Nonlinear Analysis: Modelling and Control, Vilnius, IMI, 1998, No 3

SYMBOLIC COMPUTATION: SYSTEMS AND APPLICATIONS

Algimantas Juozapavi èius
Vilnius University, Naugarduko 24, 2006, Vilnius, Lithuania
Mathematics and Informatics Institute, Akademijos 4, 2600 Vilnius, Lithuania

Abstract

The article presents an overview of symbolic computation systems, their classification-in-history, the most popular CAS, examples of systems and some of their applications. Symbolics versus numeric, enhancement in mathematics, computing nature of CAS, related projects, networks, references are discussed.

INTRODUCTION

Symbolic computation is the science and technology that aims to automate a wide range of the computation involved in mathematical problem solving.

S***ynonyms and truncations*** for symbolic computation are: symbolic manipulation (SYM), computer algebra (CA), symbolic and algebraic manipulation (SAM), exact computation). It emphasizes discrete computation on symbols representing mathematical objects. The symbols represent not only numbers like integers and rationals, but also other mathematical objects like polynomials, radicals, rational and trigonometric functions, power series, algebraic numbers, groups, ideals, and tensors. The tasks to be solved may come from areas such as logic, statistics, automated theorem proving, but most common ones come from algebra and calculus. Computer software systems handling such tasks are usually called computer algebra systems (CAS).

Typically, the computation is ***exact***, or at least not complete numerical. The problem given to the computer is considered as a mathematical problem in a manner as human do, and it is treated by the computer in such a way. Computer manipulates with symbols and gives the answer in symbolic terms. This contrasts to the most ***numeric*** calculations where computations use approximate floating point arithmetic. Numeric computation is the development and implementation of ***computational*** algorithms.

Often though, exact and approximate calculations are used together in many important tasks. An example for such task is: in computing exactly the first ***n*** terms of a

power series that is an approximation to the solution of a differential equation. The truncated series can then be evaluated at a particular point using floating point arithmetic to get an approximation to the numerical solution of the differential equation.

The first attempts to use a digital computer for formal algebraic manipulation was done in 1953, by ***H. G.Kahrimanian*** and independently, by ***J. Nolan*** (for the history see [1]). These innovative works were done by students. Then, a ***30 years later*** it was estimated that more than 60 systems exist for doing some form of computer algebra. In our days, the current systems, and most popular are:

Axiom, Macsyma, Maple, Mathematica, Reduce, and Derive.

Notable results have been achieved in symbolic computation over the four decades, and the ***major advances*** came in last two decades: ***algorithmics*** and in ***software:***

- algorithms have been discovered for ***integration in fmite terms*** and for computing ***closed form*** solutions of differential equations;
- fast algorithms have been devised for ***factoring polynomials*** and computing greatest common divisors;
- ***powerful interactive*** systems for symbolic computation have been designed and built;
- the software ***has improved the productivity*** of scientists and engineers, it has made possible the solution of problems that were previously intractable.

However, ***only the surface*** of this iceberg has been scratched. It is still confronted with a ***wide spectrum of challenging problems*** whose solution will have a crucial influence on the technological problem solving ability.

CLASSIFICATION-IN-HISTORY OF SYMBOLIC ALGEBRA SYSTEMS

Tracking and overviewing the history of the development and implementation of CAS, it's possible to group them into classes:

- CAS, representing a collection of subroutines for some programming language (standard library), and including systems ALTRAN, FORMAC, SAC-1 and SAC-2 for Fortran, RATSIMP, CHARYBDIS for Lisp, etc. This approach was very popular a few decades ago, and it is reborning nowadays because of ubiquity of programming languages C and C++, their templates, libraries and standards;
- Systems designed to solve problems in a specific area of research and engineering, including TRIGMAN, CAMAL, ALAM, SCHOONSHIP for High Energy Physics, Celestrial Mechanics and General Relativity Theory, KANT for Algebraic Number Theory, NQA for Group Theory, CoCoA, MACAULEY for Algebraic Geometry;

