Computing with Subgroups of Automorphism Groups of Finite Groups

Charles C. Sims
Department of Mathematics
Rutgers University
New Brunswick, NJ 08903, USA
sims@math.rutgers.edu

Abstract

Let G be a finite group which possesses a chain of characteristic subgroups with quotients of moderate size. This paper describes methods for computing the order of and for deciding membership in a subgroup of the automorphism group of G. An application to the study of the Burnside group $B(2,5)$ is discussed.

1 Introduction

Techniques for determining the order of the automorphism group $\mathrm{Aut}(G)$ of a finite group G are known and have been implemented in large software systems such as GAP [5] and Magma [1] and in specialized packages such as the ANU p-group package [4]. In some cases it is even possible to obtain generators for $\mathrm{Aut}(G)$. What has not been readily available are techniques for solving the following problems: Let A be a subset of $\mathrm{Aut}(G)$ and let $\mathcal{H}$ be the subgroup of $\mathrm{Aut}(G)$ generated by A. Determine the order of $\mathcal{H}$ and decide, for a given automorphism α of G, whether α belongs to $\mathcal{H}$.

The need to solve the first of these problems arose in connection with the study of $B = B(2,5)$, the two-generator Burnside group of exponent 5 and its largest finite quotient $R = R(2,5)$, which has order 5^{34}. The order of $\mathrm{Aut}(R)$ is easily seen to be

$$|\mathrm{GL}(2,5)|\, 5^{64} = 2^5 \cdot 3 \cdot 5^{65},$$

or about 2.6×10^{47}. There is an obvious homomorphism from $\mathrm{Aut}(B)$ to $\mathrm{Aut}(R)$. Some automorphisms of B are known and it was of interest to determine the image in $\mathrm{Aut}(R)$ of the known part of $\mathrm{Aut}(B)$.

The focus of this paper is on algorithms which are "practical". Practicality is a somewhat ill-defined, subjective concept which means roughly that the results obtainable from an implementation with current technology can be expected to be worth the effort required to write the program.

Readers desiring background information about computing in finite groups can consult [6] for permutation groups and [7] for p-groups.

2 The general approach

There are practical algorithms for computing with subgroups of a finite group G whenever one has a chain of subgroups

$$G = G_1 \supseteq G_2 \supseteq \ \ldots\ \supseteq G_k \supseteq G_{k+1} = 1$$

such that for $1 \le i \le k$ the index $|G_i\colon G_{i+1}|$ is not too big and the cosets of G_{i+1} in G_i are easy to work with. The meaning of "not too big" is time dependent, but "not exceeding 10^5" is probably a reasonable definition for the present time. The phrase "easy to work with" will be made more precise below.

The most common examples of algorithms based on subgroup chains are techniques for working with permutation groups on a finite set Ω and methods for studying finite solvable groups. The literature on these algorithms goes back more than 30 years to papers such as [6] and [3].

Let G be a finite group for which we have practical algorithms to compute with elements. Thus we have concise descriptions of elements of G, we can decide when two descriptions refer to the same element, and given descriptions for elements g and h, we can compute descriptions for gh and g^{-1}.

Suppose in addition that we have a chain of subgroups G_i, $1 \le i \le k+1$ as above, and for $1 \le i \le k$ we know a set $\mathcal{U}_i$ of right coset representatives for G_{i+1} in G_i such that the following conditions are satisfied:

1. 1 is in $\mathcal{U}_i$ for each i.
2. Given a description of an element g of G and an integer i, we can decide whether g is in G_i.
3. If g is in G_i, then we can determine the element of $\mathcal{U}_i$ representing the coset $G_{i+1}g$.

Each element g of G can be represented uniquely as $u_k u_{k-1} \cdots u_1$, where for $1 \le i \le k$ the element u_i is in $\mathcal{U}_i$. If $u_i \ne 1$ and $u_j = 1$ for $1 \le j < i$, then we will call u_i the *leading representative* of g and i the *depth* of g.

