Abstracts — icms2006

Plenary Lectures.

Date: Sep 2 (Jean B. Lasserre) and Sep 3 (John Harrison)

1. Jean B. Lasserre, Moments, Sum of Squares and Semidefinite Programming
   Abstract: We will introduce the generalized problem of moments (GPM) and briefly describe some of its many potential applications, including optimization, probability, mathematical finance and optimal control. When data are polynomials we then show how the GPM can be approximated (and sometimes even solved) via semidefinite programming relaxations. On the way we will develop some aspects of the fruitful duality between moments and sums of squares.

2. John Harrison, Formalizing Mathematics
   Abstract: In principle, most present-day mathematics can be formalized in various standard formal systems, such as ZF set theory or higher order logic. That is, the required mathematical structures can be defined in terms of basic set-theoretic constructs, and the theorems can be proved by sequences of elementary steps in formal logic. In the last few decades, this has become a practical reality thanks to the use of various computerized proof assistants, which check the correctness of formal proof texts and help to construct parts of them automatically. Several non-trivial theorems including the Jordan Curve Theorem and the Prime Number Theorem have recently been completely formalized.

   A major attraction of formal proofs is that they admit a simple algorithmic checking process, and are therefore more reliable than the usual ‘social process’ of peer review. We can imagine a day when most published mathematical papers will be accompanied by a formally checkable proof text, so that formal correctness will no longer be a subject for debate. This is particularly attractive for proofs that rely on very lengthy and complex arguments and/or computer assistance, where traditional peer review falters. Recently, the Four Colour Theorem has been formalized, and Hales’s proof of the Kepler Conjecture is a promising future target.

   However, current proof assistants are quite difficult to use. If we want to bring them into the mathematical mainstream, they need to be improved in various respects. In particular, there seem to be many lessons from the world of computer algebra. In my talk, I will survey what has already been achieved in the formalization of mathematics and outline my impression of future prospects.

1 Session 1. New Developments on Computer Algebra Packages, Andres Iglesias, Tetsuo Ida

Date: Sep 3. Morning and Afternoon.

1. Alfred G. Noel, A general computational scheme for testing admissibility of nilpotent orbits of real Lie groups of inner type ,(moved to Sep 1, Evening)

   Abstract: One of the most fundamental problems in the field of Representation Theory is the description of all the unitary representations of a given group. For non-compact real reductive Lie groups, there is evidence that new unitary representations can be obtained from data provided by their *admissible* nilpotent orbits. In this paper, we describe a general scheme for determining the admissibility of a given real nilpotent orbit. We implement some parts of the scheme using the software system *LiE*. We give a detailed example and study the complexity of the algorithms.

2. A. Iglesias, R. Ipanaque, R.T. Urbina, Symbolic Manipulation of Bspline Basis Functions with Mathematica
   Abstract: Bspline curves and surfaces are the most common and most important geometric entities in many fields, such as computer design and manufacturing (CAD/CAM) and computer graphics. Because of that, a huge number of algorithms and packages for dealing with Bsplines have been developed during the last three decades. However, most of them perform computations exclusively in a numerical way. In particular, up to our knowledge no computer algebra package includes especialized routines for Bsplines so far. In this paper, we describe a new Mathematica program to compute the Bspline basis functions in a fully symbolic manner. The program has been implemented in Mathematica v4.2 and later releases - such as v5.0, v5.1 and v5.2 - are also supported. The program provides the user with a highly intuitive, mathematical-looking output that is consistent with Mathematica's notation and syntax. The performance of our code is discussed by means of some illustrative and interesting examples.

3. V. Alvarez, J.A. Armario, M.D. Frau, P. Real, A Mathematica notebook for computing the homology of iterated products of groups
   Abstract: Let G be a group which admits the structure of an iterated product of central extensions and semidirect products of abelian groups G_i (both finite and infinite). We describe a *Mathematica 4.0* notebook for computing the homology of G, in terms of some homological models for the factor groups G_i and the products involved. Computational results

provided by our program have allowed the simplification of some of the formulae involved in the calculation of $H_n(G)$. Consequently the efficiency of the method has been improved as well. We include some executions and examples.

4. Vitaly Eltekov, Applications of Planner B in Mathematical Logic
   Abstract: The methods of formal and algebraic representations in logic are considered on the basis of Planner language. In the formal logic approach some productive and re-writing rules are introduced. When applying to the predefined assertions, axioms, they allow sometimes to achieve a given goal. In the algebraic approach some Boolean expressions are evaluated. Among them the tautologies play important role. In this paper the Planner-B program, which joins these approaches, is proposed. Such a joint approach gives a tool for possible explanation of some logical paradoxes.

5. Igor Gachkov, Package CodingTheory.m in Mathematica for finding quasi-perfect codes
   Abstract: By using of possibilities of the package “Coding Theory” in “Mathematica” author have an opportunity to check and correct some classical codes (A. A. Hashim and A. G. Constantinides 1974) and show that one of codes is quasi-perfect code.

6. Xin Li, Marc Moreno Maza, Efficient Implementation of Polynomial Arithmetic in a Multiple-level Programing Environment
   Abstract: The purpose of this study is to investigate implementation techniques for polynomial arithmetic in a multiple-level programing environment. Indeed, certain polynomial data-types and algorithms can further take advantage of the features of lower level languages, such as their specialized data structure or direct access to machine arithmetic. whereas other polynomial operations, like Gröbner basis over an arbitrary field, require generic programming in a high-level language.

   We are interested in the integration of polynomial data-type implementations realized at different language levels, such as LISP, C and ASSEMBLY. In particular, we consider situations for which code from different levels can be combined together within the same application in order to achieve high-performance.

   To this end, we have developed implementation techniques in the multiple-level programming environment provided by the computer algebra system AXIOM. For a given algorithm realizing a polynomial operation, available at the user level, we combine the strengths of each level including the machine architecture. Our experimentations show that this allows us to improve performances of this operation in a significant manner.

7. Masayoshi Sekiguchi, Satoshi Yamashita, and Setsuo Takato, Development of a Maple Macro Package Suitable for Drawing Fine TEX-Pictures
   Abstract: The authors have developed a Maple macro package: *KETpic*

which generates LaTeX source codes for clear drawings. KETpic enables us to draw every kind of complicated figure easily. Users are simply required to command Maple, with KETpic loaded, to plot graphs, to create LaTeX source codes by KETpic commands, and to embed them into LaTeX source files. Desired figures are obtained by the usual LaTeX compilation. Two versions of KETpic for Macintosh and Windows are provided, both of which are available for Linux users. Figures finally obtained, either on a PC display or on printed matter, are clear and possess the highest accuracy. KETpic does not require an expensive printer. Carefully prepared figures are significantly advantageous for mathematical education because they facilitate students' understanding of difficult mathematical notions. In this paper, we describe the advantages of KETpic with typical examples.

