SINGULAR: Using and Programming

Viktor Levandovskyy

RWTH Aachen, Germany

15.11.2007, RWTH

Where to find the information about SINGULAR?

On the SINGULAR homepage

- `http://www.singular.uni-kl.de/` (Download, Online documentation, SINGULAR Discussion Forum etc.)

In the book

- ”A Singular Introduction to Commutative Algebra”

  by G.-M. Greuel and G. Pfister, Springer 2002
- The second edition appeared in 2007 (ask me about it).

SINGULAR is a free service to the mathematical community, it is distributed under GPL license.

Background

The development of SINGULAR started in early 80's in order to support the research in

- commutative algebra
- algebraic geometry
- singularity theory

as well as aiming at real life applications of these disciplines. Meanwhile, the area of applications of SINGULAR grew significantly and includes coding theory, cryptoanalysis, development of electric circuits, control theory and more.

• SINGULAR is one of the fastest computer algebra systems in the area of polynomial computations and Gröbner bases.

• In 2004, SINGULAR was awarded with the *Richard D. Jenks Memorial Prize for Excellence in Software Engineering Applied to Computer Algebra*.

Introduction to Data Types

”Lord Of The Rings” Principle

Almost all computations in SINGULAR are done inside of some **ring**, which must be defined explicitly.

Example (There are data types, not depending on a ring)

- `int, intvec, intmat`: integer number, vector and matrix
- `bigint`
- `string`
- `list` a collection of any data
- `def` special universal data type
- `link` communication link (e.g ASCII, DBM)
- `package` groups identifiers into collections

Ring-dependent Data Types

The ring data types are `ring` and `qring`.

Example (Data types, depending on a ring)

- `number` element from the ground field
- `poly, vector, matrix`
- `ideal, module`
- `resolution`
- `map`

Introduction to Programming

”Hello World”;

”Eternal Ending Semicolon” Principle

End every self–contained input string with the ”;”.
Exceptions: control structures `if, then, else, for, while, {, }` etc.

Example (Control Structures)

- `if (CONDITION) then { DO THIS; } else { DO THAT;}` : if–then–else
- `for (CONDITIONS) { DO THIS; }` : classical cycle
- `break;` escapes from the innermost `for` or `while` cycle
- `while (CONDITION) { DO THIS; }`: while cycle
- `return` returns value (from the procedure)

Rings::Generalities

"Lord Of The Rings" Principle

Almost all computations in SINGULAR are done inside of some ring, which has to be defined explicitly.

Assumption

A ring R contains the identity 1 and is finitely generated.

For constructing a ring, we need

- a field $\mathbb{K}$ (together with *parameters* $p_1, \ldots, p_m$)
- a set of variables, e.g. `x,Y1,Psmall,Dt,u',XA_3`
- a monomial (module) ordering $\prec$ on the variables

Rings::Possibilities

In SINGULAR, one can set up the following commutative rings

(P) a polynomial ring $\mathbb{K}[x_1, \ldots, x_n]$ over a field $\mathbb{K}$

(S) a localization of a polynomial ring, e.g. $\mathbb{K}[x_1, \ldots, x_n]_{\langle x_1, \ldots, x_n \rangle}$

- a factor ring (also called quotient ring) by an ideal P/I or S/J
- a tensor product over a field $P/I \otimes_{\mathbb{K}} S/J$

The non-commutative subsystem PLURAL provides a possibility to set up and to work with non–commutative polynomial algebras (*GR*–algebras a.k.a. PBW algebras), where the variables $x_1, \ldots, x_n$ obey the relations

$$x_j x_i = c_{ij} x_i x_j + d_{ij} \quad \forall 1 \leq i < j \leq n, \qquad c_{ij} \in \mathbb{K}^*$$

with some more technical conditions.

