

# An Overview of CASA – A System for Computational Algebra and Constructive Algebraic Geometry

Trần Quốc-Nam and Franz Winkler \*

Research Institute for Symbolic Computation (RISC–Linz),

Johannes Kepler University

A–4040 Linz, Austria

E-mail: {tqnam, winkler}@risc.uni-linz.ac.at

May 7, 1997

CASA is a special-purpose system for computational algebra and constructive algebraic geometry. The system has been developed since 1990. With its newest version, version 2.3, in May 1997, CASA is the ongoing product of the Computer Algebra group at the Research Institute for Symbolic Computation (RISC–Linz), the University of Linz, Austria, under the direction of the second author. The system is built on the kernel of the widely used computer algebra system Maple.

## 1 Introduction

Constructive methods in algebra and algebraic geometry are gaining more and more importance with the availability of computers and computer algebra softwares. Since its first version (see [1]), CASA has been designed to perform computations and utilize reasoning about algebraic and geometrical objects in the classical affine and projective spaces over algebraically closed fields of characteristic zero. More precisely, the field has to be a computable field in the sense of the underlying computer algebra system Maple, i.e. all the arithmetic operations have to be available in the system. Usually, the field of computation is the rational numbers  $\mathbb{Q}$  or a finite algebraic extension thereof.

The system has been developed since 1990. With its newest version, version 2.3, in May 1997, CASA is the ongoing product of the Computer Algebra group at the Research Institute for Symbolic Computation (RISC–Linz), the University of Linz, Austria. Several people have contributed to the system in one way or another. The names of the main contributors (see [9, 3], [8], [2], [4, 5, 7, 6]) are R. Gebauer, M. Kalkbrener, M. Mňuk, J. R. Sendra, P. Stadelmeyer, Q.-N. Tran (current coordinator), B. Wall, F. Winkler (director).

In the system, an algebraic set – a central notion in algebraic geometry – can be represented in four different ways:

- *Implicit representation:* An algebraic set is the set of common zeros of a system of polynomial equations. To give an algebraic set in implicit form means to give finitely many polynomials.
- *Projected representation:* As a consequence of the primitive element theorem every irreducible  $d$ -dimensional algebraic set in  $n$ -dimensional space is, after a suitable linear transformation of coordinates, birationally projectable onto an irreducible  $d$ -dimensional algebraic set in  $(d + 1)$ -dimensional space, which can be specified by a single polynomial in  $d + 1$  variables. This can be generalized to unmixed-dimensional algebraic sets. An algebraic set in projected form is given by a polynomial and a tuple of rational functions (specifying the birational mapping).
- *Parametric representation:* Some irreducible algebraic sets can be parametrized by rational functions. An algebraic set in parametric form is given by a tuple of rational functions that parametrizes the algebraic set.

---

\*Supported by the Austrian Science Foundation (FWF) project HySaX, Proj. No. P11160-TEC

- *Representation by places*: All algebraic curves can be locally parametrized by a set of power series that are convergent around a point of the curve. An algebraic set is given by places if for each branch passing through a certain point on the algebraic set a tuple of power series that parametrizes the algebraic set around the point is specified.

The system provides a variety of operations on algebraic sets. As the efficiency of these operations is tightly bound to the way algebraic sets are represented, conversion routines have been provided to support various views on one object, to deepen the understanding of its principles, and to speed up algorithms working on algebraic sets.

CASA also works with the polynomial ideals corresponding to these geometric objects. The basic operations available in CASA include:

- ideal theoretic operations  $+$ ,  $*$ ,  $\cap$ ,  $/$ ,
- creating algebraic sets in different representations,
- generating curves of fixed multiplicities at given points,
- intersection, union, and difference of algebraic sets,
- computing tangent cones and tangent spaces,
- computation of the dimension of an algebraic set,
- decomposition into irreducible components,
- transformations of algebraic sets to hypersurfaces,
- computation of the singularities, genus, neighborhood graphs and adjoint curves of an algebraic curve.

Besides these basic operations, the following more advanced operations are available:

- rational parametrization of rational curves,
- implicitization of parametrically given algebraic sets,
- Puiseux series expansions,
- multivariate resultants and Dixon's resultants,
- Gröbner bases of ideals and modules,
- Gröbner walk,
- computation of rational points on conics,
- hybrid methods for finding the roots of an arbitrary system of equations,
- plotting both explicitly and implicitly given curves and surfaces,
- computation of syzygy-bases.

