polynomial $e \in \mathbb{Q}[x]$ we have $e(\alpha)^{\varphi_i \mu} = e(\alpha^{\varphi_i \mu}) = e(\alpha^{\mu_i})$. Application of automorphisms can thus be replaced by selection of the appropriate embedding μ_i.

To test whether the roots of the polynomials of a set F form a block, we take the roots of all the images F^μ. Taking the embeddings μ_i corresponding to these roots, we check whether the images F^{μ_i} preserve the roots. (This is exactly the criterion of the preceding section.)

We suppose also that f is an irreducible monic integer polynomial. Thus all its roots are algebraic integers, and so lie in the valuation ring $R = \mathbb{Z}_p$ (every valuation ring will contain the algebraic integer elements of its quotient field). Accordingly, the images f_i^μ (which are products of linear factors corresponding to the roots) are contained in $R[x]$. The test for blocks thus takes place in R.

Since we cannot compute exactly with p-adic numbers, we have to rely on approximations modulo a chosen prime power. Restricted to R, the approximation is a homomorphism. Since the test for blocks (checking whether a given set of numbers form the roots of given polynomials) consists only of ring operations, this test may take place just as well in the homomorphic image. The only condition this image has to fulfil is that roots can be distinguished. Since f is square-free modulo p, the coarsest approximation—computing in the field $\mathbb{F}_p$—is sufficient.

4. THE ALGORITHM

Using the ideas from the preceding sections, the computation proceeds as follows:

1. If f is not a monic integer polynomial, replace it by a monic polynomial defining the same field: for example, $a^{\deg f} f(x/a)$, with a a sufficiently large divisor of the lowest common multiple of the denominators of the coefficients of f.
2. Factorize f modulo different primes to check which block sizes are possible, as described in [Dixon 1990].
3. Take the quotient field $\mathbb{Q}[x]/(f)$, which is isomorphic to the extension of $\mathbb{Q}$ defined by a root of f. This root is represented by the coset $(f)+x$, which is denoted by α.
4. Factorize f over $\mathbb{Q}(\alpha)$ to obtain the factors f_i.
5. Select the combinations of the f_i that include $f_1 = x - \alpha$ such that the sum of the degrees corresponds to a possible nontrivial block size.
6. Search for a prime p that does not divide the discriminant of f and such that f splits into linear factors modulo p. This prime has probably already been found in step 2.
7. For each combination obtained in step 5, check whether it corresponds to a block, using the criterion from Section 3.
8. Finally, compute ideal decompositions $f|(g \circ h)$ for all blocks found (see below). In these decompositions, g is a minimal polynomial for the corresponding subfield. If f was modified in step 1, the reverse transformation $h(ax)$ has to be applied to h to obtain a decomposition of the original polynomial.

We now comment on the individual steps of the algorithm. Step 1 is trivial.

Possible block sizes (Step 2)

In step 2 we factor f modulo different primes to obtain cycle structures of elements in G. This will restrict the possible sizes of blocks. For example, if an n-cycle and an $(n-1)$-cycle are found, the group is doubly transitive and thus primitive, and we can stop. If we know the transitive permutation groups of appropriate degree, we can even, as already suggested in [Soicher and McKay 1985], use this knowledge for partial identification of the Galois group. Using this information we can check the list of groups for possible block system sizes.

We would like to obtain cycle structures of all elements in G this way. However, Tschebotareff's theorem only guarantees this in the limit, though [Lagarias and Odlyzko 1977] gives error estimates.

On the other hand, factoring modulo a prime is extremely fast. We thus factor modulo different primes, until we have found a prime modulo which f splits into linear factors. We would have to find a prime with this property in step 6 anyhow, so nothing is wasted. On the other hand, Tschebotareff's theorem (only the identity element has the corresponding cycle structure) asserts that this happens, on average, only every $|G|$ primes. Thus this is the earliest point where one might hope to have found all cycle structures.

Algebraic factorization (Step 4)

Factoring f over $\mathbb{Q}(\alpha)$ is by far the hardest part of the algorithm. Experiments show that the running time is completely determined by this factorization. A thorough discussion of the factoring process can be found in [Abbott 1989]; we will just comment briefly on the routines used.

