Computational Algebraic Geometry

Thomas Markwig
Fachbereich Mathematik
Technische Universität Kaiserslautern

A short course taught at the
EMALCA 2010 in Villahermosa, Mexico

August 2010

Contents

Some comments on the origin of the images

Most of the images were created by myself with the programme `surfex` [HL08] or the LaTeX-package `texdraw`. The singular plane curve in Figure 3 was produced by Rüdiger Stobbe at the University of Kaiserslautern many years ago, and the space curve in Figure 4 was produced with the aid of `Maple`. The images in Figure 34 and 35 are snapshots of the programme `surfex` produced by `xv`.

The image in Figure 5 can be found on the web page:

http://de.wikipedia.org/wiki/Stewart_Platform

as well as the first image in Figure 6. The second image in Figure 6 can be found on the web page:

http://www.physikinstrumente.de/de/produkte/primages.php?sortnr=700800

The image in Figure 9 was obtained from the web page:

http://static.howstuffworks.com/gif/microprocessor-athlon-64.jpg

Finally the image in Figure 10 was taken from the web page:

http://en.wikipedia.org/wiki/Sudoku

1 Introduction

We want to study the *solution set* $V(f_1, \ldots, f_k)$ of a system of equations

$$\begin{aligned} f_1(x_1, \ldots, x_n) &= 0 \\ &\vdots \\ f_k(x_1, \ldots, x_n) &= 0 \end{aligned}$$

where the f_i are polynomials.

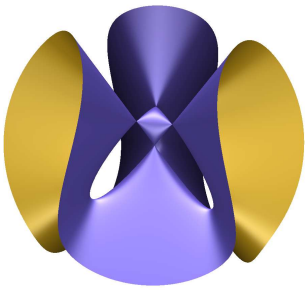


Figure 1. The Cayley Cubic

$$x_1x_2 + x_1x_3 + x_2x_3 + x_1x_2x_3 = 0.$$

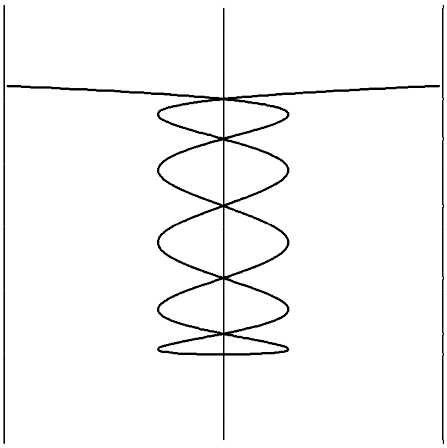


Figure 3. A plane curve

$$\begin{aligned} 32x_1^2 - 2097152x_2^{11} + 1441792x_2^9 \\ - 360448x_2^7 + 39424x_2^5 \\ - 1760x_2^3 + 22x_2 - 1 = 0 \end{aligned}$$

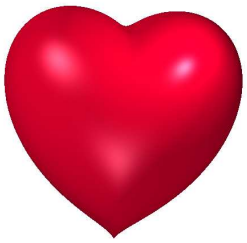


Figure 2. The Heart

$$\left(2x_1^2 + x_2^2 + x_3^2 - 1\right)^3 - \tfrac{1}{10}x_1^2x_3^3 - x_2^2x_3^3 = 0$$

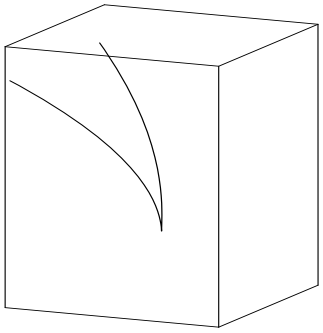


Figure 4. A Space Curve

$$\begin{aligned} x_1 - x_3^2 &= 0 \\ x_2^2 - x_3^3 &= 0 \end{aligned}$$

The first case to consider would be $n = k = 1$, that is, we have a single polynomial

$$f = a_m \cdot x^m + a_{m-1} \cdot x^{m-1} + \ldots + a_1 \cdot x + a_0$$

and we are looking for the *roots* of that polynomial. In theory, this is not too hard to deal with. There are at most n roots, and if the base field is algebraically closed

like $\mathbb{C}$, then there are exactly n roots counted with the appropriate multiplicity. However, it is very hard in practise to find any of the roots.

The second case to consider would have n and k arbitrary, but the polynomials are all linear, i.e. they are of the form

$$f_i = a_{i1} \cdot x_1 + \ldots + a_{in} \cdot x_n + b_i.$$

Then the solution set is an *affine vector space*, and it is even easy to compute using the Gauß algorithm.

In general, life is much more difficult and much more interesting as already the above examples show. Before introducing the theory we want to mention some applications where such polynomial systems of equations come up naturally.

A) **Robotics**

A frequently used robotic system is the hexapod (see Figure 5). A platform is

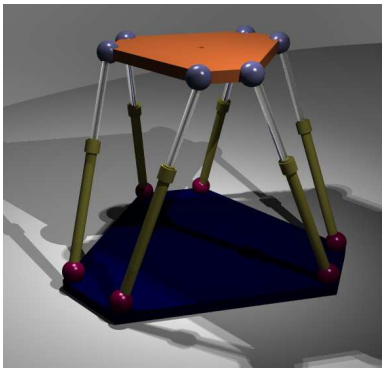


FIGURE 5. Schematic image of a hexapod robot

connected to six legs via rotational joints. The length of the legs can be varied in order to adjust the position of the platform in space. Note that one has three translational and three rotational directions of movement, that is one has six degrees of freedom, and fixing the length of the six legs one thus expects to fix the position of the platform. Robots using this basic scheme are used in many applications, e.g.