The major goal of CASA was to design a system which provides a comfortable, easy to use, efficient, flexible and mathematically exact working environment for computational algebra and constructive algebraic geometry where all basic theoretical concepts map easily to available data structures. CASA is built on the kernel of Maple and is fully independent of the operating system; hence, it can be used on every hardware where Maple is running.

There are software systems which partially cover some of the fields of CASA, however, to the authors' knowledge, currently no other system provides all the functionalities of CASA.

## 2 A Quick Tour

### 2.1 Getting Started

CASA is distributed as a single Maple archive of compiled routines and is loaded via the `with` function after having the language Maple loaded.

```
> with(casa):
Welcome to CASA 2.3.

Copyright (C) 1993-1997 RISC Linz.
For help type '?casa'.
```

The code is organized into a number of separate modules. The modules and their major parts are loaded into memory using the `readlib` facility when they get used for the first time. This significantly cuts the amount of memory occupied by the code.

### 2.2 Basic Operations

We can define an algebraic set in implicit, parametric or projected representation.

```
> A:= mkImplAlgSet([x*(x^2+y^2-1), x*z, x^2+y^2+z^2-1], [x,y,z]);
A := algebraic_set([x (x^2 + y^2 - 1), x z, x^2 + y^2 + z^2 - 1], [x, y, z])

> A2 := mkParaAlgSet([t^2-1, 1/(t-1), t+1], [t]);
A2 := algebraic_set([t^2 - 1, 1/(t - 1), t + 1], [t])

> A3:=mkProjAlgSet([u^2-v^3+v^2], [u+v, u-v, 1/v], [u,v]);
A3 := algebraic_set([u^2 - v^3 + v^2], [u + v, u - v, 1/v], [u, v])
```

We may perform conversion between representations, for example:

1. Birational mapping: We may compute a projection of A. The result is a set of algebraic sets, each given as a hyper-surface and a birational mapping to 3D-space. The union of these algebraic sets is equal to A, the intersection of the corresponding ideal is equal to the radical of A. In this case we obtain two one-dimensional circles.

```
> Aproj:=convertRep(A,proj);
Aproj := algebraic_set([x^2 + y^2 - 1, [x, y, 0]], [y, x]),
algebraic_set([x^2 + z^2 - 1, [0, -x, z]], [x, z])
```

2. Parametrization:

```
> B:= mkImplAlgSet([x^2-y^3, z+y+x]);
B := algebraic_set([x^2 - y^3, z + y + x], [x, y, z])
```

```
> Bpara:=convertRep(B,para);
```

```
Bpara := algebraic_set([
- (3 t + t^3 + 3 t^2 + 1) / (3 t - 3 t^2 - 1 + t^3), (1 + t^2 + 2 t) / (t^2 - 2 t + 1), 2 (1 + t^2 + 2 t) / (3 t - 3 t^2 - 1 + t^3)], [t])
```

3. Representation by places (Puiseux expansion)

```
> Bplac:=impl2plac(B);
> shAlgSet(Bplac,10);
The algebraic set is known to have the following properties:
It is given by the following places:
```

$$\left[ \left[ \frac{1}{8} T^3 + \frac{3}{32} T^4 + \frac{21}{256} T^5 + \frac{5}{64} T^6 + \frac{1287}{16384} T^7 + \frac{21}{256} T^8 + \frac{46189}{524288} T^9 + \right. \right. \\ \left. O(T^{10}), \frac{1}{4} T^2 + \frac{1}{8} T^3 + \frac{3}{32} T^4 + \frac{21}{256} T^5 + \frac{5}{64} T^6 + \frac{1287}{16384} T^7 + \frac{21}{256} T^8 \right. \\ \left. + \frac{46189}{524288} T^9 + O(T^{10}), -\frac{1}{4} T^2 - \frac{1}{4} T^3 - \frac{3}{16} T^4 - \frac{21}{128} T^5 - \frac{5}{32} T^6 - \right. \\ \left. \frac{1287}{8192} T^7 - \frac{21}{128} T^8 - \frac{46189}{262144} T^9 + O(T^{10}) \right] \Bigg]$$