- Systems operating on user interface level as computer algebra calculators, and it is much obligated to the success of microprocessors (microcomputers), including systems muMATH, SiMATH, DERIVE (perhaps the most succesful system in high school education), etc. There are a few real Computer Algebra calculators, like TR-92, designed and produced by computer companies;
- General purpose systems, such systems are called “giants” between CAS, they are oriented to solve various problems from different topics of science and engineering. These systems are usually acting as a powerful CA calculators, incorporating also their own programming language to be used in case of complex problem. The most succesful commercial systems in this group, sharing the market worldwide, are MAPLE, MATHEMATICA, AXIOM (previous SCRATCHPAD), REDUCE, MATLAB, MATHCAD, MACSYMA;
- Systems being developed under the influence of the new emerging information technologies (particularly multimedia, Internet, JAVA), like MAGMA, MuPAD, as well as a few general purpose systems.

THE MOST POPULAR CAS

The widespread availability and power of computers and tremendous innovations in computer technology has greatly changed the way how scientific and engineering problems are to be solved. This begin also to change the way how to solve purely theoretical problems. The trend is expected to continue and to accelerate in the future.

As a consequence, a solid background to use CAS by active mathematicians, scientists and engineerers was created. The most popular CAS are sharing prevalence, as well as the standartization of:

- user interfaces,
- graphical representations,
- routes of data flow and the sequence of operations for data modelling.

Vice versa, most popular CAS sharing the market worldwide are differing mostly in algorithms (of data representation and internal realization of functions), not in such procedures as user interfaces, plotting of data, internal programming language. The market for CAS has to be segmented into:

- research;
- education;
- industry.

There are a few CAS, which are widespread in market segments mentioned above:

- The most popular CAS in research are AXIOM, MACSYMA, MAPLE, REDUCE, MACAULEY, MATHEMATICA, MATLAB, some others (specialized);
- In education – the most popular CAS are MAPLE, MATHEMATICA, DERIVE, MATLAB, MATHCAD;
- In industry – the most widespread CAS (or modelling of industrial processes) are MAPLE, MATHEMATICA, MACSYMA, MATLAB, MATHCAD.

AN EXAMPLE OF CAS: MACSYMA

The Macsyma computer algebra system [2] enables user to produce a meaningful and accurate computations in almost all the topics of mathematics, including arithmetics, algebra and geometry, calculus, special functions, linear algebra and matrices, vector and tensor analysis, statistics and data analysis, etc.

The ***ARITHMETIC*** expressions to be calculated or manipulated include:

- integer and rational numbers of any length, single, double, arbitrary precision floating point numbers;
- rational, algebraic, transcendental, complex numbers, etc.

For example, manipulations with complex numbers include finding expressions for realpart, imagpart, rectform, polarform, rational simplification of complex numbers, as wellas any composition of such manipulations. Any arithmetic expression also, can be computed as well as with arbitrary numbers, even ones like 1000!

Basic ***ALGEBRA*** operations include simplification, factoring and expansion of expressions, solutions of the systems of algebraic equations. To solve such equations Macsyma is using so-called "***Gröbner bases***", which can be viewed as a generalization of Euclid algorithm (cf. [3]). For a system of polynomial equations:

$$\begin{aligned} x_1^2 + x_2^2 + x_3^2 - 3 &= 0 \\ x_1 x_2 x_3 - 1 &= 0 \\ x_1 x_3 + x_2 - 2 &= 0. \end{aligned}$$

the Gröbner bases is found to be as the set of polynomials:

$$\{x_1 + x_2 x_3 - 1/2x_3^5 + 2x_3^3 - 7/2x_3, x_2^2 - 2x_2 + 1,$$

$$x_2 x_3^2 - x_2 + 1/2x_3^4 - 2x_3^2 + 3/2, x_3^6 - 3x_3^4 + 3x_3^2 - 1\}.$$

and this last set can be solved easily, from the last equation having $x_3 = \pm 1$. Solutions then are calculated by substituting these values in the first three equations:

$$(1,\ 1,\ 1)\ \ \&\ \ (-1,\ 1,\ -1)$$

Other algebraic operations include basic factoring of polynomials, as well as summation in closed form, with the finite sum, like:

$$\sum_{n=0}^{n} \left(n^2 + 2n\right) n!^2 = (n+1)^2 - 1$$

or with an infinite one:

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = \frac{\pi}{4}$$

Some operations in addition, cover exact symbolic solutions of systems of multi-variate equations, series and numerical ones, as well as inequalities, recurrence equations, numerical solutions, extremums of functions. The evaluation of complexity of algorithm like quicksort [4] by using Macsyma capabilities is a good example.

EXAMPLE. The quicksort algorithm is to sort data, and it uses a divide-and-conquer strategy in it's design. The nonrecursive part of the algorithm involves constructing subinstances through a partitioning technique, and this allows to establish recurrences for the average number of comparisons (C_N), which has to be evaluated:

$$C_N = N + 1 + \frac{1}{N} \sum_{1 \le j \le N} (C_{j-1} + C_{N-j}), \quad \text{for } N > 0. \qquad NC_N = N(N+1) + 2 \sum_{1 \le k \le N} C_{k-1} \quad \text{for } N \ge 1 \text{ with } C_0 = 0.$$

or

To calculate C_N's, the generating function could be used:

$$C(z) = \sum_{N \ge 0} C_N z^N$$

Making some manipulations with Macsyma for expressions given, the formula below can be changed and used for this recurrence to lead to a differential equation:

$$C'(z) = \frac{2}{(1-z)^3} + 2\frac{C(z)}{1-z}.$$

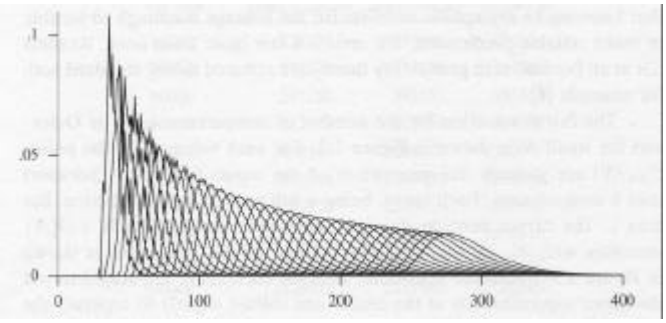
Solving this quation with Macsyma, the solution for generating function is found:

$$C(z) = \frac{2}{(1-z)^2} \ln \frac{1}{1-z}.$$

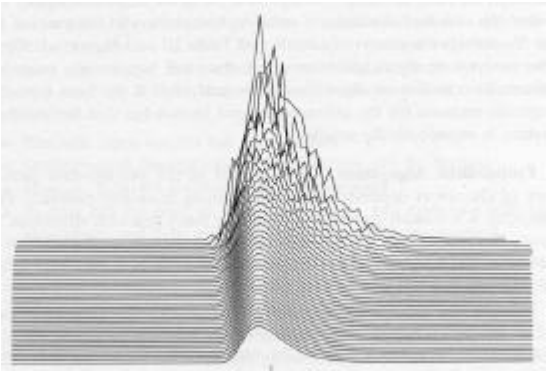
as well as for the average number of comparisons:

$$C_N = [z^N] \frac{2}{(1-z)^2} \ln \frac{1}{1-z} = 2(N+1)(H_{N+1} - 1).$$

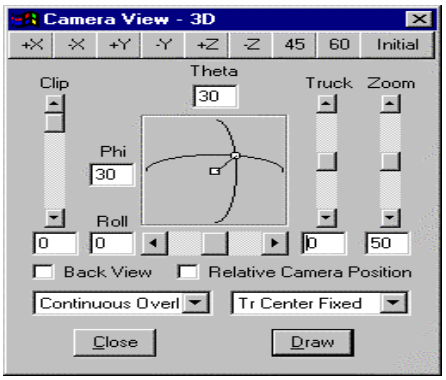
The final step in the analysis is to make useful plots of distribution of comparisons in quicksort. For each value of N, the points of $C_{Nk}/N!$ are plotted (the number $C_{Nk}/N!$ is the proportion of the inputs for which quicksort uses exactly k comparisons). In the plot to follow $10 < N < 50$:



The same plot scalled and translated to center, and separating the curves, look like:



The ***SCIENTIFIC GRAPHICS*** capabilities include 2D and 3D plots of data points, functions, implicit relations, parametric curves and surfaces, contours, vector fields. The plots are to be controlled by a specific interface, called camera view tool:



This interface involves two angles of rotation (theta corresponding to tilt, and phi corresponding to pan), zooming and tracking parameters, as well as a part of plot to be cipped. All these parameters can be controlled manually, by using mouse, and create an interactive viewing.

CALCULUS traditionally is the most reach part in operations of any of CAS, including Macsyma. It involves differential and integral calculi (differentiation and limits, analytic optimization, calculus of variations; Taylor and Laurent series, power series, Pade approximants), exact indefinite and definite integration (indefinite integration, definite integration; transforming integrals). It includes also numerical integration by using different methods, like Gauss quadrature, Romberg method, Newton-Cotes 8 panel quadrature rule, Simpson's rule, trapezoidal rule, other numerical quadrature methods. The useful group is presented by Laplace and Fourier transforms, and their inverses.

Differential and integral equation operations include exact, series and numerical solutions of first and second order O.D.E.s, symbolic systems of linear O.D.E.s, linear control problems, P.D.E.s, like Lie symmetries and solutions of nonlinear systems, generating input to PDEASE for finite element analysis, solving integral equations of the 1^{st} & 2^{nd} kinds.

VECTOR AND TENSOR CALCULUS cover complete facilities for vector and tensor calculus in coordinate-invariant form, and in specific coordinates.

STATISTICS AND DATA ANALYSIS in Macsyma has is a specific toolkit ***DataViewer*** for import/export, viewing, editing, graphing large numerical data sets, univariate and multivariate descriptive statistics, (non)linear multivariate least squares fit of data, polynomial, rational, spline interpolation of tabulated data, many probability densities and cumulative distributions, dimensional analysis and units conversion.

LINEAR ALGEBRA in Macsyma is supported by over 360 commands for many symbolic and numerical linear algebra operations, for sparse matrix facilities, special matrices, like Hadamard, Krylov, Pascal, Toeplitz, Vandermonde matrices, functions on matrices; all major normal forms and decompositions, eigenanalysis.

Macsyma has an add-on package *NUMKIT*, which speeds up real and complex floating point linear algebra and some other numerical operations, and the speed is comparable to that of FORTRAN libraries for affected operations. Macsyma uses NumKit for matrix and vector norms, determinants, matrix inversion, LU decomposition, SVD, eigenvalues and eigenvectors, numerical roots of polynomials, least squares fits, and solving underdeterrnined and overdetermined systems of linear equations. NumKit is based on LAPACK library for floating point linear algebra [5].

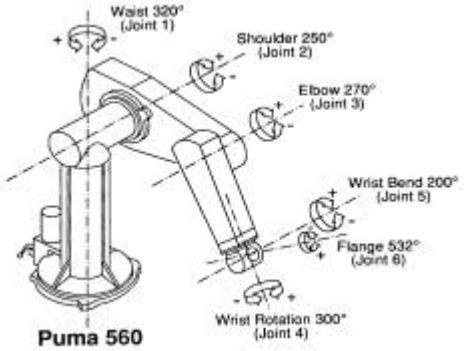
Special emphasis is made to normal forms: ***Cholesky decomposition*** (factorization of the numerical, square, positive definite Hermitian matrix ***M*** by lower triangular matrix ***L*** with ***LL* = M***); ***Hessenberg form*** (transforming square matrix ***A*** into upper Hessenberg form ***H***, an almost triangular matrix); ***Jordan form*** (over complex numbers, finite fields, algebraic extensions, transcendental extensions); ***LDU-decomposition*** (decomposing

square matrix ***N*** into lower triangular matrix ***L***, diagonal one ***D***, upper triangular matrix ***U***); ***LU-decomposition*** (decomposing square matrix into lower triangular and upper triangular matrix); ***QR-decomposition*** (decomposing numeric matrix into column unitary one and an upper triangular one); ***Schur form***; ***Singular Value Decomposition*** (SVD).