For the actual factorization, basically three algorithms are known. The first [Trager 1976] factors the norm of the polynomial over the base field and takes gcd's with these factors. This is only feasible for comparatively small problems; a typical limit would be $\deg(\text{norm}) \leq 20$.

The algorithm of [Weinberger and Rothschild 1976] uses Hensel lifting as in the ordinary case. If the selected prime p has the property that the minimal polynomial of the extension splits over $\mathbb{F}_p$ (by Tschebotareff's theorem, this will happen for all primes in many cases of Galois groups with nontrivial block systems), factorizations corresponding to all the factors of the minimal polynomial must be lifted. They can be recombined to search for factors in characteristic zero by using the Chinese Remainder Theorem, but during this process all possible combinations of factors corresponding to different factors of the minimal polynomial have to be combined (if, for example, the minimal polynomial has two factors modulo p and yields two liftings with 5 factors of the same degree each,

$$\sum_{i=1}^{5} \binom{5}{i}^2 = 251$$

combinations have to be tested in total). This gives an even worse performance than the typical exponential factor search.

Lastly, [Lenstra 1982] tries to avoid these expensive recombinations. A final lattice reduction enables one to obtain algebraic numbers from a factorization corresponding to one factor of the minimal polynomial that had been lifted to a higher accuracy. (This just avoids the Chinese Remainder part. The usual combination of factors at the end of the lifting process could also be avoided by another lattice reduction, but in practice this is infeasible [Lenstra 1983].) In many cases this lattice reduction greatly speeds up the factoring process. However, if the coefficients and degrees involved become bigger, it seems that the lattice reduction and the additional lift to the higher accuracy are too expensive, and the method of [Weinberger and Rothschild 1976] should be used again.

Combining the factors (Step 7)

In general, the polynomial splits into few factors, and any naïve algorithm can be used to obtain the combinations.

The hardest case is when the Galois group is in its regular representation. Then the degrees of the f_i are all one and a maximal number of combinations has to be tested. In this case, however, we can obtain the Galois group from the factorization of f over $K(\alpha)$, where f must split into linear factors. The Galois group consists of all those mappings that map α to another root, extended to polynomials in α. As the image of α is again a polynomial in α, multiplication in this group can be computed. Thus one could even compute the corresponding permutation representation for which efficient block finding algorithms are available.

Computing the ideal decomposition (Step 8)

Without loss of generality, we can suppose that the set of factors $\{f_1, \ldots, f_s\}$ corresponds to a block. We form the product

$$e = \prod_{i=1}^{s} f_i.$$

Its coefficients are the elementary symmetric functions of the roots in the block and therefore invariant under all automorphisms in the block stabilizer. Thus e is a polynomial defined over the intermediate field L corresponding to the block stabilizer. Since α is a root of e and the degree of e is equal to the size of the block, e is not defined over any proper subfield of L and L is the field defined by the coefficients e_j of e. By the primitive element theorem there is a positive $\mathbb{Z}$-linear combination $\sum z_j e_j$ that serves as a primitive element for L. We will find this element by examining $\mathbb{Z}$-linear combinations of the e_j until a primitive element has been found.

If $\gamma = \sum z_j e_j$ is a linear combination, γ is a polynomial in α, say $\gamma = h(\alpha)$. We then compute the minimal polynomial g of γ by solving the system of linear equations

$$\{h^k \equiv 0 \pmod{f} \mid k = 0, \ldots\}.$$

If $\deg g \cdot \deg e = \deg f$, then γ defines the correct subfield, g is a minimal polynomial of this subfield, and we obtain in addition the ideal decomposition $f | (g \circ h)$. Otherwise we select another linear combination of the e_j.

In practice it appears that the first combination almost always defines the correct field already.

5. AN EXAMPLE

To illustrate this algorithm we apply it to the polynomial $f = x^8 + 4x^6 + 10x^4 + 12x^2 + 7$ (see entry 4 in Table 1). This group has two block systems, of sizes 2 and 4. Factorization f over $\mathbb{Q}(\alpha)$ yields the factors

$$\begin{aligned} f_1 &= x - \alpha, \\ f_2 &= x + \alpha, \\ f_3 &= x^2 - \alpha^6 - 3\alpha^4 - 6\alpha^2 - 3, \\ f_4 &= x^2 + \alpha^6 + 3\alpha^4 + 6\alpha^2 + 5, \\ f_5 &= x^2 + \alpha^2 + 2. \end{aligned}$$

The prime $p = 641$ fulfills all the necessary conditions. We will denote cosets modulo 641 by their representative.

