SINGULAR – A Computer Algebra System for Polynomial Computations

G.M. Greuel, G.Pfister, H. Schönemann
Fachbereich Mathematik; Universität Kaiserslautern
67653 Kaiserslautern; Germany
email: [greuel,pfister,hannes]@mathematik.uni-kl.de

September 23, 2000

SINGULAR is a specialized computer algebra system for polynomial computations with emphasize on the needs of commutative algebra, algebraic geometry, and singularity theory.

Main functionality. SINGULAR's main computational objects are polynomials, ideals and modules over a large variety of rings. SINGULAR features one of the fastest and most general implementations of various algorithms for computing standard resp. Gröbner bases. Furthermore, it provides multivariate polynomial factorization, resultant, characteristic set and gcd computations, syzygy and free-resolution computations, numerical root–finding, visualisation, and many more related functionalities.

Based on an easy-to-use interactive shell and C-like programming language, SINGULAR's internal functionality is augmented and user-extendable by libraries written in the SINGULAR programming language or in C++. A general and efficient implementation of links as endpoints of communications allows SINGULAR to make its functionality available to and be easily incorporated into other programmes.

The main goal of the SINGULAR–group is to further develop and implement *advanced* algorithms to be used for mathematical research, in particular in commutative algebra, algebraic geometry and singularity theory.

Indeed there exist already several libraries providing such algorithms, including full primary decomposition for several ground fields, ring normalization (integral closure), versal deformations of arbitrary isolated singularities, monodromy and spectral numbers for hypersurface singularities, Hamburger–Noether (Puiseux)–expansions of plane curve singularities and many more. Most of these algorithms are not available in any other system.

Recently, due to nonmathematical applications of SINGULAR, we are experimenting with symbolic–numerical polynomial solving.

As a specialized system, SINGULAR's aim is not to provide all the functionality of a general purpose system. The main strength of the system, besides the above mentioned functionality, is the speed of the important basic algorithms such as Gröbner basis, syzygy, and free resolution computations for modules. It is impossible to detail any of the algorithms, we rather refer to the literature, given in the references.

SINGULAR's online help system is available in various formats where the HTML format is especially user-friendly.

Availability. SINGULAR is publicly available as a binary program for all common Unix platforms including LINUX, for Windows 95/98/NT and for MacOS. The current version number is 1.3.8. It can be downloaded by anonymous ftp from `ftp://www.mathematik.uni-kl.de/pub/Math/Singular` or per WWW from `http://www.singular.uni-kl.de/distribution.html`

Besides the executable SINGULAR program, the distribution contains the source code of all SINGULAR libraries, the user manual (resp. tutorial) in various formats (PostScript, info, and HTML) and some utility programs (for visualisation etc.).

Moreover, on request, we provide the source code of the FACTORY library for multivariate polynomial gcd, resultant, and factorization which is part of SINGULAR but may be used by other systems (as it is e.g., by Macaulay 2).

For more and always up-to-date information, SINGULAR's home page can be reached at `http://www.singular.uni-kl.de`

Mathematical Features. SINGULAR's primary computational objects are ideals resp. modules which are generated by polynomials resp. polynomial-vectors over polynomial rings or, more generally, over the localization of a polynomial ring with respect to any ordering on the set of monomials which is compatible with the semigroup structure.

Supported baserings include:

- polynomial rings with a large variety of polynomial orderings (common simple, block, elimination, weighted, and general matrix orderings),
- localization of a polynomial ring at a prime ideal generated by a subset of the variables
- factor rings by an ideal of one of the above,
- rings of tensor products of one of the above.

Supported ground fields for these rings include:

- rational numbers (of arbitrary length) Q,
- finite fields Z/p (where p is a prime ≤ 32003),
- Galois fields (finite fields with $q = p^n \leq 2^{15}$ elements),
- transcendental extensions ($K(A, B, C, \ldots)$, $K = Q$ or Z/p), The
- algebraic extensions ($K[t]$/minimal-polynomial, $K = Q$ or Z/p), and
- floating point real and complex numbers with arbitrary predefined precision.

main algorithms implemented in SINGULAR are:

- general standard basis algorithm for *any* monomial ordering which is compatible with the natural semi-group structure of the exponents. This includes well-orderings (Buchberger algorithm) and tangent cone orderings (Mora algorithm) as special cases,
- Hilbert–driven Gröbner basis algorithms, weighted–ecart–method and high–corner–method, FGLM algorithm for change of ordering,
- factorizing Buchberger algorithm,
- intersection, quotient, elimination and saturation of ideals,
- Schreyer's, La Scala's and Siebert's algorithm for computations of syzygies and free resolutions of modules,
- combinatorial algorithms for computations of dimensions of factor rings, Hilbert series and multiplicities of modules,
- multivariate polynomial gcd, resultant, and factorization algorithms,
- Wang's algorithm to compute characteristic sets.