## 2.3 Set and Ideal Operations

The implicit representation of algebraic sets suggests a possibility of dealing with algebraic sets by working on their corresponding defining ideals. The radical of an ideal generates set-theoretically the same algebraic set (possibly without multiple components).

```
> A:=mkImplAlgSet({x*(x^2+y^2-1), x*z, x^2+y^2+z^2-1}, {y,z,x});
      A:=algebraic_set([x(x^2+y^2-1), xz, x^2+y^2+z^2-1], [x,y,z])
```

```
> radA:=computeRadical(A);
```

```
      radA:=algebraic_set([x^2+y^2-1,z], [x,y,z]),
      algebraic_set([x,y^2+z^2-1], [x,y,z])
```

During the computation of the radical, the set was split into two components. We can compute the union of these two algebraic sets by computing the intersection (i.e. the least common multiple) of their ideals.

```
> G1:=implUnionLCM(radA);
      G1:=algebraic_set([x^2+y^2+z^2-1, xz, zy^2+z^3-z], [x,y,z])
```

This gives us the ideal we started with, hence we know that the ideal by which A was given is a radical ideal.

The set-theoretic difference of sets is computed by determining the quotients of ideals.

```
> H:=mkImplAlgSet([x, y^2+z^2-1], [x, y, z]);
      H:=algebraic_set([x, y^2+z^2-1], [x,y,z])
```

We can compute the difference of the sets A and H and form the Zariski closure.

```
> implDifference(A,H);
      algebraic_set([x^2+y^2-1,z], [x,y,z])
```

## 2.4 Computing Dimensions and Irreducible Decomposition

The dimension is an important invariant of algebraic sets. The system CASA allows to compute the dimension of sets given in implicit representation. Let us define an algebraic set in three dimensions.

```
> N:=mkImplAlgSet([-y^2*z^2 + 10*y^2*z - y*z^2 - 25*y^2 + 6*y*z + x - 5*y + 1,
> y^3*z^2 - 10*y^3*z + 25*y^3 - y^2*z^2 + 5*y^2 - y*z + y]);
      N:=algebraic_set([-y^2 z^2 + 10 z y^2 - y z^2 - 25 y^2 + 6 z y + x - 5 y + 1,
      z^2 y^3 - 10 z y^3 + 25 y^3 - z y^2 + 5 y^2 - z y + y], [x,y,z])
```

N turns out to be a curve.

```
> dimension(N);
```

1

There are two possibilities to compute the irreducible decomposition of N. The first one is based on characteristic sets and works directly on the implicit representation of N.

```
> N;
```

```
      algebraic_set([-y^2 z^2 + 10 z y^2 - y z^2 - 25 y^2 + 6 z y + x - 5 y + 1,
      z^2 y^3 - 10 z y^3 + 25 y^3 - z y^2 + 5 y^2 - z y + y], [x,y,z])
```

```
> U:=decompose(N);
```

```
U := algebraic_set([x+1,y],[x,y,z]), algebraic_set([
-11 y^3 x - 76 y^3 + x^2 y^3 - 15 y^2 x - 15 y^2 - 3 y - 3 y x - 1 - x, 2 x^3 y^2
+ 2 x z - 1064 y^2 - 306 y^2 x - 108 - 8 x^2 y^2 - 16 y x^2 - y x^3 - 197 y x
- 44 x - 438 y + x^2 z + 65 z,
20 z y + x z + 5 z - 152 y^2 - 22 y^2 x + 2 x^2 y^2 - y x^2 - 19 y x - 54 y - 4
, 4 z^2 + x z - 23 z - 2 x^2 y^2 + 22 y^2 x + 152 y^2 + 106 y + 41 y x + 36
+ 16 x - y x^2], [x, y, z])
```