We have $f(35) \equiv 0 \pmod p$. Taking $\alpha \equiv 35 \pmod p$ one obtains the zeros for the factors as follows:

f_1	f_2	f_3	f_4	f_5
35	606	295, 346	174, 467	170, 471

The combination $F_1 = \{f_1, f_2\}$ has roots $B_{F_1} = \{35, 606\}$. To check whether this set forms a block, we set $\alpha \equiv 606 \pmod p$. Then f_1 has root 606 and f_2 root 35, so B_{F_1} remains invariant. Therefore we have found a block.

To obtain the ideal decomposition we form the product $f_1 f_2 = x^2 - \alpha^2$. We take $h_1(\alpha) = \alpha^2$, from which the minimal polynomial $g_1 = x^4 + 4x^3 + 10x^2 + 12x + 7$ is obtained as described in the discussion of Step 8 in the previous section. (This decomposition could have been found trivially by looking at the polynomial.)

The combination $F_2 = \{f_1, f_2, f_3\}$ has roots

$$B_{F_2} = \{35, 606, 295, 346\}.$$

Taking $\alpha \equiv 295 \pmod p$ we get for f_3 a root 471, which is not in B_{F_2}. This combination cannot correspond to a block. Similarly, $F_3 = \{f_1, f_2, f_4\}$ includes the root $174 \in B_{F_3}$, but setting $\alpha \equiv 174 \pmod p$ yields for f_4 the root $471 \notin B_{F_3}$.

Finally, $F_4 = \{f_1, f_2, f_5\}$ has roots

$$B_{F_4} = \{35, 606, 471, 170\}.$$

Replacing α with each of these values fixes B_{F_4}; we have found another block. The product $f_1 f_2 f_5$ has only one coefficient that is not rational. It yields $h_4 = x^4 + 2x^2$ (as the block contains the one previously found, h_4 can be selected to be a polynomial on h_1). Accordingly, $g_4 = x^2 + 6x + 7$.

Any remaining combinations would be of the wrong degree, so we have found all block systems.

6. COMPARISON WITH OTHER ALGORITHMS

The major advantage of the algorithm introduced here is that the hard computation (the factorization) has to be performed only once, and testing a combination for being a block is extremely cheap afterwards. This should be an advantage over algorithms such as the ones in [Casperson et al.] or [Dixon 1990], where each test of a set of roots forming a block is expensive. The lattice reduction that these algorithms rely on to search for relations between the roots or minimal polynomials are, like the L^3-algorithm [Lenstra et al. 1982], of polynomial complexity. In practice, however, the matrix coefficients are quite big, and larger examples take a significant amount of time.

The Polred reduction algorithm [Cohen and Diaz y Diaz 1991], originally intended to compute small polynomials for number fields, can also find polynomials for subfields, but it is not guaranteed to find all subfields. As we obtain the decomposing polynomials via symmetric functions of the roots, however, they tend to have larger coefficients than the polynomials obtained by Polred. Moreover, Polred returns only polynomials for the subfields. The connection with the original polynomial—that is, the h in the decomposition $f|(g \circ h)$—has to be computed separately. As the algorithm might return several polynomials for the same field, this also implies that conjugate subfields cannot be recognized and distinguished without further tests.

As observed in [Lazard and Valibouze 1993], every subfield can be found via symmetric functions of the roots of the corresponding block. These authors suggest factoring of symmetric resolvents as a way to obtain all subfields. If the degree of f is n, this amounts to factoring polynomials of degree up to $\binom{n}{\lfloor n/2 \rfloor}$, with relatively large roots. While the