Rings::Fields

Finite Fields

- $\mathbb{Z}/\mathbb{Z}p,\ p \leq 2147483629$, p a prime
- Galois fields $GF(p^n)$ with $p^n \leq 2^{15}$ elements

Extensions

- transcendental field extension by *parameters* $\mathbb{K}(p_1, \ldots, p_m)$
- simple algebraic extension with a parameter and it minimal polynomial $\mu(a)$ produces $\mathbb{K}[a]/\mu(a)$
- multiparametric algebraic extensions may be converted to a simple algebraic extension by using a library PRIMITIV.LIB

Numerical fields

- `(real,10,20)` for $\mathbb{R}$: 10 valid digits, 20 digits for the rest
- `(complex,30,50)` for $\mathbb{C}$, where $\sqrt{-1} =: i$

Rings::Fields Examples

Finite Fields

- `ring r1 = 11111,(x),dp;` gives $(\mathbb{Z}/11093\mathbb{Z})[x]$
  $11111 = 41 \cdot 271$
- `ring G = (1024,g),(x,y),dp;` gives $GF(2^{10})[x, y]$, where g is a generator of the cyclic group of units of $GF(2^{10})$

Extensions

- $\mathbb{Z}/7\mathbb{Z}(a, b, c)[X_1, X_2]$: `ring t = (7,a,b,c),(X1,X2),dp;`
- $(\mathbb{Q}[i]/(i^2 + 1))[y]$:

```
ring A = (0,i),(y),dp;
minpoly = i^2+1;
```

Remark

Arbitrarily long integers are handled with the data type `bigint`.

Rings::Fields Examples

Cf. the file `ex-live.tst`.

Multiparametric extensions

Suppose we want to introduce an extension of $\mathbb{Q}$ with x, y s.t. $x^2 = -1, y^3 = 3$. We can present it as a simple algebraic extension.

```
LIB "primitiv.lib";
ring ex = 0,(x,y),dp;
ideal I = x2+1,y3-3;
primitive_extra(I);
```

The result is

$\mathbb{K}[a]/(a^6 + 3a^4 - 6a^3 + 3a^2 + 18a + 10)$, where
$x = -\frac{1}{179}(24a^5 - 27a^4 + 80a^3 - 234a^2 + 201a + 273)$.

Important

The factorization over most of fields is provided with `factorize`.

Rings::Orderings

There is the following classification:

Definition (Monomial Ordering)

Let $\mathrm{Mon}(R) = \{x^\alpha \mid \alpha \in \mathbb{N}^n\}$.

- $\prec$ is a global ordering, if $1 \prec x^\alpha \; \forall \; \alpha \neq 0$ (polynomials)
- $\prec$ is a local ordering, if $x^\alpha \prec 1 \; \forall \; \alpha \neq 0$ (series)
- otherwise, $\prec$ is a mixed (product) ordering

Robbiano's Construction

Indeed, any monomial ordering can be represented by a matrix $M \in \mathbb{GL}(n, \mathbb{Z})$ by

$$\alpha \prec_M \beta \;\Leftrightarrow\; M\alpha \prec_{lex} M\beta$$

Rings::Orderings Examples

Global and Product Monomial Orderings

- `lp` lexicographical ordering
- `dp` degree reverse lexicographical ordering
- `wp(`w_1`,...,`w_n`)` w–weighted degrevlex ordering
- `Dp` degree lexicographical ordering
- `Wp(`w_1`,...,`w_n`)` w–weighted deglex ordering
- `(ord1,...,ordN)` a product ordering (e.g. `(dp(2),lp(3))`
- `M(`m_{11}`,...,`m_{nn}`)` matrix–defined ordering
- `(a(`w_1`,...,`w_n`),ord)` extra weight ordering

Module Orderings

- Position–over–Term `(c,dp)` resp. Term–over–Position `(dp,C)`
- descending (”C”) resp. ascending (”c”) order of components

Data Types `poly` and `ideal`

Cf. the file `ex-live.tst`.

