SINGULAR 2-0
A COMPUTER ALGEBRA SYSTEM FOR POLYNOMIAL COMPUTATIONS

G.-M. GREUEL, G. PFISTER, AND H. SCHÖNEMANN

1. Introduction

1.1. **Aims and scope of Singular.** Singular is a Computer Algebra system for polynomial computations with emphasis on the special needs of commutative algebra, algebraic geometry, and singularity theory. However, Singular is not restricted to these areas, and it has been used in various non-mathematical applications.

Singular's main computational objects are ideals and modules over a large variety of baserings. The baserings are polynomial rings or localizations thereof over a field (e.g., finite fields, the rationals, arbitrary length floating points, algebraic extensions, transcendental extensions) or quotient rings with respect to an ideal.

Singular features one of the fastest and most general implementations of various algorithms for computing Gröbner, respectively standard bases, with respect to arbitrary monomial orderings, developed by the authors. The implementation includes Buchberger's algorithm (if the ordering is a well ordering) and Mora's algorithm (if the ordering is a tangent cone ordering) as special cases. Furthermore, it provides polynomial factorizations, resultant, characteristic set and gcd computations, syzygy and free-resolution computations, parametrization of curves, resolution of arbitrary singularities, ring normalization and computation of invariant rings, and many more related functionalities. Using triangular sets, Singular provides also symbolic-numerical algorithms for solving polynomial systems.

Moreover, the system extension Singular:Plural provides several algorithms related to Gröbner basis calculations in "polynomial like" non-commutative rings. In order not mix with the commutative world, Plural has its own description.

Based on an easy-to-use interactive shell and a C-like programming language, Singular's internal functionality is augmented and user-extendible by libraries written in the Singular programming language. A general and efficient implementation of communication links allows Singular to make its functionality available to other programs.

The original motivation for the authors to develop a computer algebra system was the need to compute invariants of ideals and modules in local rings. In the sequel, the development of Singular was always influenced by concrete problems coming either from mathematics or from applications.

1.2. **Availability of Singular.** SINGULAR is free software (cf. Section 4.1 for detailed license information). It is available for various platforms:

- Windows 95/98/ME/NT/2K/XP,
- Unix: Linux, HP-UX, Solaris, IRIX, AIX, OSF, FreeBSD,
- MacOS X: Macintosh PPC

It can be downloaded from its homepage

`http://www.singular.uni-kl.de`

2. Internal structure and interfaces of Singular

SINGULAR consists of a kernel written in C/C++, and libraries written in SINGULAR's programming language. These libraries augment the internal functionality provided by the kernel and are easily changed and extended by the user.

2.1. **User Interfaces and the Singular language.** SINGULAR can either be run in an ASCII-terminal or within Emacs (or XEmacs).

SINGULAR interprets commands given interactively on the command line as well as given in the context of user-defined procedures. In fact, SINGULAR makes no distinction between these two cases. Thus, SINGULAR offers a powerful programming language as well as an easy-to-use command line interface without differences in syntax or semantics.

In many aspects, the SINGULAR language is similar to the C programming language. The major building blocks of the SINGULAR language are expressions, commands, and control structures. In particular, the SINGULAR libraries are written in this language.

2.2. **Communication Links.** A general and efficient implementations of communication links allows SINGULAR to make its functionality available to other programs. This includes

- ASCII-link provided via files/sockets (GAP, Maple),
- MP-link (Mathematica, MuPAD),
- connection to C/C++ libraries via dynamic modules,
- interface to the visualization program surf (currently available only for Linux PC's and SUN workstations).

3. Mathematical functionality

3.1. **Algebraic objects in Singular.** For almost all computations to be carried out in SINGULAR a ring has to be defined first. If no ring has been created, basically only integer and string operations are available.

SINGULAR offers very many different commutative and a class of non-commutative rings. In the following, ring always means a *commutative ring.* The current ring in use is called *basering.*

The rings in SINGULAR are either

- polynomial rings over a field,
- localizations of polynomial rings, or
- quotient rings with respect to an ideal.

All rings come equipped with a monomial order. Ideals are represented as lists of polynomials which generate the ideal.

The basic fields in SINGULAR are prime fields, determined by their characteristic, and the rational numbers. Furthermore, other finite fields, algebraic extensions and transcendental extensions, and the real and complex numbers are also implemented in SINGULAR.

Modules in SINGULAR are submodules of a free module over a basering R, given by lists of vectors generating the submodule. A quotient module M of a free module F is represented in SINGULAR by a submodule U of F such that $M = F/U$. In this way, any finitely generated R-module can be represented in SINGULAR by its module of relations.

3.2. **Algorithms in the Singular kernel.** The basic algorithm in SINGULAR is a general standard basis algorithm for any monomial ordering which is compatible with the natural semi-group structure of the exponents.

