

SINGULAR – A computer algebra system for polynomial computations

Christian Eder

University of Kaiserslautern

June 29, 2011

The following section is about

- 1 **What is Singular?**
- 2 How to get Singular?
- 3 Singular community
- 4 Recent developments in Singular

What is SINGULAR?

SINGULAR

- ▶ is a computer algebra system for polynomial computations.

SINGULAR

- ▶ is a computer algebra system for polynomial computations.
- ▶ has a special emphasis on commutative and non-commutative algebra, algebraic geometry, and singularity theory.

SINGULAR

- ▶ is a computer algebra system for polynomial computations.
- ▶ has a special emphasis on commutative and non-commutative algebra, algebraic geometry, and singularity theory.
- ▶ is free and open-source under the GPL.

What is SINGULAR?

SINGULAR provides

- ▶ highly efficient core algorithms,

What is SINGULAR?

SINGULAR provides

- ▶ highly efficient core algorithms,
- ▶ an intuitive, C-like high-level programming language,

What is SINGULAR?

SINGULAR provides

- ▶ highly efficient core algorithms,
- ▶ an intuitive, C-like high-level programming language,
- ▶ easy ways to make it user-extendible through libraries,

What is SINGULAR?

SINGULAR provides

- ▶ highly efficient core algorithms,
- ▶ an intuitive, C-like high-level programming language,
- ▶ easy ways to make it user-extendible through libraries,
- ▶ a comprehensive online manual and help function.

What is SINGULAR?

SINGULAR's core algorithms handle

- ▶ Gröbner resp. standard bases and free resolutions,

SINGULAR's core algorithms handle

- ▶ Gröbner resp. standard bases and free resolutions,
- ▶ polynomial factorization,

SINGULAR's core algorithms handle

- ▶ Gröbner resp. standard bases and free resolutions,
- ▶ polynomial factorization,
- ▶ resultants, characteristic sets, and numerical root finding.

SINGULAR's core algorithms handle

- ▶ Gröbner resp. standard bases and free resolutions,
- ▶ polynomial factorization,
- ▶ resultants, characteristic sets, and numerical root finding.

SINGULAR's advanced algorithms are contained in **more than 90 libraries**, written in a high-level programming language, handle

What is SINGULAR?

SINGULAR's core algorithms handle

- ▶ Gröbner resp. standard bases and free resolutions,
- ▶ polynomial factorization,
- ▶ resultants, characteristic sets, and numerical root finding.

SINGULAR's advanced algorithms are contained in **more than 90 libraries**, written in a high-level programming language, handle

- ▶ primary decomposition,

SINGULAR's core algorithms handle

- ▶ Gröbner resp. standard bases and free resolutions,
- ▶ polynomial factorization,
- ▶ resultants, characteristic sets, and numerical root finding.

SINGULAR's advanced algorithms are contained in **more than 90 libraries**, written in a high-level programming language, handle

- ▶ primary decomposition,
- ▶ resolution of singularities,

SINGULAR's core algorithms handle

- ▶ Gröbner resp. standard bases and free resolutions,
- ▶ polynomial factorization,
- ▶ resultants, characteristic sets, and numerical root finding.

SINGULAR's advanced algorithms are contained in **more than 90 libraries**, written in a high-level programming language, handle

- ▶ primary decomposition,
- ▶ resolution of singularities,
- ▶ normalization,

SINGULAR's core algorithms handle

- ▶ Gröbner resp. standard bases and free resolutions,
- ▶ polynomial factorization,
- ▶ resultants, characteristic sets, and numerical root finding.

SINGULAR's advanced algorithms are contained in **more than 90 libraries**, written in a high-level programming language, handle

- ▶ primary decomposition,
- ▶ resolution of singularities,
- ▶ normalization,
- ▶ absolute factorization, classification of singularities, deformation theory, Gauss-Manins systems, Hamburg-Noether (Puiseux) development, invariant theory, (non-)commutative homological algebra, and many more.

