New Developments in SINGULAR

J. Böhm, W. Decker, H. Schönemann, M. Schulze

boehm, decker, hannes, mschulze @mathematik.uni-kl.de

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

Technische Universität Kaiserslautern

Algorithmic and Experimental Methods in Algebra, Geometry, and Number Theory
DFG Priority Project SPP 1489

What is SINGULAR?

- One of the leading computer algebra systems for polynomials computations, over 30 development teams worldwide, over 130 libraries for advanced topics.
- Special emphasis on algebraic geometry, commutative and non-commutative algebra, and singularity theory. Packages for convex and tropical geometry.
- Polynomial rings over $\mathbb{Q}$, $\mathbb{F}_{p^r}$, algebraic and transcendental extensions, $\mathbb{Z}$, $\mathbb{Z}/n$.
- Exterior and Weyl algebras.
- Interpreter language for easy access.

Gröbner Bases

- Highly efficient implementations of Buchberger's and Mora's algorithm, factorizing and Hilbert-driven Buchberger, FGLM, F4.
- Schreyer's algorithm for free resolutions.
- Sheaf cohomology: `sheafCoh.lib`.

Primary Decomposition

- Highly efficient algorithms for multivariate polynomial factorization: FACTORY.
- Primary decomposition algorithms of Gianni-Trager-Zacharias, Shimoyama-Yokoyama, Eisenbud-Huneke-Vasconcelos: `primdec.lib`.

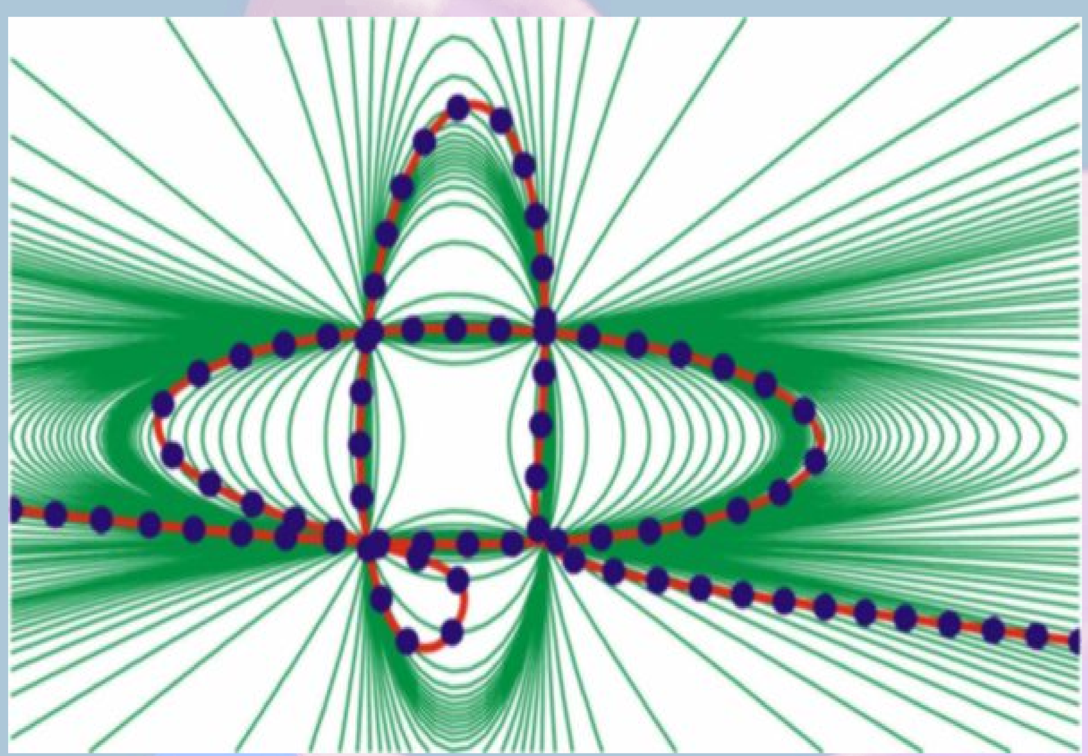
Singularities

- Localizations, Milnor and Tjurina numbers.
- Monodromy, spectral pairs: `gmssing.lib`.
- Classification of complex/real singularities: `classify.lib`, `realclassify.lib`.
- Resolution of singularities: `resolve.lib`

New Algorithms in Singular

- Improved algorithm for free resolutions: `schreyer.lib`.
- Framework for hyperplane arrangements: `arr.lib` (Möbius-function and Poincaré-polynomial, Orlik-Solomon algebra, logarithmic derivations and freeness, multi-arrangements).
- F4/signature based Gröbner basis algorithms: `MathicGB`, `sba`.
- Fast Gröbner bases over algebraic number fields: `nfmodstd.lib`.
- Riemann-Roch spaces via the algorithms of Hess and Brill-Noether: `hess.lib`, `brillnoether.lib`.
- Localizations at arbitrary prime ideals: `graal.lib` (filtration, associated graded ring, initial ideals, free resolutions, test for regularity).
- Local-to-global and modular algorithms for normalization: `locnormal.lib`. Integral bases via Hensel lifting.
- Intersection theory: `schubert.lib`.
- GIT-fan in geometric invariant theory: `gitfan.lib`.
- New algorithms for computing tropical varieties: `gfanlib.so`.

