SINGULAR 3-1-3 – Tackling New Challenges

Anne Frühbis-Krüger, Andreas Steenpaß, Stefan Steidel

Leibniz Universität Hannover,
Technische Universität Kaiserslautern

June 01, 2011

New Tasks

Driving force behind SINGULAR's development:

mathematical problems

New Tasks

Driving force behind SINGULAR's development:
mathematical problems

Most recent challenges: nationwide DFG-Priority Program

'Algorithmic and Experimental Methods in Algebra, Geometry and Group Theory'

New Tasks

Driving force behind SINGULAR's development:
mathematical problems

Most recent challenges: nationwide DFG-Priority Program

'Algorithmic and Experimental Methods in Algebra, Geometry and Group Theory'

SINGULAR connects to projects in different directions:

- tropical geometry
- toric geometry
- monodromy of families of varieties

New Tasks

Driving force behind SINGULAR's development:
mathematical problems

Most recent challenges: nationwide DFG-Priority Program

'Algorithmic and Experimental Methods in Algebra, Geometry and Group Theory'

SINGULAR connects to projects in different directions:

- tropical geometry
- toric geometry
- monodromy of families of varieties
- Deligne-Lusztig varieties
- regular models for arithmetic surfaces

New Tasks

Driving force behind SINGULAR's development:
mathematical problems

Most recent challenges: nationwide DFG-Priority Program

'Algorithmic and Experimental Methods in Algebra, Geometry and Group Theory'

SINGULAR connects to projects in different directions:

- tropical geometry
- toric geometry
- monodromy of families of varieties
- Deligne-Lusztig varieties
- regular models for arithmetic surfaces
- cryptographical applications

New Tasks

Driving force behind SINGULAR's development:
  mathematical problems

Most recent challenges: nationwide DFG-Priority Program

'Algorithmic and Experimental Methods in Algebra, Geometry and Group Theory'

SINGULAR connects to projects in different directions:

- tropical geometry
- toric geometry
- monodromy of families of varieties
- Deligne-Lusztig varieties
- regular models for arithmetic surfaces
- cryptographical applications
- commutative and non-comm. Gröbner Bases in group theory

New Challenges - Technical Side

- Facilitate interaction with other software packages
  ($\Longrightarrow$ e.g. libsingular,
  communication interfaces to other systems)

New Challenges - Technical Side

- Facilitate interaction with other software packages
  ($\Longrightarrow$ e.g. libsingular,
  communication interfaces to other systems)
- Allow more flexibility/functionality for library programmers
  ($\Longrightarrow$ e.g. user defined structured data types,
  python_objects, python_run)

New Challenges - Technical Side

- Facilitate interaction with other software packages
  ($\Longrightarrow$ e.g. libsingular,
  communication interfaces to other systems)
- Allow more flexibility/functionality for library programmers
  ($\Longrightarrow$ e.g. user defined structured data types,
  python_objects, python_run)
- Make use of multicore processors
  ($\Longrightarrow$ parallelization)

Meeting an explicit algorithmic challenge: Gröbner Bases over Rings

- Verification of integrated digital circuits
  (electrical engineering, e.g. IC for multiplication)

Meeting an explicit algorithmic challenge: Gröbner Bases over Rings

- Verification of integrated digital circuits
  (electrical engineering, e.g. IC for multiplication)
  mathematical task: Gröbner Bases in $(\mathbb{Z}/2^N\mathbb{Z})[\underline{x}]$
  where $N = 8, 16, 32$

Meeting an explicit algorithmic challenge: Gröbner Bases over Rings

- Verification of integrated digital circuits
  (electrical engineering, e.g. IC for multiplication)
  mathematical task: Gröbner Bases in $(\mathbb{Z}/2^N\mathbb{Z})[\underline{x}]$
  where $N = 8, 16, 32$
- Regular models for arithmetic surfaces

Meeting an explicit algorithmic challenge: Gröbner Bases over Rings

- Verification of integrated digital circuits
  (electrical engineering, e.g. IC for multiplication)
  mathematical task: Gröbner Bases in $(\mathbb{Z}/2^N\mathbb{Z})[\underline{x}]$
  where $N = 8, 16, 32$
- Regular models for arithmetic surfaces
  needs wider functionality over $\mathbb{Z}$,
  not just Gröbner Bases

Regular models via Lipman's algorithm

Theorem (Lipman)

Let X be an excellent, noetherian, reduced scheme of dimension 2. Then X has a desingularization of the form

$$X_r \xrightarrow{\pi_r \circ n_r} \dots \xrightarrow{\pi_2 \circ n_2} X_1 \xrightarrow{\pi_1 \circ n_1} X_0 = X$$

where each n_i is a normalization, π_i a blow-up at a 0-dim. center.