8. Hitoshi Yanami, Hirokazu Anai, SynRAC: A Maple Toolbox for solving real algebraic constraints
   Abstract: We present a Maple package for solving real algebraic constraints. This toolbox, named SyNRAC, is aimed at becoming a core toolbox on which MATLAB toolboxes are built to assist in solving various practical problems derived from such areas as engineering, bioinformatics, and mathematical biology. SyNRAC now has two types of speical QE procedures and a general QE procedure based on cylindrical algebraic decomposition as well as several formula simplification procedures.

9. Akemi Galvez, Andres Iglesias, Matlab-Based Problem-Solving Environment for Geometric Processing of Surfaces
   Abstract: In this paper a new problem-solving environment (PSE) for geometric processing of surfaces is introduced. The PSE has been designed to be responsive to the needs of our collaboration with an industrial partner, the Spanish company CANDEMAT S.A., devoted to build moulds and dies for the automotive industry. The PSE has been implemented in *Matlab* and is aimed to support the full range of activities carried out by our partner in the field of geometric processing of surfaces for the automotive industry. Firstly, the paper describes the architecture of the system and some implementation details. Then, some examples of its application to critical problems in the automotive industry - such as the computation of the intersection curves of surfaces, the generation of tool-path trajectories for NC machining and the visualization of geometric entities stored in industrial files of several formats - are briefly described. The PSE has shown to provide our partner with accurate, reliable solutions to these and other problems and to serve as a communication channel for exchange of geometrical data as well as a platform for trial and research support.

2 Session 2. Interfacing Computer Algebra and Mathematical Visualization, Konrad Polthier

Date: Sep 1. Morning.

1. Predrag Janicic, GCLC — A Tool for Constructive Euclidean Geometry and More than That
   Abstract: We present GCLC/WINGCLC— a tool for visualizing geometrical (and not only geometrical) objects and notions, for teaching/studying mathematics, and for producing mathematical illustrations of high quality. GCLC uses a language GC for declarative representation of figures and for storing mathematical contents of visual nature in textual form. In GCLC, there is a build-in geometrical theorem prover which directly links visual and semantical geometrical information with deductive properties and machine–generated proofs.

2. Tim Hoffmann and Markus Schmies, jReality, jtem, and oorange — a way to do math with computers
   Abstract: This paper presents our approach to the question of how to code mathematics (mostly experimantal and motivated from geometry) in Java. We are especially interested in the question how the development of mathematical software and the mathematics itself influence each other and how the design of programming tools and code can support this interrelationship.

3. Christopher Creutzig, MuPAD's graphics system
   Abstract: Starting with MuPAD Pro 3.0, we introduced a new framework for 2D and 3D graphics including animations in computer algebra. Based around the concept of graphical objects that are fully manipulable from the programming level as well as interactively, the framework has proven to be well-designed and flexible. We will present both the users' and the developers' perspective, including how to implement new graphical primitives and a discussion of current limitations.

4. Hans-Christian Hege, Visualization of Dynamical Systems
   Abstract:

5. Martin von Gagern, Hyperbolic Ornaments - Drawing in Non-Euclidean Crystallographic groups
   Abstract: The rigid motions of the euclidean plane form discrete groups; visualizing them is of artistic and educational value. After an introduction to hyperbolic geometry and its properties, the concept of rigid motions can be applied there as well.

   Based on these foundations, hyperbolic triangle groups and their subgroups are studied. Triangle tilings allow for intuitive input of symmetry operations. For several such operations the group generated by them can be calculated, resulting in the identification of orbits and fundamental domains of the pattern.

To allow the user to actually draw in the defined group in realtime, the concept of reverse pixel lookup is introduced, mapping pixels in the final display to pixels in a single fundamental domain. This method has applications to other self-mappings of two-spaces as well.

3 Session 3. Computer Mathematics, Freek Weidijk, advisory organizer: Henk Barendregt

Date: Sep 3. Morning and Afternoon.

1. Georges Gonthier, The Computer Mathematics of Planarity
   Abstract: The computer formalization of a theorem isn't just an exercise in verification; it can also provide new mathematical insight, that come from rigorously defining all key concepts. We will illustrate this thesis with examples drawn from our recent formalization of the proof of the Four Colour Theorem in Coq. The concept of planarity is central to this proof. Planarity has topological and combinatorial characterizations, which are often confused in arguments that are both pictorially appealing and logically incomplete. The rigor of our computer proof imposed a strict separation between the two. On the one hand we developed a purely combinatorial theory base on a symmetrical presentation of hypermaps, and to new results such as an elegant analogue of the Jordan Curve property. These led to several simplifications in the proof, in particular for constructing embeddings. On the other hand we realized that the Theorem could be proved under minimal topological assumptions; in particular our proof is independent of the Jordan Curve theorem, because we only use the Euler formula to move from topological to combinatorial planarity.

2. Jeremy Avigad, Verifying real inequalities
   Abstract: Formal verification of ordinary mathematical proofs, such as proofs of the prime number theorem and parts of the proof of the Kepler conjecture, requires verifying inequalities between real-valued expressions. I will describe some of the inferences that arise in practice, and present joint work with Harvey Friedman that provides a strategy for dealing with such inferences. I will also present some of our related decidability and undecidabilty results.

3. Tom Hales, Relaxation and the Kepler Conjecture
   Abstract: The Kepler Conjecture on sphere packings may be formulated as a large-scale nonlinear optimization problem. One of the methods that was used to solve this nonlinear optimization problem is linear relaxation. Linear relaxation reduces the original nonlinear problem to a series of linear optimization problems, which can then be solved by linear programming methods. In the 1998 proof of the Kepler Conjecture, the linear

relaxation was carried out mostly by hand. This talk will describe how to automate the linear relaxation parts of the proof of the Kepler Conjecture.

4. John Harrison, Hol Light
   Abstract: HOL Light is a theorem prover for classical higher-order logic. It is a rationalized re-implementation in Objective CAML (OCaml) of Mike Gordon's original HOL system. HOL Light generates proofs using very low-level primitive inference steps, but has numerous higher-level automated rules and a good library of pre-proved mathematics.

5. Andrzej Trybulec, Mizar
   Abstract: The Mizar project is mainly concerned with two objectives:

   (a) development of the Mizar language and related software,

   (b) development of Mizar Mathematical Library (MML) - the repository of texts, called articles, written in Mizar.