From the existence of the chain of subgroups, we can deduce a crude upper bound for the number of elements needed to generate subgroups of G.

Proposition 1 *If H is a subgroup of G, then H can be generated by a set having at most*

$$\sum_{i=1}^{k} |\mathcal{U}_i| - 1$$

elements.

Proof. Let X be a generating set for H. We may assume that 1 is not in X. Suppose there are two distinct elements g and h of X with the same depth i and the same leading representative u in $\mathcal{U}_i$. Thus $g = u_k \cdots u_i$ and $h = v_k \cdots v_i$, where for $i \leq j \leq k$ both u_j and v_j are in $\mathcal{U}_j$ and $u_i = v_i = u$. Let us assume that i is as small as possible. Now

$$gh^{-1} = u_k \cdots u_i (v_k \cdots v_i)^{-1} = u_k \cdots u_i v_i^{-1} \cdots v_k^{-1} = u_k \cdots u_{i+1} v_{i+1}^{-1} \cdots v_k^{-1}$$

is in G_{i+1}. Replace h in X by gh^{-1}. We still have a generating set for H and by iterating this process we arrive at a generating set in which no two distinct elements have the same depth and the same leading representative. □

Given any subgroup H of G, we can obtain a chain of subgroups for H by setting $H_i = H \cap G_i$, $1 \leq i \leq k+1$. Suppose we have sets $\mathcal{V}_i$ of right coset representatives for H_{i+1} in H_i and assume that $\mathcal{V}_i \subseteq \mathcal{U}_i$. Then we can decide membership in H using a procedure group theorists call stripping and computer scientists call sifting.

```
Procedure STRIP(g)
   (* g is an element of G. *)
   h := g;

   For i from 1 to k do
      Let u be the element in U_i representing G_{i+1}h;
      If u ∉ V_i then break fi;
      h := hu^{-1};
   od;

   Return(h)
end;
```

Proposition 2 *An element g of G is in H if and only if* STRIP(g) *is the identity element.*

Let X be a subset of G and set $H = \langle X \rangle$, the subgroup generated by X. To find the order of H and to be able to decide membership in H, it suffices to solve the following problems:

1. Find generating sets for each of the subgroups H_i, $1 \leq i \leq k$.
2. Construct sets $\mathcal{V}_i$.

If we can solve problem (1), then solving (2) is easy using the following procedure:

```
Procedure REPS(Y, i)
   (* Y is a subset of G_i. Return a set of right
      coset representatives for H ∩ G_{i+1} in H,
      where H = ⟨Y⟩. *)
   r := 1;  v_1 := id;  j := 1;

   While j ≤ r do
      For y ∈ Y do
         w := v_j y;
         If w ∈ G_{i+1} v_t for some t with 1 ≤ t ≤ r
            then continue fi;
         r := r + 1;  v_r := w
      od;

      j := j + 1
   od;

   Return({v_1, ..., v_r})
end;
```

The main tool used to solve (1) is a fundamental result of Schreier.

Proposition 3 *Let H be a subgroup of a group G, let X be a generating set for G, and let $\mathcal{U}$ be a set of right coset representatives for H in G. For g in G, let $\overline{g}$ be the unique element of $(Hg) \cap \mathcal{U}$. Then H is generated by*

$$\{ux(\overline{ux})^{-1} \mid u \in \mathcal{U}, x \in X\}.$$

The elements $ux(\overline{ux})^{-1}$ are called the *Schreier generators* for H relative to $\mathcal{U}$ and X.

