

EFFECTIVE ALGORITHMS FOR ALGEBRAIC GROUPS

1. AIMS AND SIGNIFICANCE

Group theory is the mathematical study of symmetry. The *algebraic groups* (Humphreys, 1975) lie at the heart of mathematics. An *algebraic variety* is any mathematical structure that can be defined by a system of algebraic equations—algebraic groups describe the symmetries of a variety. The public key encryption scheme used to secure credit cards is based on algebraic varieties (Koblitz et al., 2000). The symmetries of a system of differential equations, such as those describing a physical system, are described by a *Lie group* (Bourbaki, 1989). Most Lie groups are algebraic groups defined over the real or complex numbers. The symmetries of an error-correcting code will often be an algebraic group defined over a finite field.

Computational group theory is a well established area of research at the University of Sydney (Bosma and Cannon, 1997). It has many applications and is also intimately connected to more theoretical approaches to the study of symmetry. It provides valuable computational tools for mathematicians, computer scientists and physicists. On the other hand, it frequently leads to new theory and the exposure of gaps in existing work.

This project is concerned with algorithms for computing with reductive algebraic groups. This is an area in which little work has been done, but it is ripe for further development. Major progress has been made recently in computation with *Lie algebras* (Cohen et al., 1997), the linearised versions of algebraic groups. This has opened up new possibilities for algorithms for reductive algebraic groups. Reductive groups are substantially more complicated than the Lie algebras, but also much more useful in applications.

A motivating application of this project is the matrix group recognition project, an international project headed by Charles Leedham-Green, Eamonn O'Brien and Cheryl Praeger. This project aims to construct a general suite of algorithms for computing in matrix groups, that relies on breaking a group down into simple groups, most of which are finite groups of Lie type. Once the group is broken down, efficient methods for computing with the finite groups of Lie type will be provided by our project.

2. APPROACH

Almost any structural question about a group can be reduced to questions about its subgroups, conjugacy classes, automorphisms or representations. These computational problems have good solutions for permutation groups (Cannon and Havas, 1992). However algebraic groups are best described in terms of a presentation (Sims, 1994) or as matrix groups (Carter, 1965). We will provide efficient methods for converting between these two ways of representing an algebraic group, so that we can use whichever is most useful for solving each of the fundamental problems.

Some specific innovations of our project will be:

- *Algorithms for group elements:* The project marks the opening of a serious campaign to develop algorithms for computing with the elements of reductive algebraic groups. These will be the first algorithms for computing within the groups themselves. We will use two distinct computer representations of group elements, together with effective methods for converting between them. This approach is useful because some problems are much easier in one representation than the other.
- *General reductive groups:* We will work with general reductive groups, rather than restricting our attention to some subclass such as the semisimple groups. This will require us to extend many results that exist for semisimple groups to the reductive case. It will also involve the first ever algorithms for projective matrix groups.

- *Automorphisms:* These are the structure preserving mappings from the group to itself, and play an important role in group theory. Creating a classification for reductive groups will require new theory. Once the automorphisms are in place, we can extend our study to algebraic groups twisted by an automorphism. These twisted groups include the Unitary groups, one of the most important Lie groups within physics.
- *Conjugacy classes:* Knowledge of the conjugacy classes gives fundamental insights into the types of elements that a group contains. This will be the first attempt to compute conjugacy classes of infinite groups. It will also improve the computational techniques for finite groups.
- *Representation theory:* Representations are one of the fundamental theoretical tools in the study of groups. If you take the Lie group describing the symmetry of a physical theory, the representations describe the fundamental particles that exist in that theory. This area is more important and active today than ever in its hundred year history. We shall concentrate on aspects of representation theory connected to algebraic geometry.
- *Maximal subgroups:* One of the most difficult problems in computational algebra is to determine the subgroup lattice of a group. Using standard techniques, this problem can be reduced to the problem of finding maximal subgroups of the finite simple groups, most of which are groups of Lie type. This is one of the main motivating problems for this project.

3. METHODS AND TECHNIQUES

Developing algorithms. The aim of our project is to develop effective new algorithms for algebraic groups. Where appropriate we intend to provide a run-time analysis of the resulting algorithms. This is important for theoretical computer science as well as being a useful guide to the practicality of algorithms. Ideally we would like to show that our algorithms run in polynomial time (Knuth, 1975), but this will not always be possible. For example, the conjugacy class algorithms cannot be polynomial time, because the number of classes is not polynomial in the size of the input group. In this case we will investigate other approaches, such as devising more compact descriptions of the classes.

In our experience, it is important to implement algorithms as well as analysing them. Not merely does this allow others to use our results for their own work, but the process of implementation often points out deficiencies in our approach and can lead to better theoretical ideas.

We will make our algorithms as general and flexible as possible. The wide range of applications of Lie theory makes it impossible to predict where and how our work will be applied, so this approach will prevent our preconceptions getting in the way of future users.

Algorithms for group elements. A fundamental aspect of our approach is the ability to easily convert between two different computer representations for group elements: the Steinberg presentation and the matrix representation. We will discover effective algorithms for converting between the two representations. Any problem can be solved in whichever representation is most convenient, and then converted to the other as necessary. One direction of this conversion will rely on a recent algorithm by de Graaf (2000) for Lie algebras. Together with Cohen, we have already done some preliminary work on the other direction, which will use entirely new techniques modeled on the PLU decomposition algorithm.

The Steinberg presentation is closely integrated with the Lie theory that underlies most of the theoretical results in the field. Each element is stored as a “word” in the form of a Bruhat decomposition (Carter, 1993). This makes the underlying root system, which is used to classify these groups, explicit in our description of the elements. Hence we can readily exploit the large body of theoretical knowledge about algebraic groups, which is not really practicable with the existing matrix representations.