   A small group of implementators deal with the first objective with the main goal to enable authors to write articles as close as reasonable to informal mathematics. The development of MML is an international project with strong cooperation between Poland, Japan, Canada, China, and others. Although submitted articles are peer-reviewed (since the beginning of 2006), the policy of accepting new articles is still rather liberal. A lot of work is devoted to so-called revisions of MML. Some of the revisions are caused by the progress with the development of the Mizar system, while others are due to the steady growth of MML - with larger MML we can understand better how it should be organized.

6. Rob Arthan, ProofPower
   Abstract: ProofPower is a tool for specification and proof in Higher-Order Logic (HOL) and in the Z Notation, based on a re-engineering of Mike Gordon's original HOL system. Like other systems in the LCF tradition, it uses a powerful strongly-typed functional programming language (Standard ML) to ensure logical correctness by reducing all proof steps to primitive inferences rules and to provide a rich and extensible repertoire of automated derived inference rules and decision procedures. Its is used commercially for the verification of safety-critical avionics control systems. Libraries of discrete and continuous mathematics have been and continue to be developed to support these applications and for their intrinsic interest.

7. Laurent Thery, Coq
   Abstract:

4 Session 4. Software for Algebraic Geometry and Related Topics, Nobuki Takayama, advisory organizer: Gert-Martin Greuel

Date: Sep 3. Morning and Afternoon.

1. Masayuki Noro, An Efficient Implementation for Computing Groebner bases over algebraic number fields
   Abstract: In this paper we discuss Groebner basis computation over algebraic number fields. Buchberger algorithm can be executed over any computable field, but the computation is often inefficient if the field operations for algebraic numbers are directly used. Instead we can execute the algorithm over the rationals by adding the defining polynomials to the input ideal and by setting an elimination order. In this paper we propose another method, which is a combination of the two methods above. We implement it in a computer algebra system Risa/Asir and examine its efficiency.

2. M. Caboara, S. Faridi and P. Selinger, Tree checking for Sparse Complexes
   Abstract: We detail here the sparse variant of the algorithm sketched in "M. Caboara, S. Faridi, P. Selinger. Simplicial cycles and the computation of simplicial trees" for checking if a simplicial complex is a tree. A full worst case complexity analysis is given and several optimizations are discussed. The practical complexity is discussed for some examples.

3. Amir Hashemi, Strong Noether Position
   Abstract: We introduce the notion of a homogeneous ideal in Strong Noether Position (SNP); a new definition for the notion of generic coordinates for some problems. This definition is simple to check, because one can test it for the initial ideal of the ideal with respect to the degree reverse lexicographic ordering. It is explicit, because we can provide an algorithm to decide whether a monomial ideal is in SNP or not. We propose some methods to compute the Castelnuovo-Mumford regularity of an ideal which one of them is more efficient than that of [Bermejo-Gimenez, 2005].

4. Daniel Lichtblau, Cylinders Through Five Points: Computational Algebra and Geometry
   Abstract: We address the following question: Given five points in 3 , determine a right circular cylinder containing those points. We obtain algebraic equations for the axial line and radius parameters and show that these give six solutions in the generic case. An even number (0, 2, 4, or 6) will be real valued and hence correspond to actual cylinders in 3 . We will investigate computational and theoretical matters related to this problem. In particular we will show how exact and numeric Gröbner

bases, equation solving, and related symbolic-numeric methods may be used to advantage. We will also discuss some applications.

5. Fabrizio Caruso, The SARAG Library
   Abstract: In this paper we present *SARAG*, which is a software library for real algebraic geometry written in the free computer algebra system Maxima. *SARAG* stands for "Some Algorithms in Real Algebraic Geometry" and has two main applications: extending the capabilities of Maxima in the field of real algebraic geometry and being part of the interactive version of the book "Algorithms in Real Algebraic Geometry" by S. Basu, R. Pollack, M.-F. Roy, which can be now freely downloaded.

   The routines of the library deal with: theory of signed sub-resultants, linear algebra, gcd computation, real roots counting, real roots isolation, sign determination, Thom encodings, study of the topology of curves. At the moment *SARAG* is being used as a tool to develop, implement and tune algorithms coming from new research results, e.g. an algorithm for faster gcd computation, an algorithm for the study of the topology of curves over non-Archimedian real closed fields.

6. G.Pfister, On the implementation of resolution of singularities in Singular
   Abstract: Although the problem of the existence of a resolution of singularities in characteristic zero was already proved by Hironaka in the 1960s and although algorithmic proofs of it have been given independently by the groups of Bierstone and Milman and of Encinas and Villamayor in the early 1990s, the explicit construction of a resolution of singularities of a given variety is still a very complicated computational task. In my talk, I would like to outline our modification of the algorithmic approach of Encinas and Villamayor and simultaneously discuss the practical problems connected to the task of implementing the algorithm.

7. Oliver Labs, Real-time interactive visualization of deformations of singularities
   Abstract: The talk begins with an overview of the currently available software systems for visualizing singular algebraic surfaces.

   The main part of the talk consists of a presentation of the latest features of the surface visualization tool SURFEX (www.surfex.AlgebraicSurface.net). These include the interactive real-time visualization of deformations of singular algebraic surfaces on a computer.

   Some users will prefer to use non-interactive visualizations during a talk or in a publication; in the last part of the talk we thus give an example on how to produce a movie which shows a deformation of singular surfaces using SURFEX.

8. Francisco Castro Jimenez, Explicit calculations in rings of differential operators and applications
   Abstract: The Weierstrass division theorem for formal and convergent

power series, as generalized by H. Hironaka, can be extended to the case of left ideals of rings of linear differential operators. This result has many applications in computational D-module theory. We will review some of them in this talk.

9. Francisco J. Caldereron Moreno — Luis Narvaez Macarro, Intersection Complexes by means of logarithmic D-modules
Abstract: Let $D \subset \mathbf{C}^n$ be a locally quasi-homogeneous free divisor (e.g. a free hyperplane arrangement), $\mathcal{E}$ an integrable logarithmic connection with respect to D and $\mathcal{L}$ the local system of horizontal sections of $\mathcal{E}$ on $X - D$. Let $\mathrm{IC}_X(\mathcal{E})$ be the holonomic regular $\mathcal{D}_X$-module whose de Rham complex is the intersection complex $\mathrm{IC}_X(\mathcal{L})$ of Deligne-Goresky-MacPherson. In this paper we show how to use our previous results on the algebraic description of $\mathrm{IC}_X(\mathcal{E})$ in order to obtain explicit presentations of it. Concrete examples for $n = 2$ are included.