Regular models via Lipman's algorithm

Theorem (Lipman)

Let X be an excellent, noetherian, reduced scheme of dimension 2. Then X has a desingularization of the form

$$X_r \xrightarrow{\pi_r \circ n_r} \dots \xrightarrow{\pi_2 \circ n_2} X_1 \xrightarrow{\pi_1 \circ n_1} X_0 = X$$

where each n_i is a normalization, π_i a blow-up at a 0-dim. center.

apply this over $\mathbb{Z}$, hence algorithmic tasks:

- normalization
- blow-up (= appropriate Gröbner basis computation)
- primary decomposition (as tool during normalization steps)

Normalization

Theorem (Grauert-Remmert Criterion for Normality)

R noetherian, reduced ring, $J \subset R$ ideal s.th.

1. *J contains a non-zerodivisor p on A,*
2. *J is a radical ideal,*
3. *$V(\mathcal{C}_{\overline{R}|R}) \subset V(J)$.*

Then $R = \overline{R} \Longleftrightarrow R \cong \mathrm{Hom}_R(J, J) \cong \frac{1}{p}(pJ :_R J) \subset \overline{R} \subset Q(R)$.

Normalization

Theorem (Grauert-Remmert Criterion for Normality)

R noetherian, reduced ring, $J \subset R$ ideal s.th.

1. *J contains a non-zerodivisor p on A,*
2. *J is a radical ideal,*
3. *$V(\mathcal{C}_{\overline{R}|R}) \subset V(J)$.*

Then $R = \overline{R} \Longleftrightarrow R \cong \mathrm{Hom}_R(J, J) \cong \frac{1}{p}(pJ :_R J) \subset \overline{R} \subset Q(R)$.

Resulting algorithm <u>requires</u> primary decomposition
(here over $\mathbb{Z}$ to get $\overline{\mathbb{Z}[\underline{x}]/I}$):

- for equidimensional decomposition,
- for decomposing the singular locus.

Primary Decomposition over $\mathbb{Z}$

Well-known: primary decomposition in $\mathbb{Q}[\underline{x}]$ and $\mathbb{Z}/p[\underline{x}]$
(e.g. Gianni-Trager-Zacharias)

New aspects over $\mathbb{Z}$:

- $I \cap \mathbb{Z} = \langle m \rangle$, $m \in \mathbb{Z} \setminus \{0\}$, may occur
  for original ideal or at intermediate step

Primary Decomposition over $\mathbb{Z}$

Well-known: primary decomposition in $\mathbb{Q}[\underline{x}]$ and $\mathbb{Z}/p[\underline{x}]$
(e.g. Gianni-Trager-Zacharias)

New aspects over $\mathbb{Z}$:

- $I \cap \mathbb{Z} = \langle m \rangle$, $m \in \mathbb{Z} \setminus \{0\}$, may occur
  for original ideal or at intermediate step
- Combine
  - primary components over $\mathbb{Q}$ (in absence of such m)
  - minimal associated primes over $\mathbb{Z}/p$ for prime factors p of m
  - extraction of primary components over $\mathbb{Z}$

Primary Decomposition over $\mathbb{Z}$

Well-known: primary decomposition in $\mathbb{Q}[\underline{x}]$ and $\mathbb{Z}/p[\underline{x}]$
(e.g. Gianni-Trager-Zacharias)

New aspects over $\mathbb{Z}$:

- $I \cap \mathbb{Z} = \langle m \rangle$, $m \in \mathbb{Z} \setminus \{0\}$, may occur
  for original ideal or at intermediate step
- Combine
  - primary components over $\mathbb{Q}$ (in absence of such m)
  - minimal associated primes over $\mathbb{Z}/p$ for prime factors p of m
  - extraction of primary components over $\mathbb{Z}$
- second step above is parallel in nature

Parallel Computing with SINGULAR

On two different levels:

1. SINGULAR-SINGULAR communication
   - for larger subtasks (communication overhead)
   - improved and extended communication interface
   - examples: modStd, primdecZ, modular primary decomposition

Parallel Computing with SINGULAR

On two different levels:

1. SINGULAR-SINGULAR communication
   - for larger subtasks (communication overhead)
   - improved and extended communication interface
   - examples: modStd, primdecZ, modular primary decomposition
2. Threads
   - for fine grained subtasks
   - work in progress, long term goal, affects memory management
   - examples: polynomial arithmetic, polynomial gcd

Application: Modular Standard Bases

Modular computation of standard basis G of an ideal $I \subset \mathbb{Q}[\underline{x}]$:

1. choose several primes and compute SB G_p modulo each p
2. delete unlucky primes
   (i.e. $L(I_p) \neq L(I_q)$ for most primes $q \neq p$)
3. find G by Chinese remaindering and Farey rational map
4. test Gröbner Basis property (most expensive part)
5. reiterate with further (new) primes if test fails

Modular Standard Bases: Gröbner Basis test

SINGULAR's method for final verification:

Theorem (Arnold (homogeneous)/ Idrees, Pfister, Steidel (general for global & local orderings))

Let $I \subseteq \mathbb{Q}[\underline{x}]$ an ideal and $G \subseteq I$ a set of polynomials such that

- $L(G) = L(G_p)$ *where* G_p *is a standard basis of* I_p *for some prime number* p,
- G *is a standard basis of* $\langle G \rangle$,
- $I \subseteq \langle G \rangle$.

Then $I = \langle G \rangle$.

Modular Standard Bases in Parallel

Parallel steps in `modStd`:

1. Compute G_p for different p in parallel
   via SINGULAR-SINGULAR communication

2. Parallel final verification tests:
   - test $f \in \langle G \rangle$ for each generator f of I
   - reduce s-polynomials of G w.r.t. G

   will be implemented via threads

Modular Standard Bases and Parallel Computing

Example	`std`	`modStd`	$\texttt{modStd}_4^*$	$\texttt{modStd}_9^*$
cyclic8	-	8271	4120	2927
Paris.ilias13	37734	1159	676	580
homog.cyclic7	3343	3436	886	408
	-	6	3	3

Table: Total running times (in sec) for computing a standard basis of examples chosen from The SymbolicData Project (H.-G. Gräbe) via `std`, `modStd` and its parallelized variant $\texttt{modStd}_n^*$ for $n = 4, 9$.

Modular Primary Decomposition in Dimension Zero

"Decompose" singular locus of randomly chosen rational plane curves:

degree	`minassGTZ`	`assPrimes`
7	39	166
8	193	196
9	3776	344
10	24865	969
11	270886	2535
12	943654	6369

Table: Total running times (in ms), where assPrimes uses 8 cores.

Normalization via Localization

Proposition

R noetherian domain, $\mathrm{Sing}(R) = \{P_1, \ldots, P_s\}$, $S_i = R \setminus P_i$ for $i = 1, \ldots, s$. Suppose intermediate rings $R \subset R^{(i)} \subset \overline{R}$ are given such that $S_i^{-1} R^{(i)} = \overline{S_i^{-1} R}$. Then $\sum_{i=1}^{s} R^{(i)} = \overline{R}$.

Normalization via Localization

Proposition

R noetherian domain, $\operatorname{Sing}(R) = \{P_1, \ldots, P_s\}$, $S_i = R \setminus P_i$ for $i = 1, \ldots, s$. Suppose intermediate rings $R \subset R^{(i)} \subset \overline{R}$ are given such that $S_i^{-1} R^{(i)} = \overline{S_i^{-1} R}$. Then $\sum_{i=1}^{s} R^{(i)} = \overline{R}$.

Proposition (Local Normality Criterion)

R noetherian domain, $R \subset R'$ module-finite extension, $P \in \operatorname{Sing}(R)$ minimal with respect to inclusion, $S = R \setminus P$, $J' = \sqrt{PR'}$. If $R' \cong \operatorname{Hom}_{R'}(J', J')$, then $S^{-1}R'$ is normal.

Integral Bases, Adjoint Ideals, Parametrization of Rational Curves

R coordinate ring of irreducible plane curve. Then:

- Sing(R) zero-dimensional $\Rightarrow$ can use modular primary decomposition
- local contributions to normalization also via Puiseux expansions (van Hoeij) and Hensel-lifting (Böhm-Decker-Laplagne-Seelisch)

Important applications: Adjoint ideals, parametrization of rational curves. New, fast algorithms by Böhm-Decker-Laplagne-Seelisch. Two different approaches to compute adjoint ideals via localization.

SINGULAR Developers

Recent Developments since Singular 3-1-2

New contributed/experimental libraries include:

- `integralbasis.lib` – Integral Bases
- `paraplanecurve.lib` – Rational Parametrization of Rational Plane Curves
- `monomialideal.lib` – Operations for Monomial Ideals
- `multigrading.lib` – Multigradings and Related Operations
- `primdecint.lib` – Primary Decomposition over the Integers
- `resbinomial.lib` – Desingularization of Binomial Ideals
- and many more

New/improved features include:

- user defined types
- python objects
- new Singular-Singular communication interfaces