EXAMPLE. Control algorithms for robotics are computationally demanding. The physical motions of robots impose time constrains on the computation, as well as constrains to the tolerance of coordinates computed [6]. To make computations "in-time" and within accuracy needed the mixture of symbolic and numerical methods is used. The homogeneous coordinates, translations, rotations on various axis, are used to get a rigid body motion resulting:

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The position and orientation of a robot arm having six joints, each of one degree of freedom and used in many robotics research laboratories:



can be described by a matrix: $X = P\ T_6\ E$ where $T_6 = A_0\ A_1\ A_2\ A_3\ A_4\ A_5\ A_6$, and matrices A_i are varying only with joint angles, E is so-called tool transformation matrix, and ***P*** is a matrix which relates the coordinates and orientation of the robot arm base to the world coordinates.

These are the kinematic equations, and they enable user to consider the position and orientation of the robot arm as function of the controlled joint angles. The velocity of the end effector can be determined by differentiating this matrix with respect to time: $p = f(\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4, \dot{\theta}_5, \dot{\theta}_6)$ and the transformation of coordinates is:

$$\begin{bmatrix} d_x \\ d_y \\ d_z \\ \delta_x \\ \delta_y \\ \delta_z \end{bmatrix} = J(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) \begin{bmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 \\ d\theta_4 \\ d\theta_5 \\ d\theta_6 \end{bmatrix}$$

The inverted Jacobian ***J*** of the matrix above is used to solve inverse kinematics, to explore singularities in the motion of the robot-arm, as well as to get numeric values of the location and rotation of the arm.

SPECIAL FUNCTIONS covered by Macsyma consist of many special functions, some of which can be evaluated symbolically and numerically, some of which can be evaluated only numerically. Other special functions cannot be evaluated, but are used to express results of certain integrals and solutions of differential equations.

SYMBOLIC VERSUS NUMERIC COMPUTATION

Symbolic computation aims at the automation of the steps of mathematical problem solving that precede evaluating numerical models and that, to a large extent, are still the domain of human problem solvers. Symbolic computation is an emerging research tool in many areas of pure and applied mathematics, and in scientific applications. It is becoming a well-defined area for pure mathematics research.

Symbolic computation has broad support among the pure mathematicians. Its possible impact on pure mathematics is best described by example:

- In commutative algebra and algebraic geometry the CAS MACAULAY is used to generate non-trivial examples and gain the mathematical intuition, to formulate or gather evidence for conjectures, to provide or motivate steps in the proof of theorems or provide counter-examples to conjectures;
- In geometry and topology, CAS are necessary to explore the structure of hyperbolic 3-manilolds, to find implications of geometric rigidity in geometric group theory, to systematically enumerate and prove Thurston's geometrization conjecture of a large class of' 3-manifolds, to enumerate knots and links of 13-16 crossings, to have a beautiful graphical visualization techniques on Silicon graphics machines for triangulations, tilings and other spatially periodic structures in 3 and higher dimensions;

- In analysis, work is being done verifying and discovering q-series identities. These identities have both analytic and combinatorial aspects. Some of the identities in Ramanujan's notebooks fall into this category.