10. Viktor Levandovskyy, Plural, a Non-commutative Extension of Singular: Past, Present and Future
Abstract: We describe the non–commutative extension of the computer algebra system SINGULAR, called PLURAL. In the system, we provide rich functionality for symbolic computation within a wide class of non–commutative algebras. We discuss the computational objects of PLURAL, the implementation of main algorithms, various aspects of software engineering and numerous applications.

5 Session 5. Number Theoretical Software, Ken Nakamula, Michael Pohst

Date: Sep 1. Morning.

1. Sergey S. Lyalin, Nikolai Yu. Zolotykh, Arageli: a Library for Algebraic Computations. Overview of Architecture.
Abstract: Arageli is a C++ library for doing symbolic computations. It contains arbitrary precision integer and rational numbers, vectors, matrices, polynomials, modular arithmetic, algorithms for number factorization, linear and integer programming etc. Creating new mathematical structures from existing ones (also in other libraries) is easy. Arageli is used in Skeleton, Integizer, Prelinea projects.

2. MATSUI Tetsushi, Development of NZMATH
Abstract: NZMATH is a system oriented to calculations of number theory, based on Python. Currently, it has several basic data types and several modules for number theoretic computations. NZMATH has two key visions 1) user / developer fusion and 2) speed of development, and the system has been growing along the lines. The development is of open source by nature, and we are making effort to be as agile as possible.

There are many areas to be developed, especially a module for algebraic numbers is awaited. Some experimental user interface construction is also discussed.

3. Sebastian Freundt, Aneesh Karve, Anita Krahmann, Sebastian Pauli, KASH: Recent Developments
Abstract: In recent years the computer algebra system KASH/KANT for number theory has evolved considerably. We present its new features and introduce the related components, QaoS (Querying Algebraic Objects System) and GiANT (Graphical Algebraic Number Theory).

4. Takashi, Fukuda, TC - an interpreter of multi-precision C language
Abstract: TC (Tiny C) is an interpreter of multi-precision C language suitable for floating-point calculations of several thousands digits which often appear in computational algebraic number theory. Furthermore TC ver.4, which is equipped with PARI library and turned TCP ver.1, provides frequently used PARI library functions by C-like functions which are easy to call from TC.

6 Session 6. Methods in Computational Number Theory, David Sevilla, Jaime Gutierrez

Date: Sep 1. Afternoon.

1. Daniel Lichtblau, Making Change and Finding Refigits: Balancing a Knapsack ,(moved to Sep 2, Evening.)

   Abstract: We will discuss knapsack problems that arise in certain computational number theory settings. A common theme is that the search space for the standard real relaxation is large; in a sense this translates to a poor choice of variables. Lattice reduction methods have been developed in the past few years to improve handling of such problems. We show explicitly how they may be applied to computation of Frobenius instances, Keith numbers (also called “repfigits”), and as a first step in computation of Frobenius numbers.

2. Niels Moller, Robust HGCD with No Backup Steps
Abstract: Subquadratic divide-and-conquer algorithms for computing the greatest common divisor have been studied for a couple of decades. The integer case has been notoriously difficult, with the need for “backup steps” in various forms. This paper explains why backup steps are necessary for algorithms based directly on the quotient sequence, and proposes a robustness criterion that can be used to construct a “half-gcd” algorithm without any backup steps.

3. Feng Wang, Yasuyuki Nogami, and Yoshitaka Morikawa, A High-Speed Square Root Algorithm in Extension Fields
Abstract: A square root (SQRT) algorithm in $GF(p^m)$ ($m = r_0 r_1 \cdots r_{n-1} 2^d$, r_i : odd prime, $d > 0$: integer) is proposed in this paper. First, the Tonelli-Shanks algorithm is modified to compute the inverse SQRT in $GF(p^{2^d})$, where most of the computations are performed in the corresponding subfields $GF(p^{2^i})$ for $0 \leq i \leq d-1$. Then the Frobenius mappings with an addition chain are adopted for the proposed SQRT algorithm, in which a lot of computations in a given extension field $GF(p^m)$ are also reduce to those in a proper subfield by the norm computations. Those reductions of the field degree increase efficiency in the SQRT implementation. More specifically the Tonelli-Shanks algorithm and the proposed algorithm in $GF(p^{22})$, $GF(p^{44})$ and $GF(p^{88})$ were implemented on a Pentium4 (2.6GHz) computer using the C++ programming language. The computer simulations showed that, on average, the proposed algorithm accelerates the SQRT computation by 25 times in $GF(p^{22})$, by 45 times in $GF(p^{44})$, and by 70 times in $GF(p^{88})$, compared to the Tonelli-Shanks algorithm, which is supported by the evaluation of the number of computations.

4. Hirehiro Katou, Yasuyuki Nogami, Yoshitaka Morikawa, Cyclic Vector Multiplication is Efficient for Small Extension Degrees
Abstract: In the modern information-oriented society, several pairing-based cryptographic applications such as group signature algorithm habe been proposed. Such applications employ a certain extension field as the definition field, especially small extension degrees. In this paper, we explain the extension fields such as AOPF, TypeII AOPF, OEF, GOEF and these tower field technique, and their calculation costs. Then, we show that cyclic vector multiplication algorithm is efficient for small extension degrees, and tower field technique using CVMA is quite efficient for composite extension degrees.

7 Session 7. Free Software for Computer Algebra, Joris Van der Hoeven

Date: Sep 1. Morning and Afternoon.

1. Chris Dams, GiNaC Is Not A Computer algebra system
Abstract: GiNaC is a C++-library that provides computer algebra capabilites. It does this by defining data types that represent the various kinds of expressions. These data types are especially suitable for use from within C++. Besides allowing users to write programs in C++ that perform symbolic calculations this also makes it relatively easy to combine symbolic manipulations with functionality provided by other C/C++-libraries into one program. Although developed within the high energy physics community, its application is not restricted to this field. The library is distributed

under the GNU general public license (GPL).

2. Torbjorn Granlund, GMP
   Abstract: The GNU Multiple Precisions Arithmetic Library is known for excellent performance, but its performance for really large operands still leaves a lot to be desired. We will present ongoing developments and some new algorithms, which tries to rectify this situation for GMP 5.

3. Richard Kreckel, CLN
   Abstract: CLN is a free (as in speech) C++ library for efficient computations with all kinds of numbers in arbitrary precision. Being a spin-off of a Lisp implementation in origin, it is well suited as a base for symbolic systems or languages with arbitrary precision arithmetic. The talk describes and motivates the library's architecture, in particular the type system, and skims through selected interesting implementation aspects.