Polynomials

`poly` corresponds to a finite sum of monomials in variables of the ring with coefficients from the ground field of the ring, where the monomials are ordered, according to the monomial ordering of the ring.

```
ring r = (0,a),(x,y,z),Dp;
poly p = a^7*x^2*y - 343*xz*(y - (az +x)^3);
p; // prints p in the expanded form
factorize(p); // factorization
```

Data Type `ideal`

Constructively it is a list of generators of type `poly`. `ncols` gives the total number of elements, `size` gives the number of nonzero elements. For numerous reasons, we want to compute Gröbner bases of ideals with respect to a fixed monomial ordering.

Engine: Gröbner basis

There are many possibilities to compute GB

- GB of a submodule of a free module of finite rank
- w.r.t. any monomial module ordering
- reduced resp. completely reduced Gröbner basis
- quite fast in general
- `groebner` computes a GB with heuristically chosen method
- classical all–purpose standard basis `std`
- GB and minimal basis together `mstd`
- FGLM method `stdfglm`
- Hilbert–driven method `stdhilb`
- factorizing Groebner basis algorithm `facstd`
- a recent addition: slim Groebner basis `slimgb`

GB, Syzygies and Transformation Matrix

Let A be a GR–algebra, $I = \{\bar{f}_1, \ldots, \bar{f}_k\} \subset A^r$ be a left submodule. We denote $\bar{0} = (0, \ldots, 0) \in A^r$. Then

$$\begin{pmatrix} \bar{f}_1 & \ldots & \bar{f}_k \\ 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \xrightarrow{GB} \left(\begin{array}{ccc|ccc} \bar{0} & \ldots & \bar{0} & \bar{h}_1 & \ldots & \bar{h}_t \\ \hline & & & & & \\ & \mathbf{S} & & & \mathbf{T} & \\ & & & & & \end{array}\right).$$

Let H be a matrix with columns $\bar{h}_i$. Then

- $\{\bar{h}_1, \ldots, \bar{h}_t\}$ is a Gröbner basis of I,
- columns of $\mathbf{S}$ generate $\mathrm{syz}(\{\bar{f}_1, \ldots, \bar{f}_k\})$,
- $\mathbf{T}$ is a left transition matrix between two bases of I, i.e. $H^t = \mathbf{T}^t F^t$,
- note, that if A is commutative, $H = F \cdot \mathbf{T}$.

Gröbner basics

The most important and fundamental applications of GB

- Ideal (resp. module) membership problem: `NF,reduce`
- Elimination of variables: `eliminate`
- Intersection of ideals (resp. submodules): `intersect`
- Quotient and saturation of ideals: `quot`
- Kernel of a module homomorphism: `modulo`
- Kernel of a ring homomorphism: `preimage`
- Algebraic relations between polynomials: ALGEBRA.LIB
- Hilbert polynomial of graded ideals and modules: `hilb`

Various Features

Visualization Tools

- LATEX.LIB converts SINGULAR output of different types (cf. the file `ex-latex.tst`)
- SURF, SURFVIS provide HQ plotting of curves and surfaces (cf. the file `ex-surf.tst`)
- GRAPHICS.LIB allows to use graphics with MATHEMATICA

Intercommunication Tools

- MP (Multi Protocol) provides client and server tools
- OPENMATH connectivity already in development
- ASCII links and DBM databases are supported

Third–party Functionality

- nearly 100 libraries distributed with SINGULAR
- dynamical modules mechanism is available
- namespaces (type `package`)

Selected Strong Points

- PRIMDEC.LIB, MPRIMDEC.LIB

  primary decomposition (including the absolute one)

  radical, minimal associated primes
- SOLVE.LIB, PRESOLVE.LIB, TRIANG.LIB etc.

  various methods for numerical solving 0–dimensional systems of polynomial equations
- RESOLVE.LIB, RESZETA.LIB

  resolution of singularities and its applications
- INTPROG.LIB, TORIC.LIB

  integer programming, toric ideals and Gröbner bases
- CONTROL.LIB

  algebraic analysis tools for System and Control Theory

Setting up G–algebras

After initializing a commutative ring R with the ordering $\prec$, one defines C_{ij} and D_{ij} and finally calls `ncalgebra(C,D);` (version before 3-0-3) or `nc_algebra(C,D);` (versions starting from 3-0-4)

```
ring R = 0,(x,y,z),Dp;
int  N = nvars(R);
matrix C[3][3];
C[1,2] = ...; C[1,3] = ...; C[2,3] = ...;
matrix D[N][N];
D[1,3] = ...;
ncalgebra(C,D);
```

Frequently Happening Errors

- matrix is smaller in size than $n \times n$;
- matrix C contain zeros in its upper part;
- the ordering condition is not satisfied.