Key	
1	$x^6 + 108$
2	$x^8 - 12x^6 + 23x^4 - 12x^2 + 1$
3	$x^8 - 10x^4 + 1$
4	$x^8 + 4x^6 + 10x^4 + 12x^2 + 7$
5	$x^9 - 18x^8 + 117x^7 - 348x^6 + 396x^5 + 288x^4 + 3012x^3 + 576x^2 + 576x - 512$
6	$x^{10} + 38x^9 - 99x^8 + 1334x^7 - 4272x^6 + 9244x^5 - 8297x^4 + 1222x^3 + 1023x^2 - 74x + 1$
7	$x^{10} - 20x^9 + 80x^8 + 200x^7 - 3770x^6 + 872x^5 + 29080x^4 + 36280x^3 - 456615x^2 + 541260x - 517448$
8	$x^{10} - 10x^8 + 20x^7 + 235x^6 + 606x^5 + 800x^4 + 600x^3 + 270x^2 + 70x + 16$
9	$x^{12} + 6x^9 + 4x^8 + 8x^6 - 4x^5 - 12x^4 + 8x^3 - 8x + 8$
10	$x^{12} + 9x^{11} + 3x^{10} - 73x^9 - 177x^8 - 267x^7 - 315x^6 - 267x^5 - 177x^4 - 73x^3 + 3x^2 + 9x + 1$
11	see [Casperson et al.]
12	$x^{15} + 20x^{12} + 125x^{11} + 503x^{10} + 1650x^9 + 3430x^8 + 4690x^7 + 4335x^6 + 2904x^5 + 1400x^4 + 485x^3 + 100x^2 + 15x + 1$

					algorithm →	A	B		C		D		
	G	$\|G\|$	deg	blocks	shape	t	t_{res}	t_{fac}	t	#dec	t_{fac}	t_{quot}	t_{mod}
1	S_3	6	1^6	$2^3 3$	2/3	12	2/4	4/8	0.5	3	12	64	13
2	C_2^3	8	1^8	$2^7 4^7$	2/4	21	4/8	10/276	0.2	$2^2 4^3$	85	313	86
3	$2 \times D_4$	16	$1^4 2^2$	$2^3 4^3$	2/4	0.6	5/9	5/74	0.4	$2^3 4$	30	72	33
4	$[2^4]4$	64	$1^2 2^3$	2/4	2/4	518	5/759	7/1.9k	0.2	2/4	288	305	295
5	$3^2{:}2$	18	$1^2 2^4$	3^4	3	234	102	494	3	3^2	577	599	582
6	$2^4{:}5$	80	$1^2 2^4$	2	2/5	13k	10/735	$13/*$	4	none	578	811	600
7	$2^4{:}D_5$	160	$1^2 4^2$	2	2/5	2.9k	9/1.0k	$12/*$	9	none	2.3k	2.4k	2.3k
8	$(5^2{:}4){:}2$	200	1/4/5	5	5	9.7k	1.3k	$*$	3	5	356	391	381
9	$S_3 \times S_4$	144	$1^6 2^3$	3/4	3/4/6	2.4k	$837/7.1\mathrm{k}/*$	$387/13\mathrm{k}/*$	2	3/4	94	106	100
10	$3 \times D_4$	24	1/2/3/6	2/3/4/6	2/3/4/6	4.0k	$16/462/7.2\mathrm{k}/*$	$20/454/*/*$	9	4	500	1.8k	507
11	A_4	12	1^{12}	$2^3 3^4 4$	2/3/4/6	$**$	$51/2.1\mathrm{k}/55\mathrm{k}/*$	$126/25\mathrm{k}/*/*$	550	$2/3^3$	17k	$**$	17k
12	$(5^2{:}4)S_3$	600	1/4/10	5	5	$**$	$*/*/*/*$	$*/*/*/*$	65	none	10k	10k	10k

TABLE 1. Example polynomials with imprimitive Galois group, and running times of several algorithms. For each polynomial we give information on the Galois group, the lengths of the point stabilizer orbits ("deg"), the block sizes of actual block systems of the Galois group ("blocks"), and possible block sizes after application of Step 2 of the algorithm of Section 4 ("shape"). This information on possible block sizes was given to the algorithms when possible. Exponents indicate the existence of several block systems with the same block size. Polynomial 1 is taken from [Soicher and McKay 1985], 2–4 from [Mattman and McKay], 5 from [Geyer 1993], 9 from [Dixon 1990], 10 from [Lazard and Valibouze 1993], and 11 deduced from a polynomial in [Ash et al. 1991].

Algorithm A is an implementation of [Casperson et al.] in Maple [Char et al. 1991]; it only searches for one decomposition, and finishes very fast for decompositions of the form $f(x) = g(x^m)$.

