Executive Summary

Dagstuhl-Seminar "Challenges in Symbolic Computation Software" 2006

Wolfram Decker, Mike Dewar, Erich Kaltofen, Stephen M. Watt

Symbolic computation software allows mathematicians, scientists, engineers, or educators to deal with elaborate calculations using a computer. The applications range from introducing the experimental method in fields of pure mathematics to practical applications, for instance, in cryptology, robotics, or signal theory. The software includes mainstream commercial products such as Maple or Mathematica and highly specialized, public domain systems such as CoCoa, Macaulay2, or Singular.

Symbolic computation software implements a variety of sophisticated algorithms on polynomials, matrices, combinatorial structures, and other mathematical objects in a multitude of different dense, sparse, or implicit (black box) representations.

The subject of the seminar was innovation in algorithms and software, bringing algorithm designers, software builders, and software users together. The program consisted of 21 talks of 45 minutes, 3 panel discussions and a presentation of the "Oberwolfach references on mathematical software (G.-M. Greuel). Supplementing the material directly related to the talks and discusssions, the proceedings contain additional contributions by J. Abbott ("Challenges in computational commutative algebra"), A. Storjohann ("Notes on computing minimal approximant bases"), and Stephen M. Watt ("Pivot-Free Block Matrix Inversion").

Well-known fundamental computer algebra algorithms include Buchberger's Groebner basis algorithm (adressed in the talks by A. Frühbis-Krüger, S. Laplagne, M. Noro), algorithms for system solving via triangular systems (X. Dahan, J.-G. Dumas), for linear algebra problems (E. Schost, A. Storjohann), for sparse interpolation (W.-S. Lee), for GCD computations and polynomial factorization (E. Kaltofen, S.M. Watt, L. Zhi), Fast Fourier Transformation algorithms (J. Johnson), and algorithms for dealing with differential and difference polynomials (X.-S. Gao, M.M. Maza). The variety of problems adressed in the different talks ranged from improving one of the basic algorithms (for instance, Noro) to discussing sophisticated algorithms based on the fundamental algorithms, such as Villamayor's algorithm

for resolving singularities which made an algebraic geometers dream come true (Frühbis-Krüger). Also treated were symbolic-numerical methods (Lee, Zhi) and purely numerical approaches based on algebraic ideas (T. Ashby). Various implementation details were given.

A number of talks dealt with system building in all its different flavors, adressing basic problems related to creating high-performance software code and high-level interfaces (J. Abbott, Johnson, R. Rioboo), pen based systems and computer algebra (G. Labahn), and the GNU TeXmacs platform and computer algebra (J. van Hoeven).

Particular applications discussed include applications to real algebraic geometry (Lazard) and to problems arising from the integration of rational functions (P. Paule).

The topics of the panel discussions were "Visions for computer algebra in five years", "What can't, but should be done by computer algebra", and "What will be the next killer application of computer algebra". These discussions also adressed "political" questions such as:

- How to transfer computer algebra knowledge into application areas?
- How to create computer algebra tools for people below university level?
- How to make computer algebra systems more user-friendly?
- How to evaluate the quality of work done in computer algebra?
- How can the computer algebra community and the numerical analysis community be linked?

These themes also entered the lively discussions taking place between the lectures and in the evenings.