The following algorithms are implemented in the SINGULAR kernel:

Standard basis algorithms: Implementation of a standard basis algorithm for arbitrary monomial orderings, in particular

- Implementation of Buchberger's algorithm (for well-orderings), and
- Implementation of the Mora algorithm or tangent cone algorithm (for local orderings);
- Implementation of Traverso's Gröbner basis algorithm using the Hilbert function;
- Implementation of the weighted ecart method;
- Implementation of the highest corner method;
- Computation of a list of Gröbner bases via the Factorizing Gröbner Basis Algorithm, i.e., their intersection has the same radical as the original ideal;
- Computation of a Gröbner basis of a zero-dimensional ideal via the FGLM (Faugere, Gianni, Lazard, Mora) algorithm. Assumes that a reduced Gröbner basis with respect to another ordering is given;
- Implementation of the Gröbner walk algorithm, the fractal walk algorithm, and the Gröbner fan.

Ideal/Module Theory: Computation of

- intersections of ideals and modules, ideal quotients,
- elimination and saturation of ideals and modules,
- radical membership,
- preimages of ideals under ring maps, kernel of a map,
- dimensions of ideals and modules.

Syzygies and free resolutions of modules: Computation of

- free resolutions of ideals or modules with the standard basis method
- free resolutions of homogeneous ideal or module with La Scala's method
- minimal free resolutions of an ideal or module with the Syzygy method
- first syzygy modules (i.e., the module of relations of the given generators)
- free resolutions of a n ideal or module with Schreyer's method (input has to be a standard basis)
- the Castelnuovo Mumford regularity
- graded Betti numbers of a module from a free resolution

Combinatorics: Computation of

- (Krull) dimension, codimension and multiplicity
- computation of vector space basis (consisting of monomials) of the quotient of a ring by an ideal, respectively of a free module by a submodule
- Hilbert series, Hilbert function

Polynomial computations: Computation of

- greatest common divisors (gcd) of univariate and multivariate polynomials
- resultant of two univariate polynomials using the subresultant algorithm
- factorization of univariate and multivariate polynomials into irreducible factors
- characteristic sets of polynomial ideals
- interpolation of polynomials from values at several points

Matrix computations.

- Implementation of the sparse Gauss-Bareiss method for elimination (matrix triangularization) in arbitrary integral domains
- Computation of the determinant of a square matrix, for integer matrices a modular algorithm is used, for other domains the Gauss-Bareiss method is used.

Numeric computations.

- Computation of all (complex) roots of a univariate polynomial via the Laguerre method,
- Finding all roots of a 0-dimensional ideal via triangular sets.

3.3. **Additional libraries.** In addition to the kernel functions, Singular offers libraries for computations in singularity theory, commutative algebra, invariant theory, symbolic-numerical solving, visualization, linear algebra, and coding theory.

1. Singularity Theory.

- `classify.lib`:
  Classification of isolated hypersurface singularities (by Arnold's list).
- `deform.lib`:
  Compute versal deformations of isolated singularities.
- `gaussman.lib / mondromy.lib / gmssing.lib`:
  Libraries to compute Hodge-theoretic invariants of isolated hypersurface singularities: Monodromy, V-Filtration and weight filtration, spectral pairs and spectral numbers, Bernstein polynomial, Gauss-Manin connection.
- `hnoether.lib`:
  A library for computing the Hamburger-Noether expansion, respectively Puiseux expansion, of a plane curve singularity following Campillo. The library contains also procedures for computing the topological numerical invariants of plane curve singularities (delta invariant, Puiseux pairs).
- `sing.lib`:
  A library for computing invariants associated to singularities, e.g. Milnor number, Tjurina number, T1, T2, tangent cone. Includes also a computation of the singular locus of an ideal.
- `spcurve.lib`:
  Deformations and Invariants of Cohen-Macaulay codimension 2 singularities: Kernel of the Kodaira-Spencer map, versal deformations of space curve singularities, test for quasi-homogeneity.
- `alexpoly.lib`:
  A library for computing the resolution graph of a plane curve singularity, the total multiplicities of the total transforms of the branches along the exceptional divisors of a minimal good resolution, the Alexander polynomial, and the zeta function of its monodromy operator.
- `resol.lib / zeta.lib`:
  Resolution of singularities based on the algorithm by Bravo, Encinas, Villamayor. Computation of the intersection matrix of the exceptional divisors, negative spectral numbers, the Denef-Loeser zeta function from the resolution data, and the genera of the exceptional divisors.
- `equising.lib`:
  Computation of the equisingularity stratum of a family of plane curves and the equisingular Tjurina number.

2. Commutative algebra.

- `primdec.lib`:
  A library for computing
  - a primary decomposition based on Gianni, Traeger, Zacharias and on Shimoyama, Yokoyama;
  - the radical based on the ideas of Krick, Logar and Kemper;
  - the minimal associated primes using characteristic sets.