We can now solve both problems (1) and (2).

```
Procedure SUBGROUP(X)
   Y_1 := X;

   For i from 1 to k do
      V_i := REPS(Y_i, i);
      Let Z be the set of Schreier generators for ⟨Y_i⟩ ∩ G_{i+1}
         relative to V_i and Y_i;
      Use the algorithm of Proposition 1 (or refinements
         of it) to obtain a set Z_1 of moderate size such
         that ⟨Z_1⟩ = ⟨Z⟩;
      Y_{i+1} := Z_1
   od;

   Return (Y_1, ..., Y_k) and (V_1, ..., V_k)
end;
```

3 Automorphism groups

Let G be a finite group in which we can compute with elements and subgroups using the ideas of Section 2. The automorphism group $\mathrm{Aut}(G)$ may be much larger than G and it may not be possible to find a reasonable subgroup chain in $\mathrm{Aut}(G)$. For example, if G is an elementary abelian group of order p^d for some prime p, then $\mathrm{Aut}(G)$ is the general linear group $\mathrm{GL}(d,p)$ of nonsingular d-by-d matrices over the field with p elements. In G there is a subgroup chain in which the indices $|G_i : G_{i+1}|$ are all p, but typically for $\mathrm{GL}(d,p)$ some index in a subgroup chain is at least

$$1 + p + \cdots + p^{d-1} = \frac{p^d - 1}{p - 1}.$$

The example of elementary abelian groups shows that to get a nice subgroup chain for $\mathrm{Aut}(G)$ it is not sufficient for G to have small composition factors. However, there is a useful subgroup chain for $\mathrm{Aut}(G)$ if G has a chain of characteristic subgroups for which the indices are not too big.

Let G be generated by some set $X = \{x_1, \ldots, x_d\}$. An automorphism α of G is determined by the images x_i^α, $1 \leq i \leq d$. Thus

$$|\mathrm{Aut}(G)| \leq |G|^d.$$

We will be given a set Y of automorphisms of G. Thus we will not need to be able to decide whether a given d-tuple of elements of G defines an automorphism. However, we will need to be able to multiply (that it, compose) automorphisms, invert automorphisms, and compute the image of an element of G under an automorphism. To do these things, it is enough to be able to describe elements of G by words over the set X. The techniques of Section 2 make this possible.

Now let

$$G = S_1 \supseteq S_2 \supseteq \ldots \supseteq S_n \supseteq S_{n+1} = 1$$

be a chain of characteristic subgroups of G. For $1 \leq i \leq n+1$ let $\mathcal{A}_i$ be the subgroup of $\mathrm{Aut}(G)$ consisting of those automorphisms which act trivially on G/S_i. The automorphism of G/S_i induced by an element α of $\mathrm{Aut}(G)$ is determined by the cosets $x_j^\alpha S_i$. If α is in $\mathcal{A}_i$ then the coset $x_j^\alpha S_{i+1}$ is a subset of $x_j S_i$. Thus the index of $\mathcal{A}_{i+1}$ in $\mathcal{A}_i$ is at most $|S_i\colon S_{i+1}|^d$. This means that the indices in the chain

$$\mathrm{Aut}(G) = \mathcal{A}_1 \supseteq \mathcal{A}_2 \supseteq \ldots \supseteq \mathcal{A}_{n+1} = 1$$

are probably bigger than we want. Thus we need to refine this chain further.

For $1 \leq i \leq n$ and $1 \leq j \leq d+1$ let $\mathcal{A}_{ij}$ be the subgroup of $\mathcal{A}_i$ consisting of those elements which fix the cosets $x_1 S_{i+1}$, $\ldots$, $x_{j-1} S_{i+1}$. Then

$$\mathcal{A}_i = \mathcal{A}_{i1} \supseteq \mathcal{A}_{i2} \supseteq \ldots \supseteq \mathcal{A}_{id} \supseteq \mathcal{A}_{id+1} = \mathcal{A}_{i+1}.$$

The argument in the preceding paragraph shows that the index $|\mathcal{A}_{ij}\colon \mathcal{A}_{ij+1}|$ is at most $|S_i\colon S_{i+1}|$. Our assumption that the quotients S_i/S_{i+1} are not too large means that we have a subgroup chain in $\mathrm{Aut}(G)$ for which the indices are also not too large.