Setting up Operator Algebras

Task 1

Consider the difference operator $\Delta : f(x) \mapsto f(x + \triangle x) - f(x)$.

1. Derive the non–commutative relations of Δ with the operator $X : f(x) \mapsto x \cdot f(x)$.
2. Set up the corresponding algebra in SINGULAR.
3. Compute the center of the algebra up to degree 3 or 5. What happens if instead $\mathbb{Q}$ as ground field we take $\mathbb{Z}_3$ or $\mathbb{Z}_5$?

Task 2

Let $\mathbb{K} = \mathbb{Q}(q)$ and the q–difference $\Delta_q(f(x)) = f(qx) - f(x)$.

1. Derive the non–comm relations of Δ_q with the operator X
2. Set up the corresponding algebra in SINGULAR.
3. Compute the center of the algebra up to degree 3 or 5. What happens if instead $\mathbb{Q}(q)$ as ground field we take the algebraic extension of $\mathbb{Q}$ by q, such that $q^3 = 1$ resp. $q^5 = 1$?

Solutions to Tasks

Task 1: $\Delta : f(x) \mapsto f(x + \triangle x) - f(x)$

The relation is $\Delta x = x\Delta + \triangle x\Delta + \triangle x$.

```
LIB "central.lib"; // "center.lib" in older versions
int Char = 0; // int Char = 3;
ring A=(Char,t),(x,D),dp;
matrix M[2][2];  M[1,2] = t*D + t;
ncalgebra(1,M); A;
center(3); // compare for Char =0 and 3
```

Solutions to Tasks

Task 2: $\Delta_q(f(x)) = f(qx) - f(x)$

The relation is $\Delta_q x = qx\Delta_q + (q-1)x$.

```
LIB "central.lib";
ring A=(0,q),(x,D),dp;
// minpoly = rootofUnity(3);
matrix N[2][2]; N[1,2] = q;
matrix M[2][2]; M[1,2] = (q-1)*x;
ncalgebra(N,M); A;
center(3);  // compare for minpoly case
```

Symbolic–Numerical Solving

A system of equations S over the field $\mathbb{K}$ corresponds to the ideal $I = I(S)$. There is a finite number of solutions over $\bar{\mathbb{K}}$ if and only if the *dimension* (Krull dimension) of I is 0.

What does solving mean?

There might be different wishes, like

- compute one, some or all the roots with or without multiplicities numerically with a given precision
- compute a field extension $\mathbb{K} \subseteq L$, such that there are exact symbolic expressions for the roots of S in L

SINGULAR has procedures for both ways of solving. The second way can be done, using the primary decomposition.

Symbolic–Numerical Solving. Numerical Fashion.

Recipe for the numerical solving (cf. the file `ex-live.tst`):

1. Check that a given system is consistent (i.e. it has solutions). Compute a Gröbner basis G of the given ideal I with respect to the fast ordering like `dp`. A system has solutions, if and only if G does not contain a constant.
2. A consistent system has finitely many solutions over $\mathbb{K}$ if and only if $\dim I = \dim G = 0$. Compute $\dim G$.
3. If $\dim G = 0$, compute another Gröbner basis T of I with respect to lexicographical ordering (not easy!)
4. Apply your favourite numerical solver to T. In SINGULAR, many different solvers are implemented.

Primary Decomposition: The Beginning

Definition

Let A be a Noetherian ring, and let $I \subset A$ be an ideal.

