

## Computing automorphisms of finite soluble groups

MICHAEL J. SMITH

In this thesis an algorithm for computing the automorphism group of a finite soluble group is described. Recent work by Leedham-Green (described in [2]) has resulted in the development of *special power conjugate presentations* for representing finite soluble groups. These presentations provide many advantages for computing information about soluble groups, and algorithms for computing them are available in both the computational algebra systems GAP [5] and Magma [1]. The automorphism group algorithm takes a group given by a special power conjugate presentation and computes a generating set for the automorphism group as well as the order of the automorphism group. The effectiveness of the algorithm is demonstrated by performance results for a prototype implementation.

A more general problem than that required for computing automorphism groups of finite soluble groups is considered in Chapter 3. This problem involves computing a generating set for the automorphism group of an extension of an elementary abelian  $p$ -group by a finitely presented group. The solution of this problem divides naturally into a number of separate computations (an exact sequence given by Robinson [3] is the basis for the separation). The simplest of these is shown to correspond to computing a basis for a vector space of functions from the finitely presented group into the elementary abelian group. This part of the computation is described in the remainder of the chapter.

The remainder of the computation is more easily performed in the case of a split extension. Chapter 4 describes the solution in this case. This calculation involves deciding isomorphism of modules for the finitely presented group, and much of the chapter is devoted to describing algorithms for computing with modules. Algorithms of Schneider [4] for computing direct-sum decompositions of modules into indecomposable modules are a starting point for a simple solution of the isomorphism problem for modules. A particularly important element of the split extension case is the computation of a generating set for the group of module automorphisms. This is equivalent to finding

---

Received 14th August, 1995.

Thesis submitted to The Australian National University, August 1994. Degree approved: March, 1995.  
Supervisor: Dr M.F. Newman.

---

Copyright Clearance Centre, Inc. Serial-fee code: 0004-9729/96 \$A2.00+0.00.

a generating set for the centraliser of a matrix group in the general linear group. Explicit formulae for the elements of such a generating set are given, relying only on well known formulae for generating sets of general linear groups (for example [6]) and information already computed during the module decomposition calculation.

In Chapter 5 we consider the general or nonsplit case. This relies on much of the information computed about the corresponding split extension, and involves some additional computations. The specialisation of this step to the types of extension required for soluble groups is also described.

Chapter 6 describes the algorithm for computing the automorphism group of a finite soluble group. We start the chapter by defining special power conjugate presentations, following the description given in Eick [2]. These presentations exhibit precisely the structure required to assemble the algorithm for finite soluble groups from the algorithms described in earlier chapters. We then show how some of the information exhibited by such a presentation can be used both to reduce the amount of work required to compute the automorphism group, and also to compute a generating set that exhibits more structural information for the automorphism group.

We conclude in Chapter 7 with a hand calculation following the steps of the automorphism group algorithm as described earlier. This calculation plays two roles. The first is as an extended example of the steps involved in computing the automorphism group of a split extension. More importantly, it shows that the algorithm may be applied to compute the automorphism groups of an infinite number of groups in a single calculation. An infinite family of groups is constructed, indexed by the set of all odd primes. The automorphism group of an arbitrary member of the family is computed. The result of this calculation is a single parameterised generating set for all of the automorphism groups as well as a function giving the order of each automorphism group. The example raises the possibility of incorporating algorithms for working with symbolic parameters into an implementation of the automorphism group algorithm, or indeed of other group theoretic algorithms, and having similar computations performed automatically.

#### REFERENCES

- [1] W. Bosma and J. Cannon, *Handbook of Magma functions* (Department of Pure Mathematics, Sydney University, Australia, 1993).
- [2] B. Eick, *Spezielle PAG-Systeme im Computeralgebrasystem GAP* (Diplomarbeit, RWTH, Aachen, 1993).
- [3] D.J. Robinson, 'Applications of cohomology to the theory of groups', in *London Math. Soc Lecture Note Series 71*, Groups - St. Andrews, 1981 (Cambridge University Press, 1981), pp. 46–80.

- [4] G.J. Schneider, 'Computing with endomorphism rings of modular representations', *J. Symbolic Comput.* **9** (1990), 607–636.
- [5] M. Schönert et al., **GAP** - *groups, algorithms and programming* (Lehrstuhl D für Mathematik, RWTH, Aachen, 1993).
- [6] D.E. Taylor, 'Pairs of generators for matrix groups', *The Cayley Bulletin* **3** (1987).

Centre for Mathematics and its Applications  
The Australian National University  
ACT 0200  
Australia  
e-mail: Michael.Smith@maths.anu.edu.au