See [GGM], [GP96], [GPS98], [GP98] for more details on the implemented algorithms.

Computational Features. SINGULAR has a convenient and intuitive interactive user interface (shell) which has key-bindings similar to those of Unix' `tcsh` shell. Alternatively, an Emacs mode provides the possibility to run SINGULAR within an Emacs/XEmacs window. SINGULAR's user interface provides both, access to SINGULAR's mathematical functionality and a convenient, powerful, and C-like programming language (strongly typed, and lexicographically scoped) which includes all the usual programming constructs (like loops, procedures, local/global variables, etc). Based on this programming language, users may extend SINGULAR's functionality by writing their own libraries. Libraries may be written in the SINGULAR programming language or in C++.

At the moment, the SINGULAR distribution includes the following libraries:

standard.lib	extensions of Singular kernel

General purpose

all.lib	load all other libraries
general.lib	procedures of general type
inout.lib	procedures for manipulating in- and output
poly.lib	procedures for manipulating polynomials and ideals
random.lib	procedures of random/sparse matrix and poly operations
ring.lib	procedures for manipulating rings and maps

Linear algebra

matrix.lib	procedures for matrix operations
jordan.lib	procedures to compute the jordan normal form
linalg.lib	procedures for algorithmic linear algebra

Commutative algebra

algebra.lib	procedures for computing with algebras and maps
elim.lib	procedures for elimination, saturation and blowing up
homolog.lib	procedures for homological algebra
mregular.lib	procedures for Castelnuovo-Mumford regularity
normal.lib	procedures for normalization
primdec.lib	procedures for primary decomposition
primitiv.lib	procedures for finding a primitive element
intprog.lib	procedures for integer programming
toric.lib	procedures for toric ideals

Singularities

classify.lib	procedures for the Arnold-classifier of singularities
deform.lib	procedures for computing miniversal deformation
hnoether.lib	procedures for Hamburger-Noether (Puiseux) development
mondromy.lib	procedures to compute the monodromy of a singularity
sing.lib	procedures for computing invariants of singularities
spcurve.lib	procedures for Cohen–Macaulay codimension 2 singularities

Invariant theory

finvar.lib	procedures to calculate invariant rings of finite groups
ainvar.lib	procedures for invariant rings of the additive group

Symbolic-numerical solving

presolve.lib	procedures for pre-solving polynomial equations
solve.lib	procedures to solve polynomial systems
triang.lib	procedures for decomposing zero-dimensional ideals
ntsolve.lib	procedures for real Newton solving

Visualization

graphics.lib	procedures to draw with Mathematica
latex.lib	procedures for typesetting in TeX
surf.lib	interface to the surf programme

Links. SINGULAR furthermore features links as general endpoints of communications, i.e. as something, SINGULAR can read from or write to. To this point, the following link types are implemented:

Ascii text	Output can conveniently be viewed and manipulated. Read/write is not the fastest.
DBM	Provides access to data stored in a data base.
MP file	Stores data in the binary Multi Protocol (MP) format [BGS96]. Read/write is very fast.
MP TCP	Exchanges data in binary MP format between processes (on the same or different computers); data exchange is very efficient

The functionality of theses links is provided to the user by a general, consistent and convenient link interface.

Based on MP TCP links, SINGULAR can very efficiently communicate with itself which opens the door for implementations of parallel/distributed algorithms. Furthermore, the same links can be used to communicate with other Computer Algebra programs which have an MP interface. At the moment, there are MP interfaces for MuPAD and Mathematica, enabling the use of one system from the others (e.g. one can use SINGULAR's functionality from within Mathematica or MuPAD).