1. The set of *associated primes* of I, denoted by $\mathrm{Ass}(I)$, is defined as

$$\mathrm{Ass}(I) = \left\{ P \subset A \,\middle|\, P \text{ prime}, P = I : \langle b \rangle \text{ for some } b \in A \right\}.$$

   Elements of $\mathrm{Ass}(\langle 0 \rangle)$ are also called *associated primes* of A.
2. Let $P, Q \in \mathrm{Ass}(I)$ and $Q \subsetneq P$, then P is called an *embedded prime ideal* of I. We define $\mathrm{Ass}(I, P) := \{Q \mid Q \in \mathrm{Ass}(I), Q \subset P\}$.
3. I is a *primary ideal* if, for any $a, b \in A$, $ab \in I$ and $a \notin I$ imply $b \in \sqrt{I}$. Let P be a prime ideal, then a primary ideal I is called *P–primary* if $P = \sqrt{I}$.
4. A *primary decomposition* $I = Q_1 \cap \cdots \cap Q_s$ with Q_i primary ideals, is called *irredundant* if no Q_i can be omitted in the decomposition and if $\sqrt{Q_i} \neq \sqrt{Q_j}$ for all $i \neq j$.

Primary Decomposition: Main Theorems

Theorem

Let A be a Noetherian ring and $I \subset A$ be an ideal, then there exists an irredundant decomposition $I = Q_1 \cap \cdots \cap Q_r$ of I as intersection of primary ideals $Q_1, \ldots, Q_r$.

Theorem

Let A be a ring and $I \subset A$ be an ideal with irredundant primary decomposition $I = Q_1 \cap \cdots \cap Q_r$. Then $r = \#\,\mathrm{Ass}(I)$,

$$\mathrm{Ass}(I) = \{\sqrt{Q_1}, \ldots, \sqrt{Q_r}\},$$

and if $\{\sqrt{Q_{i_1}}, \ldots, \sqrt{Q_{i_s}}\} = \mathrm{Ass}(I, P)$ for $P \in \mathrm{Ass}(I)$ then $Q_{i_1} \cap \cdots \cap Q_{i_s}$ is independent of the decomposition.

Primary Decomposition: Detailed Example I

Example (Cyclic the 3–th)

The system is given by the following equations:

$$x + y + z,$$
$$xy + yz + xz,$$
$$xyz - 1$$

The system is 0–dimensional and has 6 solutions in $\mathbb{C}$,

$$(1, \frac{1}{2}(-1 \mp \sqrt{-3}), \frac{1}{2}(-1 \pm \sqrt{-3})), (\frac{1}{2}(-1 \mp \sqrt{-3}), 1, \frac{1}{2}(-1 \pm \sqrt{-3})),$$

$$(\frac{1}{2}(-1 \mp \sqrt{-3}), \frac{1}{2}(-1 \pm \sqrt{-3}), 1).$$

Primary Decomposition: Detailed Example II

Cf. the file `ex-primdec.tst`.

Example (Cyclic the 4–th)

The system is given by the following equations:

$$\begin{aligned} &x + y + z + t, \\ &xy + yz + xt + zt, \\ &xyz + xyt + xzt + yzt, \\ &xyzt - 1 \end{aligned}$$

Unlike `cyclic(3)` or `cyclic(5)`, the system is not 0–dimensional. In $\mathbb{Q}(t)[x, y, z]$ we have however for the ideal I of the system

$$\sqrt{I} = \langle x - \frac{1}{t}, y + t, z + \frac{1}{t} \rangle \cap \langle x + \frac{1}{t}, y + t, z - \frac{1}{t} \rangle$$

Primary Decomposition: Recipe

1. Compute the Gröbner basis G of I (fast ordering)
2. Compute the radical R of G with e.g. `radical`
3. Compute the minimal associated primes with e.g. `minAssGTZ` or `minAssChar`
4. Analyze each component separately, eventually passing to an extension field

Alternatively, obtain numerical solutions for 0–dimensional components with e.g. `solve`.
If one wish to compute the multiplicities of solutions, one needs the primary decomposition (`primdecGTZ, primdecSY` etc.) and, in particular, the information on primary (and not only prime) components.
The procedures for the primary decomposition are gathered in the `primdec.lib`.