Symbolic and numeric computation enhance applied mathematics. Some of the scientific problems, such as those of weather prediction, turbulent combustion, and the mapping of the human genome, have been given a priority in the US research programs like the High Performance Computing Program, where they are identified as the ***grand challenges*** to be solved with future generations of computcr. Computational methods have gained wide acceptance in industry. For example, the Boeing 767 was designed fully on computers before wind tunnel models of it were built. The studies of the importance of computational modeling and numerical methods in mechanics outlines 14 key research areas, which have a substantial mathematical content, including Computational Fluid Dynamics and parallel computation. Other recent reports emphasize the role of computations for increasing competitiveness globally and for ensuring the security. Computational methods will in time, replace many traditional analytical methods in research and design.

Computational methods (both symbolic and numeric) are widely used in the life sciences. In computational chemistry, energy levels and molecular orbital structure can be computed, with results comparable to experimental measurement (when direct measurement is possible). Symbolic computation is useful in dealing with implicitly defined quantities (properties of eigenvectors of very large matrices, understanding how these cigenvectors change as a function of position on the potential hypersurface). Structural biology is another venue for large-scale computation, such as a computational approach to the protein folding problem. Both symbolic and numeric computation techniques are used in the human genome project, where similarities of the nucleotide code of long strings of DNA must be computed. Monte Carlo simulations of individual molecular structure (of large and small molecules) and of large systerms of molecules require both improved algorithms and numerical implementation of these algorithms.

Bifurcations, nonlinear dynamics and spatial-temporal chaos, as mathematical phenomena arise in diverse applications such as thermal convection systems, reaction-diffusion systems and nonlinear optics. These phenomena are being investigated by a combination of analytical methods, principally asymptotic analysis, and computational methods, including symbolic computation, numerical techniques and cellular automata. The main idea is to understand and predict the increasingly complex nonlinear behavior of the physical models as system parameters change.

Nonlinear scientific computation include a number of distinct projects related to parallel algorithms. These involve numerical linear algebra, dynamically adaptive meshes, and numerical optimization. The main idea is to improve computer methods for investigating natural and technological phenomena by designing new algorithms which provide qualitative advantages over existing algorithms, and which run efficiently on vector and parallel computer architectures.

Numeric computation is the development and implementation of computational algorithms. Algorithmic development includes the formulation of a computational method for solving a computational problem, and the mathematical analysis of the algorithm to ascertain whether the computed results will be a good approximation to the exact solution. Used together, symbolic and numeric computation enhance each other.

PDEASE is a software to solve partial differential equations, using the finite element method and closely related to Macsyma [7]. It combines graphical interface to provide a simple, yet powerful numerical approach to solve problems:

- To express physical problems and equations (geometry, boundary conditions, different materials properties, experimentally determined values or properties) with PDEase's input language;
- To solve problems numerically using automatic, robust grid refinement and timestep refinement algorithms to satisfy global or local error limits;
- To use Macsyma's capabilities for combining interactive graphics and animations, formatted text and live calculations in a notebook interface;
- To interpret the solution by comparing it with or using experimentally determined data that defines the problem; and
- To present results received in an high quality notebook, or publish them in printed or electronic form, or on the Internet.

COMPUTING NATURE OF CAS

CAS's are raising a lot of specifics from the software engineering point of view. The value of an algebraic variable may be an expression of arbitrary size and complexity, it leads usually to a structured object; these systems use memory more heavily than numerical programs; it is important to use dynamic memory allocation (to make best use of the hardware memory); the nature of the data is that expressions may be shortened and re-structured by the use of mathematical substitutions and simplifications (these may require user intervention, and leads to an interactive language); the variability of the data and the required operations leads to algorithmic and system design problems; a data representation or programming paradigm appropriate for one part of the system may not be suitable elsewhere (because of algorithmic complexity of sums or products of large numbers of terms); general-purpose CAS usually provide support for more than one programming method (e.g. functional and declarative programming) and may use more than one data representation (for example: the distributed and recursive representations of polynomials); the symbols used as names in a program may mathematically represent variables, and when used in such a way they can remain unassigned; mathematically, the corresponding variables of symbols can of course be assigned values, and this raises a problem of evaluation; the efficiency of expression's evaluation (especially full evaluation) is a problem, as this may result in time wasted manipulating long expressions which could equally well be represented, up to the last step of calculation.