4. Bernard Mourrain, Synaps
   Abstract: SYNAPS (SYmbolic Numeric APplications) is a C++ library devoted to symbolic and numeric computations. It provides data-structures for the manipulation of basic algebraic objects, such as vectors, matrices (dense, sparse, structured), univariate and multivariate polynomials. It contains solvers for univariate and multivariate polynomials, including generalized normal form or subdivision solvers, tools for the manipulatiion of algebraic numbers, for the construction of resultants, ...

5. John Abbott, The design of CoCoALib
   Abstract: The C++ library CoCoALib offers data structures and operations for COmputations in COmmuative Algebra, most particularly Gröbner bases. Ease of use through a clean design is paramount (with some concessions to guarantee good performance). The library comes with full documentation and numerous example programs. A "beta" release is anticipated in late 2006. A server and interactive system are planned.

6. Daniel R. Grayson, Macaulay 2, a software system for algebraic geometry
   Abstract: Macaulay 2 supports research in algebraic geometry and commutative algebra. Its versatile framework, based on Buchberger's algorithm for computing Groebner bases, combined with an object-oriented interpreted language supporting the introduction of new high-level mathematical types, allows advanced algorithms to be coded. I'll demonstrate it, and I will describe recent improvements aimed to support third-party development of code, including the package system, the debugger, and the documentation processor.

7. Gerhard Pfister, Singular
   Abstract: SINGULAR is a Computer Algebra System for polynomial computations with emphasis on the special needs of commutative algebra, algebraic geometry, and singularity theory. SINGULAR's main computational objects are ideals and modules over a large variety of rings, including

local rings and non-commutative G-algebras (in the subsystem PLURAL). Large variety of algorithms, including those based on Gröbner and standard bases, have powerful implementations in SINGULAR.

8. Steve Linton, GAP
   Abstract:

9. Andrey Grozin, TeXmacs interfaces to computer algebra systems
   Abstract: TeXmacs can call external programs and communicate with them via pipes; it can also render formulae in the LaTeX format, and practically any computer algebra system (CAS) can produce its results in LaTeX. Therefore, it is straightforward to use TeXmacs as a nice GUI for a number of CASs. I have written TeXmacs interfaces to 6 CASs (though 2 of them were later largely re-written by other people). In this talk, I'll concentrate on the interfaces to the most powerful free CASs, maxima and axiom.

10. M. Moreno Maza, Axiom
    Abstract:

11. Olivier Ruatta, Mathemagix and Mmxlib
    Abstract: Mathemagix is a general purpose software for symbolic and numeric-symbolic computation. The software relies on 3 distinct components :

    (a) A general interpreter which provides a high level, strongly typed and imperative language.

    (b) A C++ dynamic library, called Mmxlib, providing efficient data structures and fast algorithms for integer, modular, real and complex numbers, intervals, complex balls, power series, effective analytic functions, linear algebra, differential operators, symbolic objects, etc.

    (c) An extension process which allows to easily integrate data structures and algorithms from external C or C++ libraries using a "glue code mechanism".

    Mathemagix can be used from within the GNU TeXmacs program.

12. Ivan Noyer, Mechanized Calculus and Algebra: the FoCaL approach
    Abstract: The talks presents the FoCaL framework which is a set of tools that helps building software systems. It is developped by French's Universite Pierre et Marie Curie, Conservatoire National des Arts et Metiers and Institut National de la Recherche en Informatique et Automatique. FoCaL enables to specify and implement programs but also to state and ensure correctness of properties about those programs.

    The heart of the FoCaL framework is the FoCaL language which has object oriented features and enables to share specifications and properties between different components of a software system. The language also enable to share and resue definition and proofs of properties. Components of the FoCaL language are called species which have multiple inheritance and allows late binding.

FoCaL is made of a compiler producing Ocaml code which is intended to be executed, running code compares favourabily with the best computer algebra packages available. Certification is achieved by producing Coq code with the help of an automatic prover which we call Zenon.

Running examples covering computer algebra, effective mathmatics and also general software security programs are presented which illustrate the expressive and computational power of FoCaL.

The framework also has tools which enable to provide MathML documentation acting as Doxygen or JavaDOC emhance with methematical capabilities which can be viewed using current Web browsers technologies. Somme program analysis and XML conversion tools are also presented.

13. Jean-Guillaume Dumas, Linbox

    Abstract: Three major threads have come together to form the exact linear algebra library LinBox. The first is the use of modular algorithms when solving integer or rational matrix problems. The second thread and original motive for LinBox is the implementation of black box algorithms for sparse/structured matrices. Finally, it has proven valuable to introduce elimination techniques that exploit the block floating point BLAS libraries even when our domains are finite fields. The latter is useful for dense problems and for block iterative methods.

    Genericity and high performance are the twin goals of LinBox. We will show that genericity is achieved by use of a small set of interfaces. Algorithms are implemented with C++ template parameters which may be instantiated with any class adhering to the specified interface. High performance is achieved by judicious specializations of the generic algorithms. Moreover, large-scale applications and software systems are getting increasingly complex. To deal with this complexity, those systems must manage themselves in accordance with high-level guidance from humans. Adaptive algorithms enable this self-management of resources and structured inputs.

    In this talk, we illustrate and demonstrate these points on the different levels of LinBox:

    (a) The adaptive "solutions" of LinBox (determinant, rank, system solution, inverse, normal forms, etc.) which enable a high level strategic choice between the algorithms.
    (b) The generic "algorithms" (Wiedemann, Krylov, Gauss, LinBox group's, etc.).
    (c) The optimized black box / dense data structures (compressed rows, apply function, BLAS, etc.).
    (d) The finite fields "archetype" and representations (floating points, Zech, Montgommery, etc.).

8 Session 8. Software for Optimization and Geometric Computation, Komei Fukuda, Michael Joswig

Date: Sep 2. Morning and Afternoon.

1. Lars Schewe, Generation of oriented matroids using satisfiability solvers
   Abstract: We present a new method to generate oriented matroids. For this we transform the problem into an instance of the well-studied satisfiability problem. To show that this method is effective we describe how it leads to new results for the realizability problem for triangulated surfaces: All triangulations of the orientable surface of genus 6 with 12 vertices do not admit a polyhedral embedding in $\mathbf{R}^3$. Additionally we can show that there exist triangulations of the orientable surface of genus 5 (with 12 vertices) without a polyhedral embedding in $\mathbf{R}^3$. Using these examples we can construct infinite families of triangulations without polyhedral embeddings in $\mathbf{R}^3$.

