Computer Science Journal of Moldova, vol.**5**, no.1(13), 1997

Singular A Computer Algebra System for Singularity Theory, Algebraic Geometry and Commutative Algebra

G.-M. Greuel G. Pfister H. Schönemann

1 Overview

Singular is a computer algebra system designed for special needs in commutative algebra and algebraic geometry, with special features for singularity theory.

It is avaiblable via anonymous ftp from
`ftp://www.mathematik.uni-kl.de/pub/Math/Singular/bin`.

Many ground fields are supported to build rings and free modules over them. Singular can compute with ideals and modules generated by polynomials or polynomial vectors over polynomial or power series rings or, more generally, over the localization of a polynomial ring with respect to any monomial ordering. The operations include several combinatorial algorithms for computing dimensions, multiplicities, Hilbert series ….

Although Singular has a very general standard basis algorithm (including computations in local or mixed global-local rings), it is quite fast for computations of Gröbner bases for, say, homogeneous ideals in polynomial rings over fields of small characteristic and it seems to belong to the fastest systems for computations over the rationals. Nevertheless, the local and mixed global-local case is a speciality of Singular and it offers several different strategies for these cases. Those which have shown most efficient in our tests are the default strategies, but the user has the option to choose different ones. We explain some of the heuristics in section 5.

A programming language, which is C-like and which is quite comfortable and has the usual if-else, for, while, break ... constructs allows to extend the standard set of operations: the provided library of procedures, written in the SINGULAR language, are useful for many applications to mathematical problems.

SINGULAR provides also links to communicate with other systems or with itself. This communication is based on MP (by S. Gray, N. Kaijler, P. Wang) and MPP ([BGS]), a protocoll designed for both a very general and a very efficient exchange of polynomial data.

2 Principles of SINGULAR

SINGULAR is a specialized system for commutative algebra, algebraic geometry, singularity theory, therefore its main algorithms are very general standard basis (Gröbner basis) algorithms. These algorithms are designed for speed, space complexity is secondary to time complexity.

The basic operations are provided by the kernel (written in C/C++). To extend these operations SINGULAR has an intuitive mathematical language, user friendly for mathematicians. This programming language allows the constructions of libraries.

SINNGULAR can communicate with other systems (specialized or general purpose) for solving of subproblems and can also act as a compute server for other programs.

3 Ground Fields and base rings

Many ground fields are supported, such as the rational numbers Q (char 0), Z/p (p a prime ≤ 32003 (char p)), finite fields with $q = p^n$ elements ($p^n \leq 2^{15}$), transcendental and algebraic extension $K(A, B, C, \ldots)$, $K = Q$ or Z/p, and floating point (single precision) real numbers. For allmost all computations SINGULAR requires a base ring. Fixing such a ring allows efficient data structures for polynomials (the basic objects): type information is stored only once, no need for runtime type checking.

Computations over the following baserings are possible: a polynomial ring, a series ring (i.e. a localization of a polynomial ring), a factor ring by an ideal of one of the above, exterior algebra, Weyl algebra/D-modules, and tensor products of one of the above.

4 Main algorithms

- The basic algorithm in SINGULAR is a general standard basis algorithm for any monomial ordering which is compatible with the natural semigroup structure of the exponents. This includes wellorderings (Buchberger algorithm, "global case") and tangent cone orderings (Mora algorithm,"local case").
- Advanced techniques such as Traverso's Hilbert–driven Gröbner basis algorithm or the weighted–ecart–method and the high–corner–method, developed by the SINGULAR group.
- Algorithms for the usual ideal theoretic operations, such as intersection, ideal quotient, elimination and saturation.
- Different syzygy algorithms (sres, res, mres) for computation of free resolutions of modules over the above rings.
- Combinatorial algorithms for computing dimensions, Hilbert series, multiplicities, etc.
- Factorization of univariate/multivariate polynomials over the rationals and algebraic extensions
- Algorithms for primary decomposition of ideals and modules
- Algorithms for computing invariants of singularities or affine or projective varieties such as Milnor, Tjurina and discriminate numbers, singular locus, T1 and T2, Ext groups etc.
- Algorithms for computing a semiuniversal deformation of an isolated singularity or a module up to a given order

5 Strategies

It has turned out that the standard basis algorithm is, in the local and mixed global-local case, extremely sensitive to the choice of the strategy, much more than in the classical global and homgeneous case. We state conclusions for standard bases computations in this case: (for a description of the strategies, the timings and examples cf. [SBS])

1. The test for the "highest corner" is essential for 0–dimensional ideals and does not cost much time in other cases.
2. The ecartMethod applies to any ideal and can be extremely fast if one has a good feeling for the weights. In practice this option is most useful if the system automatically offers an ecart vector w.
3. If the ecartMethod is not successful, it is usually recommended to use the option morePairs.
4. If the above options are not successful one should use morePairs and sugarCrit.
5. The sugar option is generally good and is a default option in SINGULAR.
6. Lazard's method is in general slower than the other options, although it is sometimes surprisingly fast. In general it seems to be least a safe option.
7. The lex ordering is usually the fastest. Since it is **not** an elimination ordering in the local case, its use is very limited (e.g. the dimension is computed correctly but not the multiplicity).
8. the algorithm of Schreyer (sres in SINGULAR)for compting free resolutions (cf. [S1], [S2]) can be very fast for general ideals (e.g. generic determinantal ideals)

Moreover, for computations in characterstic zero or over algebraic extensions the growth of the coefficients has to be taken into account in the strategies.