Algorithm B is an implementation of [Lazard and Valibouze 1993] in GAP [Schönert et al. 1994]; t_{res} is the time needed to find the resultant, and t_{fac} is the time spent in polynomial factorization (which we list separately to indicate potential improvements arising from better factoring routines). Multiple times correspond to the possibilities for block sizes.

Algorithm C is an implementation of [Cohen and Diaz y Diaz 1991] in Pari-GP [Batut et al. 1993], version 1.38.71 (which performs significantly better than older versions for this purpose). It is not guaranteed to find all decompositions; we give the number found in each case.

Algorithm D is the one described in this paper, implemented in GAP 3.4; t_{fac} is the time spent in factorization, t_{mod} is total time spent by the full modular algorithm, and t_{quot} total time spent for the quotient test in (2.1).

Running times are in seconds; "k" stands for 10^3. Stars $*$ indicate that computation is not possible in reasonable time (in the case of $**$, the run was aborted after 50k seconds). All runs took place on an HP730 workstation, and the programs were given sufficient memory to minimize the influence of garbage collection.

factorization time could be reduced significantly by searching only for factors of appropriate degree, even the computation of those polynomials is infeasible for larger degrees.

The algorithm of Landau and Miller [Landau 1993; Landau and Miller 1985] also exploits factorization over the point stabilizer, but uses multiple algebraic extensions, which tend to be computationally hard. From the point of view of computational group theory, this algorithm also relies on Atkinson's block algorithm [Atkinson 1975], while our approach resembles newer developments like [Schönert and Seress 1994], which build blocks from stabilizer orbits.

For illustration, we applied some of the algorithms mentioned (as far as implementations were available) to a number of test polynomials. The results are shown in Table 1, where a key to the polynomials and to the algorithm codes A, B, C, D is given.

The use of different underlying languages renders a comparison difficult, but was unavoidable in view of the need for specific routines available only under one or another system. Algorithms A and C were used in their inventors' implementation, so it is hoped that they were treated fairly. Implementations of B and D are due to the author.

The following observations can be made:

- Algorithm A runs quite fast for smaller cases, but becomes significantly slower if the degree is larger than 11.
- Algorithm B performs reasonably well for small degree n. However, since a search for blocks of size m involves construction and factorization of a polynomial of degree $d = \binom{n}{m}$, the running time increases significantly if n becomes bigger then 11 or m bigger than 3. Also, algorithm B would benefit from a special factoring algorithm to search only for factors of given size. A routine of this kind was not available.
- Algorithm C is by far the fastest but usually fails to give some (and sometimes all) of the decompositions.
- The experiments show that the running time of the modular version of algorithm D is dominated completely by factorization, even when the polynomial splits completely into linear factors. Also, usage of the modular method turns out to be significantly faster than the quotient test if a lot of combinations have to be checked.

ACKNOWLEDGEMENTS

I would like to thank the staff of Lehrstuhl D for many discussions which led to major improvements of this paper and John McKay for the invitation to spend half a year at Concordia University in Montréal; a visit which introduced me to computational Galois theory. The facilities and the support provided by CICMA at Concordia University are acknowledged with thanks.

REFERENCES

[Abbott 1989] J. A. Abbott, "On the factorization of polynomials over algebraic fields", Ph.D. thesis, University of Bath, 1989.

[Ash et al. 1991] A. Ash, R. Pinch, and R. Taylor, "An $\hat{A}_4$ extension of $\mathbb{Q}$ attached to a non-self-dual automorphic form on GL(3)", *Math. Annalen* **291** (1991), 753–766.

[Atkinson 1975] M. Atkinson, "An algorithm for finding the blocks of a permutation group", *Math. Comp.* (1975), 911–913.

[Batut et al. 1993] C. Batut, D. Bernardi, H. Cohen and M. Olivier, *User's Guide to Pari-GP*. This manual is part of the program distribution, available by anonymous ftp from the host megrez.ceremab.u-bordeaux.fr.

[Casperson and McKay 1992] David Casperson and John McKay, "An ideal decomposition algorithm", *Abstracts Amer. Math. Soc.* **13** (1992), 405.

[Casperson et al.] David Casperson, David Ford, and John McKay, "An ideal decomposition algorithm", submitted to *J. Symb. Comp.*

[Char et al. 1991] B. W. Char et al., *Maple V Language Reference Manual* and *Maple V Library Reference Manual*, Springer, New York, 1991.