The Steinberg presentation is also naturally parametrised by field elements, which means that it fits immediately into the scheme-theoretic framework of modern algebraic geometry. This makes it possible to do computations that involve whole classes of groups at the same time, rather than having to restrict yourself to a single field as was the case with matrices. It will also facilitate connections between our work and algebraic geometry, which will give us a basis for computing representations.

Once we have this basic framework of presentations and matrix representations, we will design algorithms for the fundamental structural properties of groups: automorphisms, conjugacy classes, representations and maximal subgroups.

General reductive groups. The traditional Steinberg presentation applies only to the semisimple groups, a subclass of reductive groups. We will devise new presentations for all reductive groups. This extra generality allows us to compute with a number of important groups which are not semisimple, including the general linear and general unitary groups.

We will also need to use projective matrix representations, which are matrix representations with matrices that differ by a scalar identified. This representation has not been used previously in computation, and should make it much easier to describe a number of interesting groups.

Automorphisms. Every automorphism of a reductive algebraic groups can be decomposed into a diagonal automorphism, a field automorphism, and a graph automorphism. The first two types can be added into our Steinberg presentation in a relatively straightforward manner, provided that our mechanisms for dealing with such presentations is sufficiently general. This is another reason for computing with general reductive groups, rather than restricting ourselves to semisimple groups—a reductive group is essentially a simple algebraic group with diagonal automorphisms added.

A graph automorphism arises from the underlying Lie theory, so can be described within the Steinberg presentation.

Conjugacy classes. The conjugacy classes of the semisimple groups have been parametrised by certain other objects (Wall, 1980; Carter, 1993). We must create similar parametrisations for the reductive groups and solve the difficult problem of determining precisely which group elements these parameters actually correspond to. This will yield new theoretical insights as well as a practical algorithm.

The conjugacy classes of algebraic groups can be obtained by combining together the unipotent classes and the semisimple classes. The unipotent conjugacy classes are independent of the underlying field and there are only finitely many of them for any given group. These will be described within the Steinberg presentation.

Semisimple conjugacy classes are much more difficult to deal with. In finite groups, they have the advantage that the classes tend to be quite large, so they can be found by random sampling, provided that sufficiently good methods are available for generating random elements. In infinite groups, we will need to develop interesting new theory to describe the conjugacy classes in a finite manner. This will involve results from field theory as well as Lie theory.

Representation theory. This project will be distinguished by the use of the techniques of algebraic geometry for computing the representation theory. Algebraic geometry is the main tool for much of the recent work in this area, but has not yet been exploited computationally. The existence of new tools for computational algebraic geometry now makes it possible to take this approach.

For example, the varieties described by Deligne and Lusztig (1976) are vital to the representation theory of finite groups of Lie type. A great deal of progress has been made recently in computational algebraic geometry by the MAGMA project here in Sydney. We will use this resource to develop algorithms for such fundamental concepts as the Harish-Chandra induction of a representation.

Maximal subgroups. A great deal of work in this area has been done for the classical groups (Kleidman and Liebeck, 1990) and the exceptional groups have been dealt with on a case by case basis. In order to make an effective algorithm we will devise a Lie theoretic approach to this problem. We will also need many new results for cases not covered by the existing theory.

REFERENCES

- W. W. Bosma and J. J. Cannon. *Handbook of Magma functions*. School of Mathematics and Statistics, University of Sydney, Sydney, 1997.
- Nicolas Bourbaki. *Lie groups and Lie algebras. Chapters 1–3*. Springer-Verlag, Berlin, 1989. Translated from the French, Reprint of the 1975 edition.
- John Cannon and George Havas. Algorithms for groups. *Australian computer journal*, 24(2):51–60, May 1992.
- R. W. Carter. Simple groups and simple Lie algebras. *J. London Math. Soc.*, 40:193–240, 1965.
- Roger W. Carter. *Finite groups of Lie type*. John Wiley & Sons Ltd., Chichester, 1993. Conjugacy classes and complex characters, Reprint of the 1985 original, A Wiley-Interscience Publication.
- A. M. Cohen, W. A. de Graaf, and L. Rónyai. Computations in finite-dimensional Lie algebras. *Discrete Math. Theor. Comput. Sci.*, 1(1):129–138, 1997. Lie computations (Marseille, 1994).
- Willem A. de Graaf. *Lie algebras: theory and algorithms*. North-Holland Publishing Co., Amsterdam, 2000.
- P. Deligne and G. Lusztig. Representations of reductive groups over finite fields. *Ann. of Math. (2)*, 103(1):103–161, 1976.
- James E. Humphreys. *Linear algebraic groups*. Springer-Verlag, New York, 1975. Graduate Texts in Mathematics, No. 21.
- Peter Kleidman and Martin Liebeck. *The subgroup structure of the finite classical groups*. Cambridge University Press, Cambridge, 1990.
- Donald E. Knuth. *The art of computer programming*. Addison-Wesley Publishing Co., Reading, Mass.-London-Amsterdam, second edition, 1975. Volume 1: Fundamental algorithms, Addison-Wesley Series in Computer Science and Information Processing.
- Neal Koblitz, Alfred Menezes, and Scott Vanstone. The state of elliptic curve cryptography. *Des. Codes Cryptogr.*, 19(2-3):173–193, 2000. Towards a quarter-century of public key cryptography.
- Charles C. Sims. *Computation with finitely presented groups*. Cambridge University Press, Cambridge, 1994.
- G. E. Wall. Conjugacy classes in projective and special linear groups. *Bull. Austral. Math. Soc.*, 22(3):339–364, 1980.