6 Libraries

SINGULAR contains several libraries, with procedures written in the SINGULAR language. The user may add his own procedures/libraries. At the moment the following libraries are distributed with SINGULAR:

inout.lib	procedures for manipulating in– and output
general.lib	procedures of general type
deform.lib	procedures for computing miniversal deformation
elim.lib	procedures for elimination, saturation and blowing up
factor.lib	procedures for calling the REDUCE factorizer (UNIX only)
homolog.lib	procedures for homological algebra
matrix.lib	procedures for matrix operations
poly.lib	procedures for manipulating polys and ideals
random.lib	procedures for random matrix and poly operations
ring.lib	procedures for manipulating rings and maps
sing.lib	procedures for singularities

In one of the next versions we shall also add the libraries

solve.lib	procedures for solving polynomial equations
invar.lib	procedures for computing invariants of algebraic groups
primedec.lib	procedures for primary decomposition

7 Links

Links describe the communication channels of SINGULAR, i.e. something SINGULAR can read from and/or or write to. There are the following links: Acii, DBM nad MP.

Via Ascii links any data can be written. Data exchange via ascii links is not the fastest way, but the most general. DBM links provide access to data stored in a data base. MP links encode data in a binary format which may be send to other programs for evaluation or to disc for storing.

Based on these links, SINGULAR is able to communicate with other systems or with itself. At the moment a communication is implemented between SINGULAR - SINGULAR, SINGULAR - FACTORY (a C++ library for multivariate polynomials) SINGULAR - Mathematica and, on a lower level, SINGULAR - Maple.

8 What will come?

- A factorizing Gröbner algorithm (for solving systems of polynomial equations)
- A library realizing the Arnold classifier of singularities
- Graphical interface (A test version with these topics realized exists already.)
- FGLM–techniques for fast change of ordering
- Trace algorithm for speeding up computations in characteristic 0
- Links to other general purpose systems (Maple, Macsyma, ...)
- Puiseux–expansion of plane (and space) curve singularities
- Special functions allowing very fast computation in semigroup rings
- Compilation of libraries

References

[BGS] Bachmann, O.; Gray, S.; Schönemann, H.: MPP: A Framework for Distributed Polynomial Computations. Accepted for ISSAC 1996.

[SBS] Grassmann, H.; Greuel, G.-M.; Martin, B.; Neumann, W.; Pfister, G.; Pohl, W.; Schönemann, H.; Siebert, T.: On an implementation of standard bases and syzygies in SINGULAR. To be published in Methods of Lie Theory (AAECC).

[G] Greuel, G.-M.: Description of SINGULAR, a Computer Algebra System for Singularity Theory, Algebraic Geometrie and Commutative Algebra. Euromath Bulletin, Vol. 2 No. 1, 1996

[GP] Greuel, G.-M.; G.Pfister: Advances and improvements in the theory of standard bases and syzygies. Arch. Math. **66**, 163 – 1796 (1996).

[GPS] Greuel, G.-M.; Pfister, G.; Schönemann, H.: SINGULAR Reference Manual. Available via ftp://www.mathematik.uni-kl.de/pub/Math/Singular

[NS] Nüßler, T.; Schönemann, H.: Gröbner bases in algebras with zerodivisors. Preprint, Univ. Kaiserslautern, 1993.

[Pf] Pfister, G.: The tangent cone algorithm and some applications to algebraic geometry. Proc. of the MEGA Conference 1990. Birkhäuser Vol. 94 (1991).

[Po] Pohl, W.: About the weighted ecart method for computing standard bases. Preprint, Univ. Kaiserslautern, 1994.

[PS] Pfister, G.; Schönemann, H.: Singularities with exact Poincarè complex but not quasihomogeneous. Revista Matematica Madrid (2) 2y3, 161 -171 (1989).

[S1] Siebert, T.: On strategies and implementations for computations of free resolutions. Proc. of the POSSO Meeting, March 1995, Paris.

[S2] Siebert, T.: How to compute syzygies. Preprint.

G.-M. Greuel, G. Pfister, H. Schönemann,
Department of Mathematics
University of Kaiserslautern
Kaiserslautern, Germany
e–mail: $\{greuel, pfister, hannes\}@mathematik.uni-kl.de$

Received November 11, 1996