To apply the methods of Section 2 to this chain of subgroups in $\mathrm{Aut}(G)$, we must be able to decide membership in the subgroups $\mathcal{A}_{ij}$ and to work with the right cosets of $\mathcal{A}_{ij+1}$ in $\mathcal{A}_{ij}$. Both of these problems reduce to working with the groups S_i and their cosets. An automorphism α of G is in $\mathcal{A}_i$ if and only if x_r^α is in $x_r S_i$ for $1 \leq r \leq d$. If this is the case, then α is in $\mathcal{A}_{ij}$ if and only if x_r^α is in $x_r S_{i+1}$ for $1 \leq r < j$. Since we can decide membership in subgroups of G, we can decide membership in cosets.

Here is a simple example to illustrate the data structures involved in this algorithm. Let P be the group generated by elements a, b, and c subject to the defining relations

$$a^5 = b^5 = c^5 = 1, \quad ba = abc, \quad ca = ac, \quad cb = bc.$$

The order of P is 125 and the subgroup Z generated by c is both the center and the commutator subgroup of P. Thus Z is characteristic in P. The elements of P have unique representations of the form $a^i b^j c^k$, where the integers i, j, and k are between 0 and 4. The order of $\mathrm{Aut}(P)$ is 12000.

An automorphism α of P will be represented by the triple $\|a^\alpha, b^\alpha, c^\alpha\|$. A triple of elements $\|u, v, w\|$ represents an automorphism if and only $v^{-1}u^{-1}vu = w \neq 1$. Let $\mathcal{H}$ be the subgroup of $\mathrm{Aut}(P)$ generated by the following four automorphisms of P:

$$\alpha = \|ab, b, c\|, \quad \beta = \|a, b^2c, c^2\|, \quad \gamma = \|ac^3, b, c\|,$$

$$\delta = \|a, bc^2, c\|.$$

Let us describe $\mathcal{H}$ using the chain of characteristic subgroups $P \supset Z \supset 1$.

The first step is to determine the orbit of the coset aZ under $\mathcal{H}_{11} = \mathcal{H}$. The 25 cosets of Z in P have representatives $a^i b^j$. The orbit of aZ under $\mathcal{H}$ is described by the following table. Here the representatives for the cosets in

coset	aut
a	id
ab	α
ab^2	α^2
ab^3	α^3
ab^4	α^4

the orbit are listed and for each of these cosets uZ an automorphism σ is listed such that $(aZ)^\sigma = uZ$. The identity automorphism is denoted "id".

It turns out that the subgroup $\mathcal{H}_{12}$ of $\mathcal{H}_{11}$ fixing aZ is generated by β, γ, and δ. (This fact and similar statements can be verified by computing Schreier generators as illustrated below.) We next need to find the orbit of bZ under $\mathcal{H}_{12}$. This orbit has the four elements shown in the table

coset	aut
b	id
b^2	β
b^3	β^3
b^4	β^2

The subgroup $\mathcal{H}_{21}$ of $\mathcal{H}_{12}$ fixing bZ is generated by γ and δ. We now look at the action of $\mathcal{H}_{21}$ on the cosets of the trivial subgroup. These cosets can be identified with the elements of P. We start by finding the orbit of a under $\mathcal{H}_{21}$. This orbit has five elements:

coset	aut
a	id
ac	γ^2
ac^2	γ^4
ac^3	γ
ac^4	γ^3

The subgroup $\mathcal{H}_{22}$ of $\mathcal{H}_{21}$ fixing a is generated by δ. The orbit of b under this subgroup also has five elements.