2. Ewgenij Gawrilow, Michael Joswig, Flexible object hierarchies in polymake
   Abstract:

3. Anders Nedergaard Jensen, A presentation of the Gfan software
   Abstract: Gfan is a command line tool for enumerating the reduced Gröbner bases of a polynomial ideal in n variables. Hereby the Gröbner fan, an n-dimensional polyhedral complex, is computed. The tropical variety is a certain subcomplex which can also be computed by the software. Gfan uses Gmp and Cddlib for exact arithmetic and polyhedral computations, respectively.

4. Anton Leykin, Jan Verschelde, and Yan Zhuang, Parallel Homotopy Algorithms to Solve Polynomial Systems
   Abstract: Homotopy continuation methods to compute numerical approximations to all isolated solutions of a polynomial system are known as "embarrassingly parallel", i.e.: because of their low communication overhead, these methods scale very well for a large number of processors. Because so many important problems remain unsolved mainly due to their intrinsic computational complexity, it would be embarrassing not to develop parallel implementations of polynomial homotopy continuation methods. This paper concerns the development of "parallel PHCpack", a project which started a couple of years ago in collaboration with Yusong Wang, and which currently continues with Anton Leykin (parallel irreducible decomposition) and Yan Zhuang (parallel polyhedral homotopies). We report on our efforts to make PHCpack ready to solve large polynomial systems which arise in applications.

5. Manuel Abellanas and Alfredo Vegas, DEpthLAUNAY
   Abstract: This paper describes the software DEpthLAUNAY. The main goal of the application is to compute Delaunay depth layers and levels of a planar point set “M. Abellanas, M. Claverol, F. Hurtado. Point set stratification and Delaunay depth (http://arxiv.org/abs/cs/0505017)”. Some other geometric structures can be computed as well (convex hull, convex layers and levels, Voronoi diagram and Voronoi levels, Delaunay triangulation, Delaunay empty circles, etc.) The application has been developed using CGAL “Computational Geometry Algorithms Library, http://www.CGAL.org”.

6. Peter M Huggins, iB4e: A Software Framework for Parametrizing Specialized LP Problems
   Abstract: The C++ library iB4e is introduced, which incorporates a user-defined specialized LP solver (such as the Needleman-Wunsch algorithm for biological sequence alignment), and computes the implicitly defined polytope for the LP instance. Thus iB4e can be used for parametric inference and sensitivity analysis in the sciences, as well as classical polyhedral computations such as convex hull and Minkowski sum. iB4e has special optimized subroutines for low dimensions and supports either native arithmetic or arbitrary precision arithmetic via GMP. Thus iB4e is robust but can still be used for particular high performance computing. After presenting the iB4e library and underlying algorithms, I’ll discuss recently reported computational advances (such as the first-ever whole genome parametric alignment) which used beta versions of the software.

7. Thomas Gerstner and Markus Holtz, Algorithms for the Cell Enumeration and Orthant Decomposition of Hyperplane Arrangements
   Abstract: We propose a novel paradigm for the design of hyperplane arrangement management software, a one-to-one correspondence between a certain set of intersection points and cells. This paradigm allows the development of very efficient algorithms which we illustrate with two examples. The first algorithm enumerates all cells in a hyperplane arrangement. The second one performs an orthant decomposition of a hyperplane arrangement using exactly one orthant per cell. Both algorithms are not difficult to implement and run in optimal order complexity. The efficiency of the algorithms is illustrated in computational results for hyperplane arrangements of varying size.

8. Colin N. Jones, Jan M. Maciejowski, Primal-Dual Enumeration for Multiparametric Linear Programming
   Abstract: Optimal control problems for constrained linear systems with a linear cost can be posed as multiparametric linear programs (pLPs) and solved explicitly offline. Several algorithms have recently been proposed in the literature that solve these pLPs in a fairly efficient manner, all of which have as a base operation the computation and removal of redundant constraints. For many problems, it is this redundancy elimination that re-

quires the vast majority of the computation time. This paper introduces a new solution technique for multiparametric linear programs based on the primal–dual paradigm. The proposed approach reposes the problem as the vertex enumeration of a linearly transformed polytope and then simultaneously computes both its vertex and halfspace representations. Exploitation of the halfspace representation allows, for smaller problems, a very significant reduction in the number of redundancy elimination operations required, resulting in many cases in a much faster algorithm.

9. Joshua D. Griffin, Tamara G.Kolda, A parallel, asynchronous method for derivative-free nonlinear programs
   Abstract: A strong need for derivative-free algorithms exists in the context of real-world optimization problems where function evaluations can be computationally expensive and noisy. The objective and constraint functions commonly exist as simple script interfaces to CPU-intensive model analysis software. A single evaluation may involve invoking cumbersome simulation codes whose run time is measured in hours. In this context, we present an asynchronous parallel implementation of a derivative-free augmented Lagrangian algorithm for handling general nonlinear constraints. The method requires approximate minimizers to a series of linearly constrained subproblems involving the augmented Lagrangian of the nonlinear constraints. These subproblem are solved using a generating set search algorithm capable of handling degenerate linear constraints. The objective and nonlinear constraint functions are computed asynchronously in parallel. A description and theoretical analysis of the algorithm will be given followed by numerical results.

9 Session 9. Methods and software for computing mathematical functions, Amparo Gil, Javier Segura

Date: Sep 2. Morning and Afternoon.

1. Nico M. Temme, Numerical Aspects of Special Functions
   Abstract: In this lecture we discuss the following topics.

   - **The complete revision of 'Abramowitz and Stegun' (1964)** For the first time after the 1964 version (of which several reprints with minor corrections were published), Abramowitz and Stegun's *Handbook of Mathematical Functions* is being completely rewritten. We give a short status report on this project.
   - **Recursion relations to compute special functions** Recursion relations of the form $A_n y_{n-1} + B_n y_n + C_n y_{n+1} = 0$, where $n = 0, 1, 2, \ldots$, are very important for computing special functions. We discuss the notions of minimal and dominant solutions, forward and

backward computation, and the Miller algorithm. We also mention function classes for which recursion relations are very useful for computation.