- `normal.lib`:
  Library for computing the normalization of a ring, the genus, and the global delta invariant.
- `elim.lib`:
  Library dealing with elimination, saturation, projection, and blowing up.
- `homolog.lib`:
  Computation of Hom, Ext, Tor, Massey-products, versal deformation of modules, Koszul homology, regular sequences, depth, Flattening stratification, Tensor product of modules, Fitting ideals.
- `mregular.lib`:
  A library for computing the Castelnuovo-Mumford regularity of a subscheme of the projective space that does not require the computation of a minimal graded free resolution of the saturated ideal defining the subscheme.
- `reesclos.lib`:
  A library to compute the integral closure of an ideal in a polynomial ring using the Rees Algebra.

3. Invariant Theory.

- `finvar.lib / ainvar.lib / rinvar.lib`:
  Invariant rings: libraries for computing polynomial invariants of finite matrix groups, respectively additive groups, respectively reductive groups, and generators of related varieties (in the modular and non-modular case): Reynolds operator, Molien series, Primary and secondary invariants.
- `qhmoduli.lib`:
  Moduli spaces for semi-quasihomogeneous isolated hypersurface singularities.
- `stratify.lib`:
  Algorithmic stratification with respect to the action of a unipotent group (based on Greuel, Pfister).

4. Solving.

- `solve.lib / triang.lib`:
  A library for finding real (or complex) solutions to polynomial systems (with parameters): symbolic-numerical solving, Laguerre solver, sparse multivariate resultants, u-resultants (Gelfand, Kapranov, Zelevinsky), decomposition of zero-dimensional ideals into triangular sets (Moeller, Lazard).

5. Visualization.

- `latex.lib`:
  Library for typesetting Singular objects in LaTeX.
- `surf.lib`:
  Interface to the visualization program surf for plotting curves and surfaces (only for Linux PCs and Sun workstations).

6. Linear Algebra.

- `matrix.lib / linalg.lib`:
  Procedures for elementary matrix operations (e.g. row and column operations), Gauss elimination, inverse matrix, characteristic polynomial, eigenvalues and eigenvectors, spectrum, adjoint matrix, Gram-Schmidt orthogonalization, and Hessenberg and Jordan normal form.

7. Others.

- `brnoeth.lib`:
  Implementation of the Brill-Noether algorithm for solving the Riemann-Roch problem and applications in Algebraic Geometry codes. The computation of Weierstrass semigroups is also implemented.
- `intprog.lib`:
  Integer Programming, Toric Varieties, Applications of the Buchberger algorithm to binomial ideals.

4. License information, documentation, and help system

4.1. **License information.** This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation (version 2 of the License); with the following additional restrictions (which override any conflicting restrictions in the GPL):

The following software used with Singular have their own copyright: the omalloc library, the readline library, the Gnu Multiple Precision Library (GMP), NTL: A Library for doing Number Theory (NTL), the Multi Protocol library (MP), the Singular-Factory library, the Singular-libfac library, and, for the Windows distributions the Cygwin DLL and the Cygwin tools (Cygwin), and the XEmacs editor (XEmacs).

4.2. **Documentation.** The complete user manual and a quick reference card can be downloaded from the Singular homepage in different formats such as DVI, PostScript, HTML, PDF, and TeXinfo. The HTML version can be viewed online and, moreover, there is a keyword search.

The book "A Singular Introduction to Commutative Algebra" (by G.-M. Greuel and G. Pfister, with contributions by O. Bachmann, C. Lossen and H. Schönemann) illustrates the usage of Singular by means of concrete problems from algebraic geometry, singularity theory and commutative algebra. It has appeared at the Springer Verlag.

4.3. **Help system.** The on-line help system provides information on what commands are available and how to use them. The whole manual is available online by typing the command `help;`.

The help information is displayed using the default help browser. Platform depending, this may for instance be a web browser. Type `help browsers;` for a list of the supported browsers and for information on how to switch to another browser.

5. Contributors and Acknowledgements

The development of SINGULAR is directed and coordinated by Gert-Martin Greuel, Gerhard Pfister, and Hans Schönemann.

Currently, the SINGULAR team has the following members: Anne Frühbis-Krüger, Thomas Keilen, Kai Krüger, Christoph Lossen, Viktor Levandovskyy (PLURAL), Wilfred Pohl, Mathias Schulze, and Eric Westenberger.

Past members of the SINGULAR team are: Olaf Bachmann, Hubert Grassmann, Wolfgang Neumann, Jens Schmidt, Rüdiger Stobbe, Tim Wichmann.

Further contributions to SINGULAR were made by: Thomas Bayer, Isabelle Bermejo, Stephan Endrass, Jose Ignacio Farran Martin, Wolfram Decker, Philippe Gimenez, Christian Gorzel, Agnes Heydtmann, Dietmar Hillebrand, Tobias Hirsch, Martin Lamm, Bernd Martin, Michael Messollen, Thomas Nüssler, Moritz Wenk.

We should like to acknowledge the financial support given by the Volkswagen-Stiftung, the Deutsche Forschungsgemeinschaft and the Stiftung für Innovation des Landes Rheinland-Pfalz to the SINGULAR project.

©1986-2004 G.-M. Greuel, G. Pfister, and H. Schönemann