The types of design difference make performance comparisons between systems treacherous: ***a class of problems may run well in one system and not another.***

Even on problems of the same type, performance can depend crucially on the size or structure of the data: ***even renaming the variables may have significant timing effects.***

Each system has some areas where it excels. Users are advised to try problems of the class they really want to solve before choosing.

Computer algebra systems are ***big programs***. Correspondingly, some systems have very large executables, and cannot be used at all on small computers, though others, were designed to have a small kernel and loadable modules.

Almost all systems have some such demand-loadable parts, and are designed in a modular way, and almost all provide a high-level programming language in which extensions to the distributed system can be written. Scoping rules vary significantly between systems and can create pitfalls in converting from one system to another.

Although computer algebra systems can handle problems much larger than those for which hand calculation is feasible, the comparison of scales is not as great as one might

hope: one can handle expressions and calculations greater by a factor of about 10^4 and one of reasons is that the complexity of some operations naturally grows exponentially.

CAS ON INTERNET

Many activities related to CAS are presented on Internet, as it is common now for many fields of human liveliness. There are special networks, projects, descriptions of research and applications, and journals, regular series of conferences, books, etc. A few of them are reflected below:

- Computer Algebra Information Network: *http://www.can.nl/*
- Symbolic Computation Network: *http://symbolicnet.mcs.kent.edu/*
- OPENMATH Project: *http://www.openmath.org*
- ATLAST Project: *http://www.umassd.edu/SpecialPrograms/Atlast/welcome.html*
- Numerical Algorithms Group (NAG) *http://www.nag.co.uk/*
- International DERIVE USER GROUP *http://www.derive.com/dugappli.htm*
- 'Journal of Symbolic Computation' *http://www.hbuk.co.uk/ap/journals/sy/*
- 'Applicable Algebra in Engineering, Communication and Computing' *http://link.springer-ny.com/link/service/journals/00200/index.htm*
- the International Symposia in Symbolic and Algebraic Computation (ISSAC) *http://www.inria.fr/safir/whoswho/Stephen.Watt/issac/index.html*
- Bulletin of the Special Interest Group SIGSAM *http://pineapple.apmaths.uwo.ca/~rmc/sigsam/*
- AXIOM *http://www.nag.co.uk/symbolic/AX.html*
- DERIVE *http://www.derive.com/*
- MACSYMA *http://www.macsyma.com/*
- MAPLE *http://www.maplesoft.com/*
- MATHEMATICA *http://www.wolfram.com/*
- MATLAB *http://www.mathworks.com/*
- REDUCE *http://www.rrz.uni-koeln.de/REDUCE/*

Other relevant conference series: AAECC meetings, CADE (differential equations), CoCoA on commutative algebra, DISCO (Design and Imple- mentation of Symbolic Computation), IMACS-SC and IMACS-ACA, MEGA on algebraic geometry, and PASCO for parallel symbolic computation.

REFERENCES

1. B.Buchberger, G.E.Collins, R.Loos, *Computer Algebra: symbolic and algebraic computation*, **2**nd ed. Wien: Springer-Verlag, 1983.

2. *The Macsyma User's Guide*, **2**nd ed. Arlington, Mass.: Macsyma Inc. 1996.

3. R.Fröberg, *An Introduction to Gröbner Bases*, Chichester: John Wiley and Sons, 1997.

4. Ph.Flajolet, R.Sedgewick, *An Introduction to the Analysis of Algorithms*. Addison-Wesley Publishing Co., Reading, MA, 1996.

5. http://www.netlib.org/lapack/lug/lapack_lug.html.

6. W.M.Gentleman, "Case Studies of Real-time Processing in Robotics". *NATO ASI Series, E: Applied Sciences*, **v.**232, p.165-182, 1994.

7. *PDEase Documentation Set*, Arlington, Mass.: Macsyma Inc. 1997.