The following section is about

- ① What is Singular?
- ② **How to get Singular?**
- ③ Singular community
- ④ Recent developments in Singular

How to get SINGULAR?

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

- ▶ binaries for Linux, Mac, and Windows (via cygwin)
- ▶ sources to compile
- ▶ nightly builds of developer version

How to get SINGULAR?

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

- ▶ binaries for Linux, Mac, and Windows (via cygwin)
- ▶ sources to compile
- ▶ nightly builds of developer version

Sage uses SINGULAR for its polynomial arithmetic and Gröbner bases computations.

The following section is about

- ① What is Singular?
- ② How to get Singular?
- ③ Singular community**
- ④ Recent developments in Singular

- ▶ `http://www.singular.uni-kl.de/forum`
Great community – newbies up to experts

- ▶ <http://www.singular.uni-kl.de/forum>
Great community – newbies up to experts

- ▶ <http://www.singular.uni-kl.de:8002/trac>
Give notice of bugs, propose new feature

- ▶ <http://www.singular.uni-kl.de/forum>
Great community – newbies up to experts
- ▶ <http://www.singular.uni-kl.de:8002/trac>
Give notice of bugs, propose new feature
- ▶ <http://www.singular.uni-kl.de/.../open-tasks.html>
Direct feedback of open tasks and ongoing developments

The following section is about

- ① What is Singular?
- ② How to get Singular?
- ③ Singular community
- ④ **Recent developments in Singular**

- ▶ Facilitate interaction with other software packages
 - ▷ `libsingular`
 - ▷ communication interfaces (tropical geometry, toric geometry, group theory, etc.)
 - ▷ links, e.g. SCSCP from SCIENCE project

- ▶ Facilitate interaction with other software packages
 - ▷ `libsingular`
 - ▷ communication interfaces (tropical geometry, toric geometry, group theory, etc.)
 - ▷ links, e.g. SCSCP from SCIENCE project

- ▶ More flexibility/functionality for library programmers
 - ▷ user defined data types
 - ▷ Python interface
 - ▷ GAP interface (Groups, Algorithms, and Programming)

- ▶ Complete restructure of the SINGULAR kernel
 - ▷ easier build process
 - ▷ better documentation
 - ▷ more intuitive

- ▶ Complete restructure of the SINGULAR kernel
 - ▷ easier build process
 - ▷ better documentation
 - ▷ more intuitive

- ▶ Library functionality for different levels
 - ▷ libpoly
 - ▷ libgroebner
 - ▷ etc.

There are two different ways to achieve parallelization of your software

- ▶ different processes, and
- ▶ different threads.

There are two different ways to achieve parallelization of your software

- ▶ different processes, and
- ▶ different threads.

Currently SINGULAR can only use different processes.

- ▶ communication slower than in the threaded world \Rightarrow improved and extended link interface
- ▶ no data corruption possible

There are two different ways to achieve parallelization of your software

- ▶ different processes, and
- ▶ different threads.

Currently `SINGULAR` can only use different processes.

- ▶ communication slower than in the threaded world \Rightarrow improved and extended link interface
- ▶ no data corruption possible

Useful situations:

- ▶ big tasks
- ▶ modular computations, e.g. `modstd`, `primDecZ`, and modular primary decomposition

Implementation of `parallel.lib` for different processes

- ▶ basic framework for modular computations
- ▶ networks
- ▶ worker farms, etc.

Implementation of `parallel.lib` for different processes

- ▶ basic framework for modular computations
- ▶ networks
- ▶ worker farms, etc.

What's about thread-safeness?

- ▶ Redesign of Singular's highly-optimized special-purpose memory manager
- ▶ Combining multiple threads and processes
- ▶ Dynamic action handler

Future Gröbner basis algorithms

- ▶ signature-based algorithms, e.g. F5, G2V
- ▶ F4-like Gaussian Elimination for reduction steps
- ▶ recent research
- ▶ parallelized, more dynamic

Thank you for your attention!