J. Symbolic Computation (1997) **24**, 233–234

Special Issue on Computational Algebra and Number Theory: Proceedings of the First MAGMA Conference

Foreword of the Guest Editors

This special issue of the *Journal of Symbolic Computation* is a collection of papers arising from the MAGMA conference that was held at Queen Mary and Westfield College, London, 23–27 August, 1993, and organized by Charles Leedham-Green. The MAGMA system, developed by John Cannon and associates at Sydney University, is a successor to CAYLEY and, like its predecessor, is geared towards efficient computation within specific, and well-defined, mathematical structures, such as groups, rings, fields, modules, algebras and incidence structures.

To quote from Geddes *et al.* (1992), there are three recognizable forces at work in the development of symbolic mathematical computation, namely *algorithms*, *systems*, and *applications*. All three themes are well-represented in the papers in this volume and although several of the papers cover material from more than one of the themes, we have attempted to order the contents according to this categorization.

The system papers include detailed descriptions by their authors of the design philosophy, together with a summary of the syntax and scope, of the MAGMA language. The algorithm and application papers cover computation in such diverse areas as group theory (finite and infinite), number theory, polynomial algebra (including Gröbner bases), Galois theory, lattices and modules. Attention is generally focused on efficient implementation of algorithms (either as stand-alone programs or as part of systems) and performance analysis, in addition to mathematically accurate theoretical descriptions.

With such a broad range of subject matter, the question arises as to just what are the distinguishing features of the areas of mathematics or computer algebra that concern us here. This question would have been easier to answer 10 or 15 years ago, so let us start by casting our minds back to the early 1980s.

In 1982, a conference on "Computational Group Theory" was held in Durham, England, and the proceedings were published in Atkinson (1984). At that time, the area of computational group theory stood out clearly as something distinct from the rest of symbolic computation. The most striking distinguishing feature was the emphasis on manipulating complete structures (principally groups and their subgroups, and character tables) rather than just the elements of these structures. The actual computations that were carried out for nontrivial applications were more often than not exceedingly CPU and memory intensive, and so efficiency of implementation was of paramount importance, whereas generality and portability of software were considered secondary. In fact, the whole field revolved around two or three central families of associated algorithms, most notably Todd–Coxeter methods for handling finitely presented groups, and Schreier–Sims and base/strong-generating set methods for finite permutation groups.

The most significant change since that time is that, even if one wanted to, it would no longer be possible to treat a particular area of computational algebra, such as computational group theory, as an isolated and self-sufficient branch of mathematics. As the

0747–7171/97/030233 + 02 $25.00/0 sy970124

various different branches of computational algebra mature they are seen to rely on a common set of fundamental tools. To take a simple example, if we wish to compute chief factors of a group G, we may perhaps quickly find elementary abelian sections M/N of G (where M and N are normal in G). But we then need to refine the section, and to do this, we need to consider M/N as a module for G over a finite field, and to find a composite series of the module. Currently, the best method known of achieving this involves factorizing polynomials over finite fields, which is drawing us much closer to the realms of traditional symbolic computation. Another example concerns techniques for the efficient computation of Hermite and Smith normal forms for integral matrices (and the LLL algorithm) which are key tools in both computational group theory and algebraic number theory.

This observation concerning the interconnected nature of the various branches of computational algebra led to the conception of Magma as a system designed to support computation across all branches of algebra and number theory and constructed around a core consisting of highly efficient implementations of the fundamental algorithms of computational algebra.

Certainly there is much more common functionality between Magma and a computer algebra system such as Maple or Mathematica than there would have been 15 years ago. They each include extensive facilities for computing with arbitrarily large integers, primality testing, factorizing integers, manipulating and factorizing polynomials and so on. Increased priority has also been given to using well-designed and portable code. Indeed, perhaps in another 15 years it will be possible to have a single system for the whole of symbolic mathematical computation; that is hard to predict.

For the time being, however, the emphasis on computation with complete and specific structures, and the indispensability of perfectly tuned efficient implementations of the fundamental algorithms for these computations remain distinguishing features of Magma and the areas that it is representing. For any system to be usable for serious computations in these areas, it needs to be equipped with an extensive and detailed knowledge of the structures with which it is dealing, whether they be groups, rings, algebras, modules or number fields.

The editors would like to apologize, particularly to those authors who submitted their contributions immediately following the conference, for the length of time that it has taken to bring this collection of papers to press. This has been due to an unfortunate combination of delaying factors, at least some of which have been beyond the editors' control!

We wish to thank the individuals who refereed the papers for this volume, and, particularly, Greg Butler, who acted as editor in the case of the two papers co-authored by Cannon and Holt.

John Cannon
Derek Holt

References

Geddes, K. O., Czapor, S. R., Labahn, G. (1992). *Algorithms for Computer Algebra*. Kluwer Academic Publishers.

Atkinson, M. D. (ed.) (1994). *Computational Group Theory*. Academic Press.