Adjoint Ideals and Parametrization of Rational Curves



- The Gorenstein adjoint ideal $\mathfrak{G}$ of a projective curve C is the conductor $\mathcal{C}_{\mathcal{O}_{C,P}}$ at every $P \in C$.
- Elements pass through the singularities of C with "sufficiently high" multiplicity.
- Parallel local-to-global algorithm computing $\mathfrak{G}$ as intersection of local adjoint ideals.
- For C over $\mathbb{Q}$, parallel modular approach with efficient verification test.
- Implementation in `adjointideal.lib`.
- Applications to computation of Riemann-Roch spaces (`brillnoether.lib`) and rational parametrizations (`paraplanecurves.lib`).

De Rham Cohomology

- General implementation of the algorithm of Oaku/Takayama/Walther for computing the de Rham cohomology of the complement U of a complex affine variety: `deRham.lib`.
- De Rham cohomology of U is the hypercohomology of the de Rham complex on U, i.e. the complex of algebraic differential forms on U.
- Grothendieck and Deligne showed that it agrees with singular cohomology, hence, can be used to compute Betti numbers.
- Ongoing: Compute Gauss-Manin systems (direct images of D-modules), local monodromy.

System Development

- User-defined data types in the interpreter: `newstruct`.
- Multi-methods via command `branchTo`.
- Online version of SINGULAR.
- New routines for screen output, LaTeX and MATHML output (ongoing).
- Modernized JIT interpreter (ongoing).
- Thread-safe kernel for parallelization (ongoing).

JIT Interpreter

Goal:

- Modern SINGULAR interpreter.
- Fully backwards compatible.
- Execution speed similar to C-code.
- Retain existing highly optimized kernel.

Approach:

- Write interpreter in modern language JULIA.
- Run-time conversion of SINGULAR code into JULIA code.
- This code is just-in-time (JIT) compiled.
- JULIA libraries will be available (generic data structures (NEMO), numerics, visualization).

Parallelization

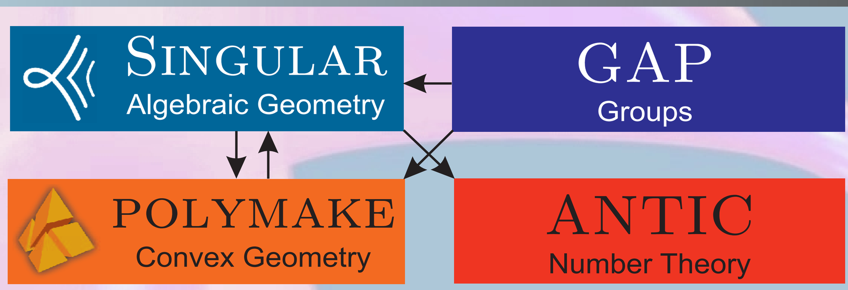
Available:

- Framework for coarse-grained parallelization: `parallel.lib`.
- Modular and local-to-global algorithms, e.g. `modstd.lib`, `nfmodstd.lib`, `locnormal.lib`, `adjointideal.lib`, `integralbasis.lib`.

Ongoing work on:

- Thread-safe memory management
- Fine-grained parallelization
- Systematic design of new parallel algorithms, e.g. for computing integral bases, GIT-fans, tropical varieties.

Connections to other Systems



Included subsystems:

- FACTORY: Polynomial Factorization
- FLINT: Arithmetic for Number Theory
- MATHICGB: F4 Gröbner Basis Algorithm

Further interfaces to:

- GFAN: Tropical Geometry
- POLYBORI: Polynomials over Boolean Rings
- NORMALIZ: Affine Semigroups

References

[1] J. Böhm, W. Decker, S. Laplagne, G. Pfister, A. Steenpaß, S. Steidel: *Parallel Algorithms for Normalization.* J. Symbolic Comput. 51 (2013).

[2] J. Böhm, W. Decker, C. Fieker, G. Pfister: *The Use of Bad Primes in Rational Reconstruction.* Math. Comp. (2015).

[3] J. Böhm, W. Decker, S. Laplagne, G. Pfister: *Local to global algorithms for the Gorenstein adjoint ideal of a curve.* arXiv: 1505.05040. (2015).

[4] D. Boku, C. Fieker, W. Decker, A. Steenpaß: *Gröbner Bases over Algebraic Number Fields.* Proc. PASCO 2015. arXiv: 1504.04564 (2015).

[5] B. Erocal, O. Motsak, F.-O. Schreyer, A. Steenpaß: *Refined Algorithms to Compute Syzygies*, arXiv: 1502.01654 (2015).