Performance. SINGULAR's kernel is implemented in C/C++. SINGULAR's main implementation design goal is speed of the mainly used algorithms. Therefore, all time-consuming operations like standard bases computations or factorization are implemented in its kernel. As another consequence, SINGULAR has the concept of a global ring which needs to be defined prior to any polynomial operations. Arbitrary precision integer and floating point arithmetic is accomplished by linking SINGULAR with the GNU multiple precision library `gmp` and modulo arithmetic is accomplished by using look-up tables. Polynomials are internally represented as linked lists of monomials, where a monomial consists of a coefficient and an exponent vector.

To illustrate SINGULAR's computing speed, the table below shows timings of various systems for solving problem 6 of the ISSAC'97 system challenge (computation of a lexicographical Gröbner basis, 16 roots). The first row shows the timings for the Gröbner basis computation (using the default command as the respective manual describes them). The second row shows the time for numerical solving (finding all complex roots) of the same system (commands `NSolve` (Mathematica), `evalf(solve(..))` (Maple) and `triangL_solve` (Singular)). (The new symbolic-numerical algorithms for polynomial system solving in SINGULAR seem to be especially promising, cf. [G00-2]). All timings were taken on a Pentium Pro 200 with 128 MB of RAM running Linux:

	Mathematica 4.0	Maple V.5.1	GB v3	SINGULAR
Gröbner basis	159.49 sec	14.66 sec	207 sec	0.61 sec
numerical solution	15.91 sec	out of mem.	-	4.28 sec

Computations with MuPAD 1.4, Reduce, CoCoA (3.0.2) and Macaulay 2 (0.8.14) could not finish the Gröbner basis computation within 15 CPU hours.

For more examples see: `http://www.symbolicadata.org`, a project benchmarking CA system.

SINGULAR 2.0. The new version, to be released at the end of 2000, has a new internal engine for polynomial arithmetic which dramatically increases its computational efficiency while at the same time decreases its memory usage. ESINGULAR is a new program for out–of–the–box, precustomized Emacs/ XEmacs which runs SINGULAR. Hence, using ESINGULAR, such features as colour input–output highlighting, pull–down menus, line truncation, a demo mode and tab completion are provided. SINGULAR 2.0 has also a new Windows port which includes ESINGULAR.

Future Work. SINGULAR's development is an actively ongoing project. Currently, the following features are under development:

- more flexible ring concepts
- faster computations by better adaption of datastructures to algorithms
- Newton polyhedron algorithms
- integral closure of an ideal
- primary decomposition for modules

Furthermore, an independent version SINGULAR::PLURAL is under development which is able to compute standard bases and syzygies in very general *non–commutative* structures with applications especially for Lie algebras.

References.

[BGS96] O. Bachman, S. Gray, and H. Schönemann. MP: A Frame work for Distributed Polynomial Systems Based on MP. In *Proc. of the International Symposium on Symbolic and Algebraic Computation (ISSAC'96)*, Zurich, Switzerland, July 1996, ACM Press.

[GGM] H. Grassmann, G.-M. Greuel, B. Martin, W. Neumann, G. Pfister, W. Pohl, H. Schönemann, and T. Siebert. On an implementaion of standard bases and syzygies in SINGULAR. *Computational methods in Lie Theory.* AAECC, 7:235-249, 1996.

[GP96] G.-M. Greuel and G. Pfister. Advances and improvements in the theory of standard bases and syzygies. *Arch. d. Math.*, 66:163-176, 1996

[GP98] G.-M. Greuel and G. Pfister. Groebner bases and algebraic geometry. In B. Buchberger and F. Winkler, editors, *Groebner bases and applications*, volume 251 of *London Mathematical Society Lecture Notes*, pages 109-143, Cambridge University Press, 1998

[GPS98] G.-M. Greuel, G. Pfister, and H. Schönemann. Singular version 1.2 User Manual. In *Reports On Computer Algebra*, number 21, Center for Computer Algebra. University of Kaiserslautern. June 1998.

[GR00-1] G.-M. Greuel. Computeralgebra and Algebraic Geometry Achievements and perspectives. To appear in JSC, 2000

[G00-2] G.-M. Greuel. Applications of Computer Algebra to Algebraic Geometry, Singularity Theory and Symbolic-Numerical Solving. To appear in *Proceedings of the 3rd ECM, Barcelona 2000*