coset	aut
b	id
bc	δ^3
bc^2	δ
bc^3	δ^4
bc^4	δ^2

We can confirm the assertions made about generators for the subgroups $\mathcal{H}_{12}$, $\mathcal{H}_{21}$, and $\mathcal{H}_{22}$ by computing the Schreier generators for each of these subgroups and stripping. Here is one example. The automorphism α is a coset representative for $\mathcal{H}_{12}$ in $\mathcal{H}_{11}$ and β is a generator of $\mathcal{H}_{11}$. The product $\alpha\beta$ takes aZ to ab^2Z. The corresponding Schreier generator is

$$\tau = \alpha\beta\alpha^{-2} = \|ac, b^2c, c^2\|.$$

To strip τ to see if it is in the subgroup generated by β, γ, and δ we first find the image of bZ under τ. This image is b^2Z. Thus

$$\tau\beta^{-1} = \|ac^3, b, c\|$$

fixes bZ. The image of a under $\tau\beta^{-1}$ is ac^3 and the product

$$\tau\beta^{-1}\gamma^{-1} = \|a, b, c\|$$

is the identity. Therefore $\tau = \gamma\beta$ is in the subgroup generated by β, γ, and δ. (In general the stripping procedure would have involved one more iteration, but we were lucky.)

The order of $\mathcal{H}$ is the product of the four orbit lengths, that is, $5 \cdot 4 \cdot 5 \cdot 5 = 500$.

The following observation is useful for implementation. Suppose that $i > 1$. As described, a coset of $\mathcal{A}_{ij+1}$ in $\mathcal{A}_{ij}$

corresponds to a coset yS_{i+1} in x_jS_i. Now yS_{i+1} is contained in x_jS_i if and only if $x_j^{-1}yS_{i+1}$ is contained in S_i. This gives a bijection between the set of cosets of $\mathcal{A}_{ij+1}$ in $\mathcal{A}_{ij}$ and a subset of S_i/S_{i+1}, and this quotient does not depend on j.

4 An implementation

An implementation of the algorithm sketched in the previous section has been written using the system GAP. Because of the intended application to subgroups of $\mathrm{Aut}(R(2,5))$, the implementation assumes that it has been given a set A of automorphisms of a p-group P and that P is given as a GAP "AG-group" in the format returned by the ANU p-quotient program. This means that the minimum number d of generators of P is known and P is generated by the first d elements of the sequence $x_1, \ldots, x_r$ of AG-generators defining a composition series for P. The implementation also assumes that each of the characteristic subgroups S_i is generated by the x_j with $j \geq m_i$ for some integer m_i.

Initially automorphisms of P were represented by objects created by the GAP function GroupHomomorphismByImages. However, those objects do not permit fast composition and inversion. They were replaced by objects developed by Bettina Eick, who also provided GAP functions to manipulate these objects. The assistance of Dr. Eick is acknowledged with deep appreciation.

Let $R = R(2,5)$. The nilpotence class of R is 12. For $1 \leq i \leq 13$ let $S_i = \gamma_i(R)$, the i-th term in the lower central series for R. The order of the quotient S_i/S_{i+1} is 5^{e_i}, where the sequence $e_1, \ldots, e_{12}$ is 2, 1, 2, 3, 2, 4, 4, 4, 6, 3, 2, 1. Thus the largest quotient is S_9/S_{10}, which has order $5^6 = 15625$, well within our bound of 10^5.

Let F be a free group of rank 2 with generators a and b. We may consider R and $B = B(2,5)$ to be quotient groups of F. If G denotes either B or one of the quotients R/S_i, then G is relatively free and hence any automorphism of F determines an automorphism of G. This gives a homomorphism from $\mathrm{Aut}(F)$ to $\mathrm{Aut}(G)$.

By the finiteness and relative freeness of R, any two pairs of generators for R are equivalent under $\mathrm{Aut}(R)$. Since R is a p-group, two elements of R generate R if and only if their images generate the Frattini quotient R/S_2 of R, which is a two-dimensional vector space over the field with 5 elements. The number of bases of R/S_2 is $|\mathrm{GL}(2,5)| = 480$ and each basis of R/S_2 can be lifted to a pair of generating elements of R in $(5^{32})^2 = 5^{64}$ ways. This gives the order of $\mathrm{Aut}(R)$ stated earlier.