[Cohen and Diaz y Diaz 1991] Henri Cohen and F. Diaz y Diaz, "A polynomial reduction algorithm", *Séminaire de théorie des nombres de Bordeaux* (sér. 2) **3** (1991), 351–360.

[Dixon 1990] John D. Dixon, "Computing subfields in algebraic number fields", *J. Australian Math. Soc.* (series A) **49** (1990), 434–448.

[Geyer 1993] Helmut Geyer, "Programme zur Berechnung der Galoisgruppen von Polynomen 8. und 9. Grades", Preprint 93-10, IWR Heidelberg, 1993.

[Landau 1993] Susan Landau, "Finding maximal subfields", *SIGSAM Bull.* **27** (1993), no. 3, 4–8.

[Lenstra 1982] A. K. Lenstra, "Lattices and factorization of polynomials over algebraic number fields", pp. 32–39 in *Computer algebra: EUROCAM '82* (edited by J. Calmet), Lecture Notes in Computer Science, **144**, Springer, Heidelberg, 1982.

[Lenstra 1983] A. K. Lenstra, "Factoring polynomials over algebraic number fields", pp. 245–254 in *Computer algebra: EUROCAL '83* (edited by J. A. van Hulzen), Lecture Notes in Computer Science, **162**, Springer, Heidelberg, 1983.

[Lenstra et al. 1982] A.K. Lenstra, H.W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients", *Math. Annalen* **261** (1982), 515–534.

[Landau and Miller 1985] Susan Landau and Garry Miller, "Solvability by radicals is in polynomial time", *J. Comp. Sys. Sci.* **30** (1985), 179–208.

[Lagarias and Odlyzko 1977] J. C. Lagarias and Andrew M. Odlyzko, "Effective versions of the Chebotarev density theorem", pp. 409–464 in *Algebraic Number Fields (L-functions and Galois properties)* (edited by A. Fröhlich), Academic Press, London, 1977.

[Lazard and Valibouze 1993] Daniel Lazard and Annick Valibouze, "Computing subfields: Reverse of the primitive element problem", pp. 163–176 in *Computational Algebraic Geometry* (edited by F. Eyssette and A. Galligo), Progress in Mathematics **109**, Birkhäuser, Boston, 1993.

[Mattman and McKay] T. W. Mattman and J. McKay, "Computation of Galois groups over function fields", to appear in *Exp. Math.*

[Schönert et al. 1994] M. Schönert et al., *GAP: Groups, Algorithms, and Programming*, Lehrstuhl D für Mathematik, RWTH Aachen, Germany, 1994. Available by anonymous ftp, together with the GAP system, on the servers ftp.mth.pdx.edu, archives.math.utk.edu, or math.rwth-aachen.de.

[Schönert and Seress 1994] Martin Schönert and Ákos Seress, "Finding blocks of imprimitivity in small base groups in nearly linear time", submitted to Proc. ISSAC '94, 1994.

[Soicher and McKay 1985] Leonard H. Soicher and John McKay, "Computing Galois groups over the rationals", *J. Number Theory* **20** (1985), 273–281.

[Trager 1976] Barry M. Trager, "Algebraic factoring and rational function integration", pp. 219–226 in *Proceedings of the 1976 ACM Symposium on symbolic and algebraic computation* (edited by R. D. Jenks), ACM, New York, 1976.

[Tschebotareff 1925] Nikolaj Tschebotareff, "Die Bestimmung der Dichtigkeit einer Menge von Primzahlen, welche zu einer gegebenen Substitutionsklasse gehören", *Math. Annalen* **95** (1925), 191–228.

[Weinberger and Rothschild 1976] P. J. Weinberger and L. P. Rothschild, "Factoring polynomials over algebraic number fields", *ACM Trans. Math. Software* **2** (1976), no. 4, 335–350.

[Zippel 1985] R. Zippel, "Simplification of expressions involving radicals", *J. Symb. Comp.* **1** (1985), 189–210.

Alexander Hulpke, Lehrstuhl D für Mathematik, RWTH, 52056 Aachen, Germany (Alexander.Hulpke@math.rwth-aachen.de)

Received July 4, 1994; accepted in revised form April 3, 1995