- **The set of 26 recursion relations for the Gauss hypergeometric functions** For the Gauss hypergeometric function we have 26 recursion relations for the functions $y_n = {}_2F_1(a+\varepsilon_1 n, b+\varepsilon_2 n; c+\varepsilon_3 n; z)$, where $\varepsilon_j = -1, 0, 1$. By using several function relations for the Gauss functions only 4 of these 26 recursions can be considered as basic forms. For these 4 basic forms we give the recursion relations and we give the minimal and dominant solutions of these recursions.
- **Numerical quadrature to compute special functions** The standard integral representations of special functions are not always suitable for numerical computations. For example, when parameters are large, integrals with oscillatory integrands can be very unstable representations. By using complex contours for these integrals, stable representations can be obtained. We give some examples of Bessel functions and Airy functions and examples of the trapezoidal rule, which can be used for developing high-precision algorithms.
- **Some pitfalls in the computation of special functions** We give several examples of minor and major difficulties in evaluating special functions and discuss the following questions.
  - (a) Can we compute the Gauss hypergeometric function by using power series and the many transformations that are available for this function?
  - (b) Can we compute this function for all values of a, b and c?
  - (c) How to interpret certain types of asymptotic expansions?
- **Can we always rely on Maple or Mathematica for computing special functions?** In many cases Maple and Mathematica provide correct numerical values, in many digits, of the frequently used special functions. In some cases, for example in the case of large complex variables, we have to be very careful with the results of these computations. We give a few examples on evaluating integrals related with special functions where in particular the occasional user of this software may get lost.
- **The project: Writing a book on the numerical aspects of special functions** These and several other topics will be discussed extensively, with examples of software, in a book to be published by SIAM, with the title of this talk, and written together with my co-authors Amparo Gil and Javier Segura.

2. S.Chevillard, Nathalie Revol, Computation of the error functions erf and erfc in arbitrary precision with correct rounding
   Abstract: The error function erf is widely used in statistical computations for instance, where it is also known as the standard normal cumulative

probability. In this paper, the computation of $\mathrm{erf}(x)$ in arbitrary precision is detailed. A feature of our implementation is correct rounding: the returned result is the exact result (as if it were computed with infinite precision) rounded according to the specified rounding mode. Our implementation uses the `MPFR` library. Finally, timings on some experiments are given, and the implementation of the complementary error function erfc is then sketched.

3. Mark Korenblit and Efraim Shmerling, Algorithm and Software for Integration Over a Convex Polyhedron
   Abstract: We present a new efficient algorithm for numerical integration over a convex polyhedron in multi-dimensional Euclidian space defined by a system of linear inequalities. The software routines which implement this algorithm are described. A numerical example of calculating an integral using these routines is given.

4. Alfredo Deano, Amparo Gil, Javier Segura, Computation of the real zeros of the Kummer function M(a;c;x)
   Abstract: An algorithm for computing the real zeros of the Kummer function $M(a;c;x)$ is presented. The computation of ratios of functions of the type $M(a+1;c+1;x)/M(a;c;x)$, $M(a+1;c;x)/M(a;c;x)$ plays a key role in the algorithm, which is based on global fixed-point iterations. We analyse the accuracy and efficiency of three continued fraction representations converging to these ratios as a function of the parameter values. The condition of the change of variables appearing in the fixed point method is also studied. Comparison with implicit Maple functions is provided, including the Laguerre polynomial case.

5. Annie Cuyt, Stefan Becuwe, Towards reliable software for the evaluation of a class of special functions
   Abstract: Special functions are pervasive in all fields of science. The most well-known application areas are in physics, engineering, chemistry, computer science and statistics. Because of their importance, several books and a large collection of papers have been devoted to the numerical computation of these functions. But up to this date, even environments such as Maple, Mathematica, MATLAB and libraries such as IMSL, CERN and NAG offer no routines for the reliable evaluation of special functions. Here the notion reliable indicates that, together with the function evaluation a guaranteed upper bound on the total error or, equivalently, an enclosure for the exact result, is computed.

   We point out how limit-periodic continued fraction representations of these functions can be helpful in this respect. The newly developed (and implemented) scalable precision technique is mainly based on the use of sharpened a priori truncation error and round-off error upper bounds for real continued fraction representations of special functions of a real variable. The implementation is reliable in the sense that it returns a sharp interval enclosure for the requested function evaluation, at the same cost as the evaluation.

6. Joris Van Deun, Ronald Cools, A Matlab implementation of an algorithm for computing integrals of products of Bessel functions
Abstract: We present a Matlab program that computes infinite range integrals of an arbitrary product of Bessel functions of the first kind. The algorithm uses an integral representation of the upper incomplete Gamma function to integrate the tail of the integrand. This paper describes the algorithm and then focuses on some implementation aspects of the Matlab program. Finally we mention a generalisation that incorporates the Laplace transform of a product of Bessel functions.

10 Session 10. Access to Mathematics on the Web, Paul Libbrecht

Date: Sep 1. Afternoon.

1. Zur'aini Dahlan and Noraimi Shafie, Multimedia Prototype of a Bilingual Model within Technology Based Learning Environment: An Implementation of a Mathematics Learning Framework.
Abstract: In response to the current globalization in the educational arena, and more importantly to the new policy of change in the medium of instruction for teaching mathematics and science in English as fostered by the Malaysian Government in 2003, we propose the e-learning Bilingual Model which has been designed and has already been tested at the university. The multimedia prototype of the model consists of text-based content of first-year mathematics subjects with exact forms available in both the English language and the native language. Content is identified to provide descriptions of core concepts dynamically using, audio, video and graphics, and also constructed to provide glossaries in both languages and to present consistent elements of the problem set to the students. The prototype is set in an integrated frameworks for Technology Based Learning Environment, through a series of case studies at the university using Short Messaging Service (SMS), MOODLE e-learning and Freeware Online-portals for Group-websites (FrOG), which are combination of Information Communication Technologies (ICT), implemented as instructional delivery tools in addition to the traditional method of teaching mathematics.

2. Paul Libbrecht and Erica Melis, Methods to Access and Retrieve Mathematical Content in ActiveMath
Abstract: This article describes how mathematical content items and formulæ are processed, retrieved, and accessed in ActiveMath. Central to the retrieval and access is a search tool which allows for searching text, attributes, relations and formulæ, and presenting items. The search tool

has been evaluated according to the standard measures of precision and recall as well as for usability. We report results of these evaluations.

3. Klaus Grue, Logiweb, a system for web publication of mathematics
   Abstract: Logiweb is a system for electronic publication and archival of machine checked mathematics of high typographic quality. It can verify the formal correctness of *pages*, i.e. mathematical papers expressed suitably. The present paper is an example of such a Logiweb page and the present paper is formally correct in the sense that it has been verified by Logiweb. The paper may of course contain informal errors like any other paper. Logiweb is neutral with respect to choice of logic and choice of notation and can support any kind of formal reasoning.