By Section II.4 of [2], $\mathrm{Aut}(F)$ is generated by the four automorphisms α_a, α_b, β_{ab}, and β_{ba}, defined as follows:

$$a^{\alpha_a} = a^{-1}, \quad b^{\alpha_a} = b, \quad a^{\alpha_b} = a, \quad b^{\alpha_b} = b^{-1},$$

$$a^{\beta_{ab}} = ab, \quad b^{\beta_{ab}} = b, \quad a^{\beta_{ba}} = a, \quad b^{\beta_{ba}} = ba.$$

Let x and y denote the images in B of a and b, respectively. There is an automorphism γ of B which fixes x and maps y to y^2. All automorphisms of B currently known lie in the subgroup $\mathcal{H}$ generated by γ and the image of $\mathrm{Aut}(F)$.

The GAP procedure was used to determine the order of the image $\mathcal{K}$ of $\mathcal{H}$ in $\mathrm{Aut}(R/S_i)$ for various values of i. For small i the computations are not very difficult. Here are some examples.

The order of $\mathrm{Aut}(R/S_4)$ is $2^5 \cdot 3 \cdot 5^7$ and $\mathcal{K}$ is equal to $\mathrm{Aut}(R/S_4)$. It took about 16 seconds to compute the order of $\mathcal{K}$ on a Sun Sparc 20.

The order of $\mathrm{Aut}(R/S_5)$ is $2^5 \cdot 3 \cdot 5^{13}$ and $\mathcal{K}$ has index 625 in $\mathrm{Aut}(R/S_5)$. It took about 37 seconds on the Sparc 20 to find $|\mathcal{K}|$.

The order of $\mathrm{Aut}(R/S_9)$ is $2^5 \cdot 3 \cdot 5^{41}$. The index of $\mathcal{K}$ in $\mathrm{Aut}(R/S_9)$ is 5^{16}. Here the time on the Sparc 20 was about 3308 seconds.

When $i > 9$, the time to determine the order of $\mathcal{K}$ becomes quite substantial since then the quotient S_9/S_{10} is involved. In the case of greatest interest, that of $\mathrm{Aut}(R)$, the computation was switched to a somewhat faster Silicon Graphics workstation. On that machine it took nearly two days to conclude that $\mathcal{K}$ has index 5^{20} in $\mathrm{Aut}(R)$.

The implementation described here is currently available by anonymous ftp from `dimacs.rutgers.edu` in the directory `pub/rkbp`. The file `pgaut.g` contains the implementation of the automorphism group algorithm for p-groups. The file `example.g` defines a function `image` with which the reader can verify the statements about the index of $\mathcal{K}$ in the groups $\mathrm{Aut}(R/S_i)$. The function `image` assumes that GAP has access to the ANU p-quotient program.

It seems quite likely that substantial improvements in the times quoted here are possible. Even without such improvements, the GAP implementation has clearly demonstrated the practicality of the algorithm.

References

[1] Bosma, W., Cannon, J. J., and Playoust, C. *The Magma algebra system.* School of Mathematics and Statistics, University of Sydney, 1996.

[2] Lyndon, R. C., and Schupp, P. E. *Combinatorial Group Theory.* Springer Verlag, 1977.

[3] Neubüser, J. Untersuchungen des Untergruppenverbandes endlicher Gruppen auf einer programmgesteurten elektronischen Dualmaschine. *Numer. Math. 2* (1960), 280–292.

[4] Newman, M. F., and O'Brien, E. A. Application of computers to questions like those of Burnside, II. To appear in *Internat. J. Algebra Comput.*

[5] Schönert, M. *GAP: Groups, Algorithms and Programming.* Rheinisch Westfälische Technische Hochschule, Aachen, 1994.

[6] Sims, C. C. Computational methods in the study of permutation groups. In *Computational Problems in Abstract Algebra* (1970), J. Leech, Ed., Pergamon Press, pp. 169–183.

[7] Sims, C. C. *Computation with Finitely Presented Groups.* Cambridge University Press, 1994.