The second decomposition algorithm works on algebraic sets in projected form by factoring the defining polynomial of the hyper-surface.

```
> decompose(convertRep(N, impl, proj));
```

```
algebraic_set([[x+1], [-(266 x z^3 - 232 x z^2 - 104 x z^4 - 9 x z + 95 x
+ 17 x z^5 - x z^6 - 11 z^2 + 9 z^3 - 55 z + 49 - z^4 + z^2 x^2 - 10 x^2 z
+ 25 x^2) / (%1), -(-11 z^2 + 9 z^3 - 55 z + 49 - z^4 - 10 x z^2 - 65 x z
+ 74 x + 9 x z^3 - x z^4 + z^2 x^2 - 10 x^2 z + 25 x^2) / (%1), z]], [x, z]),
algebraic_set([[z^2 x^2 - 10 x^2 z + 25 x^2 - 3 x z^3 + 34 x z^2 - 114 x z
+ 95 x - z^5 + 13 z^4 - 59 z^3 + 126 z^2 - 174 z + 91], [-(266 x z^3
- 232 x z^2 - 104 x z^4 - 9 x z + 95 x + 17 x z^5 - x z^6 - 11 z^2 + 9 z^3
- 55 z + 49 - z^4 + z^2 x^2 - 10 x^2 z + 25 x^2) / (%1), -(-11 z^2 + 9 z^3
- 55 z + 49 - z^4 - 10 x z^2 - 65 x z + 74 x + 9 x z^3 - x z^4 + z^2 x^2
- 10 x^2 z + 25 x^2) / (%1), z]], [x, z])
%1 := -257 z^3 + 222 z^2 + 103 z^4 - 56 z - 21 - 17 z^5 + z^6
```

## 2.5 Algebraic Curves

We define an irreducible projective algebraic curve.

```
> C1:=mkImplAlgSet([-15*y^2*z^3-76*y^3*z^2-z^5-3*y*z^4-15*x*y^2*z^2-11*x*y^3*z-x*z
^4-3*x*y*z^3+y^3*x^2],[x,y,z],[basespace=projective]);
```

```
C1 := algebraic_set([-15 y^2 z^3 - 76 z^2 y^3 - z^5 - 3 z^4 y - 15 x z^2 y^2 - 11 x y^3 z
- x z^4 - 3 x y z^3 + x^2 y^3], [x, y, z])
```

In the course of studying basic properties of C1 we will look for singular points. The result of the following function call is a list of all singularities of C1 (counted properly). Usually, algebraic numbers are introduced in this process.

```
> singularities(C1);
```

```
table([
2 = [[40
21 RootOf(5_Z^2 + 4_Z + 89) - 5
21,
- 2
21 RootOf(5_Z^2 + 4_Z + 89) - 5
21, 1], [0, 1, 0]]
3 = [[1, 0, 0]]
])
```

Let us take a point on C1, (2:alpha:1), say, "alpha" being a root of the irreducible polynomial:

$$94 y^3 + 45 y^2 + 9 y + 3.$$

```
> alias(alpha=RootOf(94*y^3+45*y^2+9*y+3));
I, alpha
```

The tangent space (i.e. the set of points on all lines which have the intersection multiplicity  $> 1$ ) at this point is the following algebraic set (a line):

```
> tangSpace(C1,[2,alpha,1]);

algebraic_set (
  [-657 y + 643 z alpha + 322 alpha^2 z + 48 z - 48 x + 14 alpha x - 322 alpha^2 x],
  [x, y, z])
```

Since the projective point (0:1:0) is a singular point on C1 the tangent space is the whole projective space.

```
> Ts:=tangSpace(C1,[0,1,0]);
Ts := algebraic_set([0],[x,y,z])
```

The tangent cone is a linear subspace generating the tangent space. It is given by the following set.

```
> Tc:=tangSpace(C1,[0,1,0],'cone');
Some of the generating polynomials for the affine tangent space/cone
are homogeneous. Returning only affine form of generators.
%
```

```
Tc := algebraic_set([-11 x z + x^2 - 76 z^2],[x,z])
```

The set Tc splits over the algebraic closure of the rational numbers into two lines.

```
> decompose(Tc,absolute);

algebraic_set([x + 1/4 RootOf(44_Z + _Z^2 - 1216) z],[x,z]), al\
gebraic_set([x + (-11 - 1/4 RootOf(44_Z + _Z^2 - 1216)) z],[x,z])
```

It is an interesting question whether coordinates of points on an algebraic set can be expressed as rational functions of some number of parameters. For curves this amounts to represent coordinates of each point in terms of a single parameter. A necessary and sufficient condition is the vanishing of the genus.

```
> genus(C1);
0
```

Now we are able to compute a birational mapping of C1 to the line. We get a parametric description.

```
> convertRep(C1,para);

algebraic_set([[-t^3 - 15 t^2 - 75 t - 126, - (5 + t) / (t^2 + 11 t + 26), 1],[t])
```

Often it is desirable to determine (systems of) curves passing through specified points with prescribed multiplicity. The system CASA offers some functions to perform this task, e.g., `passGenCurve`.