   Logiweb uses the World Wide Web to publish Logiweb pages and Logiweb pages can be viewed by ordinary Web browsers. Logiweb pages can reference definitions, lemmas, and proofs on previously referenced Logiweb pages across the Internet. When Logiweb verifies a Logiweb page, it takes all transitively referenced pages into account.

4. Josef Urban, Accessing Mizar and its Semantics on the Web
   Abstract:

11 Session 0. General

Date: Sep 2. Morning, afternoon, evening.

1. Anton Leykin, Jan Verschelde, Interfacing with the Numerical Homotopy Algorithms in PHCpack
   Abstract: *PHCpack* implements numerical algorithms for solving polynomial systems using homotopy continuation methods. In this paper we describe two types of interfaces to *PHCpack*. The first interface *PHCmaple* originally follows OpenXM, in the sense that the program (in our case Maple) that uses *PHCpack* needs only the executable version *phc* built by the package *PHCpack*. Following the recent development of *PHCpack*, *PHCmaple* has been extended with functions that deal with singular polynomial systems, in particular, the deflation procedures that guarantee the ability to refine approximations to an isolated solution even if it is multiple. The second interface to *PHCpack* was developed in conjunction with MPI (Message Passing Interface), needed to run the path trackers on parallel machines. This interface gives access to the functionality of *PHCpack* as a conventional software library.

2. Tetsuo Ida, Hidekazu Takahashi, Mircea Marin, Fadoua Ghourabi, and Asem Kasem, Computational Construction of a Maximum Equilateral Triangle Inscribed in an Origami
   Abstract: We present an origami construction of a maximum equilateral

triangle inscribed in an origami, and an automated proof of the correctness of the construction. The construction and the correctness proof are achieved by a computational origami system called *Eos* (E-origami system). In the construction we apply the techniques of geometrical constraint solving, and in the automated proof we apply Gröbner bases theory and the cylindrical algebraic decomposition method. The cylindrical algebraic decomposition is indispensable to the automated proof of the maximality since the specification of this property involves the notion of inequalities. The interplay of construction and proof by Gröbner bases method and the cylindrical algebraic decomposition supported by *Eos* is the feature of our work.

3. H.M. Buecker, A. Elsheikh, A. Vehreschild, A System for Interfacing MATLAB with External Software Geared Toward Automatic Differentiation
   Abstract: MATLAB is commonly considered to be an attractive, high-productivity programming environment by many computational scientists and engineers. So-called MEX-files are dynamically linked subroutines produced from, say, C or Fortran source code that, when compiled, can be run directly from within MATLAB as if they were MATLAB built-in functions. When applying automatic differentiation to a MATLAB program that calls external software via MEX-files, code is mechanically generated for the MATLAB part and for the external part in two separate phases. These resulting code fragments need to be put together via new MEX-files. This work introduces a novel software tool called automatic differentiation mexfunction generator that automatically generates MEX interface functions for gluing these automatically generated code fragments.

4. Joris Van der Hoeven, GNU TeXmacs: an editing platfome for scientists
   Abstract: GNU TeXmacs aims to provide a free and user-friendly software for editing various types of documents (structured texts, mathematics, computer algebra sessions, pictures, presentations, etc.). The editor enables users to compose structured documents with a high typesetting quality (similar to TeX) in a wysiwyg way. New style files and extensions to the editor can be programmed by users. TeXmacs admits interfaces for many computer algebra systems and other scientific software.

5. Tatsuyoshi Hamada, Kuniyasu Suzaki, Kengo Iijima and Arimitsu Shikoda, KNOPPIX/Math: Portable and distributable collection of mathematical software and free documents
   Abstract: We propose a new computer environment for mathematicians that can be set up easily and quickly.

6. Fabrizio Caruso, Carlo Traverso, Active Journal: an infrastructure to document algorithmic research
   Abstract:

7. Kazuhiro Yokoyama, Stability of Parametric Decomposition
Abstract: We deal with ideals generated by polynomials with parametric coefficients, and introduce "stabilities on ideal structures" based on stability of forms of Gröbner bases. Then, we extend those stabilities to radicals and irreducible decompositions and show the computational tractability on those computations by integrating existing techniques.

8. Manuel Delgado and Jose Morais, On the GAP package sgpviz
Abstract: This is a brief description of the GAP package *sgpviz*, a package designed to visualize finite semigroups through their $\mathcal{D}$-classes or Cayley graphs, as well as to make friendlier the usage of GAP when dealing with finite semigroups.

9. Francois Descouens, Making research on (q,t)-deformations of Symmetric Functions using the MuPAD package MuPAD-Combinat
Abstract: We report on the 2005 AIM workshop "Generalized Kostka Polynomials", which gathered 20 researchers in the active area of q, t-analogues of symmetric functions. Our goal is to present a typical use-case of the open source package MuPAD-Combinat in a research environment.

10. Ainhoa Berciano-Alcaraz, Symbolic software to compute A_∞-structures
Abstract: We present here a computer program allowing one to compute the explicit functional components of an $\mathcal{A}^\infty$-(co)algebra deduced from a reduction (that is, a special type of homology equivalence): our input is a reduction from a genuine dg-(co)algebra to a simple dg-module, and the output is a functional object implementing the collection $\{\mu_i\}_{i\geq 1}$ (resp. $\{\Delta_i\}_{i\geq 1}$) defining the $\mathcal{A}^\infty$-structure deduced for the dg-module from the reduction. A few simple concrete applications are given.

11. Alvarez, Armario, Frau and Real, Calculating cocyclic Hadamard matrices in Mathematica: exhaustive and heuristic searchs
Abstract: We describe a notebook in *Mathematica* which, taking as input data a homological model for a finite group G of order $|G| = 4t$, performs an exhaustive search for constructing the whole set of cocyclic Hadamard matrices over G. Since such an exhaustive search is not practical for orders $4t \geq 28$, the program also provides an alternate method, in which an heuristic search (in terms of a genetic algorithm) is performed. We include some executions and examples.

12. Hiromasa Nakayama, An Interactive User Interface for Division Algorithms and the Buchberger Algorithm
Abstract: This is an experimental project for algorithm animations and graphical or interactive user interface in computer algebra with Java and OpenXM. An animation system for the Buchberger algorithm will be discussed.

13. Hirokazu Murao, Experiment of multithreading symbolic and algebraic computations with OpenMP

Abstract: This paper describes the current status of a project for multithreading algebraic compuataions, which aims at the utilization of today's high-spec PCs with hyperthreading or dual-core technologies. Our effort is done by applying OpenXM with minimal cost of development, and includes memory management in multithreaded environment. Our empirical results show that the performance gain can be attained in numeric cases and in some cases of purely symbolic computations.