## 2.6 Gröbner Bases and Syzygies

Besides the improved classical Gröbner bases algorithm, CASA contains an implementation of Gröbner bases for multivariate polynomial modules and is able to solve systems of linear equations with polynomial coefficients. CASA also contains an implementation of the Gröbner Walk algorithm for Gröbner bases conversion.

```
> msolveGB([[x^3-y,-x+y],[x*y-1,x^3-1],[-x*y^2,y-1]],[x*y^2-y^2,-x^5+x^2-x*y^
> 2],[x,y],term,tdeg);

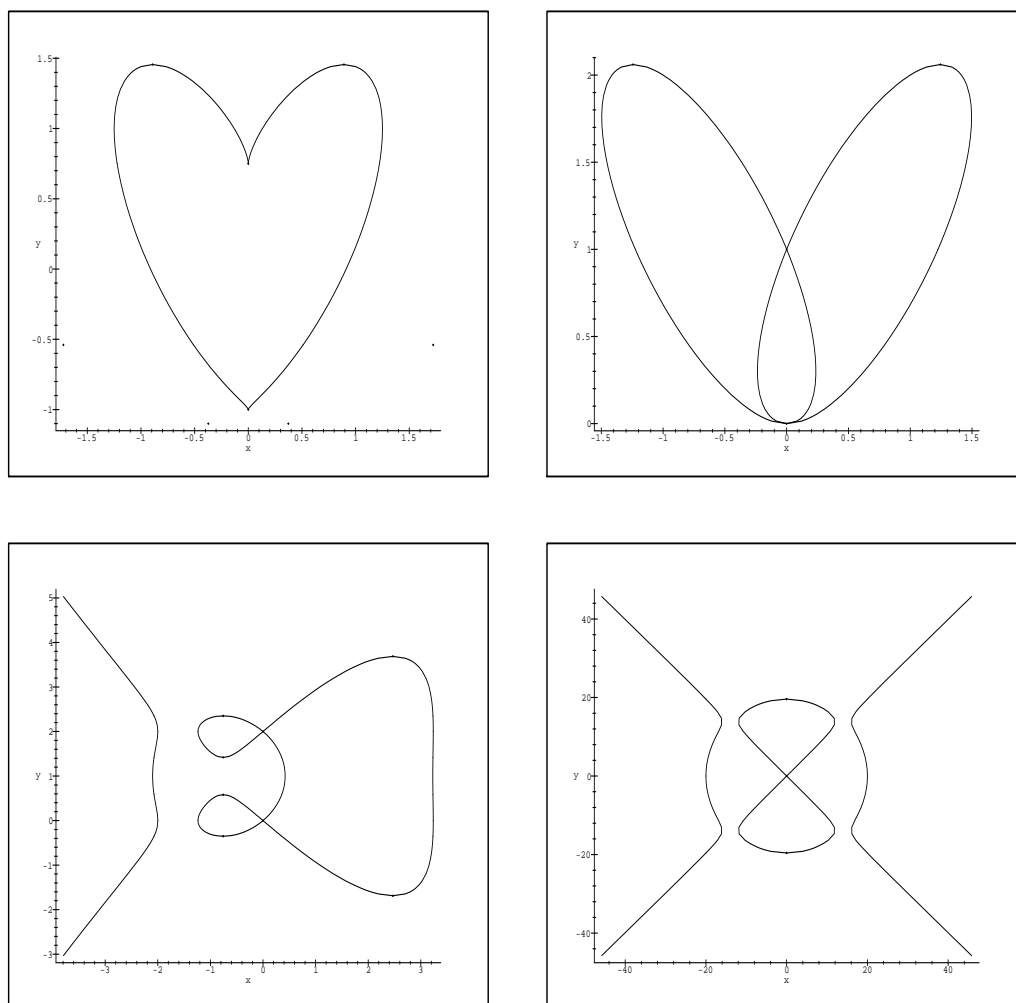
[[y, -x^2, x], [[-x^4 + x + x^3 y^2 - y^2 - x y^2 + x y + y - 1,
x y + x^3 y - x^3 - y^2 - y^3 + y - x^2 + x y^2,
x^3 + x^3 y + x y^2 - x^2 y + x - 2 y - x^6]]]
```

## 2.7 Scientific Visualization

A powerful feature of CASA is the capability to visualize algebraic sets. For the majority of tasks, only singular or other distinguished points of algebraic sets yield some interesting information. These objects have to be treated with care, requiring special analysis. CASA makes a thorough effort to obtain correct information about the local topology in the neighborhood of singularities and other critical points while still keeping the algorithm efficient by using hybrid symbolic-numerical methods. Although Maple provides a function which is claimed to plot sets in an implicit representation its output for more complicated varieties is by far not satisfactory. Figure 1 shows the graphs produced by CASA for the following examples:

1.  $\frac{93392896}{15625}x^6 + (\frac{94359552}{625}y^2 + \frac{91521024}{625}y - \frac{249088}{125}x^4 + (\frac{1032192}{25}y^4 - 36864y^3 - \frac{7732224}{25}y^2 - 207360y + \frac{770048}{25}x^2 + (65536y^6 + 49152y^5 - 135168y^4 - 72704y^3 + 101376y^2 + 27648y - 27648$
2.  $2x^4 - 3x^2y + y^2 - 2y^3 + y^4$
3.  $y^4 - 96a^2y^2 + 100a^2x^2 - x^4$
4.  $8(y^2 - x^2) - 8(y^3 + x^3) + 2y^4 + x^5$

Figure 1: Plotting



### 3 Distribution and Requirements

The system can be obtained by anonymous ftp at the URL `ftp.risc.uni-linz.ac.at`. Major releases will be located in the directory `/pub/CASA`. Bug correction and minor updates will be put in `/pub/CASA/update`. We have also a home page on World Wide Web `http://www.risc.uni-linz.ac.at/software/casa`.

The system is freely distributed under the following conditions: any research activity which uses CASA should cite the authors and the system explicitly. The system can be freely distributed to other users. New users are encouraged to notify the CASA coordinator so they can be included in a user list where they will be kept up to date about the progress of the system.

Bug reports, questions and suggestions should be sent to the e-mail address `alggeo@risc.uni-linz.ac.at`.

### References

- [1] R. Gebauer, M. Kalkbrener, B. Wall, and F. Winkler. CASA: A Computer Algebra Package for Constructive Algebraic Geometry. In S. M. Watt, editor, *ISSAC '91*, pages 403–410, Bonn, Germany, July 1991.
- [2] M. Mňuk. *Algebraic and Geometric Approach to Parameterization of Rational Curves*. PhD thesis, Research Institute for Symbolic Computation, Linz, Austria, 1995.
- [3] M. Mňuk and F. Winkler. CASA - a system for computer aided constructive algebraic geometry. In *Proceedings of the International Symposium DISCO'96*, pages 297–307, 1996.
- [4] Q.-N. Tran. A hybrid symbolic-numerical method for tracing surface-to-surface intersections. In A. H. M. Levelt, editor, *Proceedings of ISSAC-95*, Montreal, Canada, 1995. ACM.
- [5] Q.-N. Tran. On the symbolic-numerical methods for finding the roots of an arbitrary system of non-linear algebraic equations. In *Proceedings of the First Asian Technology Conference in Mathematics, ATCM-95*, Singapore, 1995.
- [6] Q.-N. Tran. *A Hybrid Symbolic-Numerical Approach in Computer Aided Geometric Design (CAGD) and Visualization*. PhD thesis, Research Institute for Symbolic Computation (RISC-Linz), University of Linz, Austria, 1996.
- [7] Q.-N. Tran. Extending newton's method for finding the roots of an arbitrary system of equations and its applications. *International Journal of Modeling and Simulation*, 17(4), 1997. To appear.
- [8] B. Wall. *Symbolic Computation with Algebraic Sets*. PhD thesis, RISC-Linz, Universität Linz, 1993.
- [9] F. Winkler. Algebraic computation in geometry. In G. Jacobs, N. E. Oussous, and S. Steinberg, editors, *IMACS'93 Conference, Lille, special issue J. Math. in Comp. Simulation*, 1993.