FIGURE 6. Applications: Flight simulator and micro surgery

for flight simulators or in micro surgery (see Figure 6). To deduce the length of the legs needed in order to achieve a given position one has to solve a system of eight polynomial equations, one of which is shown in Figure 7.

$$
\begin{aligned}
&2x_0^4 + 4x_0^2x_1^2 + 2x_1^4 + 4x_0^2x_2^2 + 4x_1^2x_2^2 + 2x_2^4 + 4x_0^2x_3^2 + 4x_1^2x_3^2\\
&+ 4x_2^2x_3^2 + 2x_3^4 - 2x_0^2x_2y_0 - 2x_1^2x_2y_0 - 2x_2^3y_0 - 2x_0^2x_3y_0\\
&- 2x_1^2x_3y_0 - 2x_2^2x_3y_0 - 2x_2x_3^2y_0 - 2x_3^3y_0 + x_0^2y_0^2 + x_1^2y_0^2\\
&+ x_2^2y_0^2 + x_3^2y_0^2 + 2x_0^2x_2y_1 + 2x_1^2x_2y_1 + 2x_2^3y_1 - 2x_0^2x_3y_1\\
&- 2x_1^2x_3y_1 - 2x_2^2x_3y_1 + 2x_2x_3^2y_1 - 2x_3^3y_1 + x_0^2y_1^2 + x_1^2y_1^2\\
&+ x_2^2y_1^2 + x_3^2y_1^2 + 2x_0^3y_2 - 2x_0^2x_1y_2 + 2x_0x_1^2y_2 - 2x_1^3y_2\\
&+ 2x_0x_2^2y_2 - 2x_1x_2^2y_2 + 2x_0x_3^2y_2 - 2x_1x_3^2y_2 + x_0^2y_2^2 + x_1^2y_2^2\\
&+ x_2^2y_2^2 + x_3^2y_2^2 + 2x_0^3y_3 + 2x_0^2x_1y_3 + 2x_0x_1^2y_3 + 2x_1^3y_3\\
&+ 2x_0x_2^2y_3 + 2x_1x_2^2y_3 + 2x_0x_3^2y_3 + 2x_1x_3^2y_3 + x_0^2y_3^2 + x_1^2y_3^2\\
&+ x_2^2y_3^2 + x_3^2y_3^2 + 4x_0x_2 + 4x_1x_2 - 4x_0x_3 + 4x_1x_3\\
&- 2x_1y_0 + 2x_0y_1 - 2x_3y_2 + 2x_2y_3 - \tfrac{7}{2} = 0
\end{aligned}
$$

FIGURE 7. One equation for the leg-length of a hexapod

B) **Elliptic curve cryptography**

Equations like

$$y^2z - x \cdot \left(x - \tfrac{1}{2} \cdot z\right) \cdot (x - z) = 0$$

define *elliptic curves* in the projective plane (see Figure 8). They carry in a natural way the structure of an abelian group, that is, one can add and subtract points on the curve from each other. If the base filed K over which one considers the solutions is finite, then these groups are very well suited as alphabets for cryptographical methods, and they are widely used nowadays. Note that, while it is easy to add a point P several times to itself, e.g. $P + P + P = 3 \cdot P$, it is very hard to find P if you are given the result $3 \cdot P$. One says, that it is very hard to compute the discrete logarithm.

C) **Coding theory**

Also for coding theory one can use projective curves over a finite field K. However, this time one considers the global sections of a divisor on the curve and its image under the map evaluating the sections at n fixed points. This gives a subspace of K^n which serves as the code and has pretty good properties. These ideas go back to Goppa.

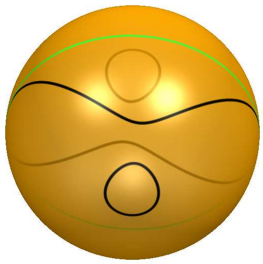


FIGURE 8. A visualisation of $y^2z - x \cdot \left(x - \frac{1}{2} \cdot z\right) \cdot (x - z) = 0$

D) **Chip design**

A modern micro processor chip contains several layers of electrical circuits with millions of transistors, resistors and capacitors. The functionality of certain units of such an electrical circuit within itself and with other units leads eventually to systems of partial differential equations. These can be treated symbolically as polynomial equations which one uses in a preprocessing step before applying numerical methods. Certain properties of the circuits can thus be predicted by pure simulation and computations without actually building the chip (see Figure 9).

FIGURE 9. A micro processor chip

E) **Sudoku**

A Sudoku is a scheme of digits as in Figure 10. One should fill the table by digits

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

FIGURE 10. Sudoku

in such a way that in each row, in each column and in each block each of the nine digits $1, \dots, 9$ occurs precisely once.

Let us now associate coordinates $x_1, \dots, x_{81}$ to each of the 81 squares and consider the set of pairs

$$B = \{(i,j) \mid 1 \le i < j \le 81, x_i \text{ and } x_j \text{ are in the same row, column or block}\}.$$

For $i = 1, \dots, 81$ we define

$$F_i = \prod_{k=1}^{9} x_i - k,$$

and for $(i,j) \in B$ we set

$$G_{ij} = \frac{F_i - F_j}{x_i - x_j}.$$

Then the F_i and the G_{ij} are polynomials. Moreover, $a_1, \dots, a_{81}$ is a feasible solution for a sudoku if and only if it is a solution of the system of equations

$$G_{ij} = 0, \quad F_k = 0 \quad \text{where } (i,j) \in B, k = 1, \dots, 9.$$

F) **Exercises**

Exercise 1.1
How do the solution sets in $\mathbb{R}^2$ of the following equations look like?

a. $y - x^2 = 0$.

b. $y - x^4 + 1 = 0$.

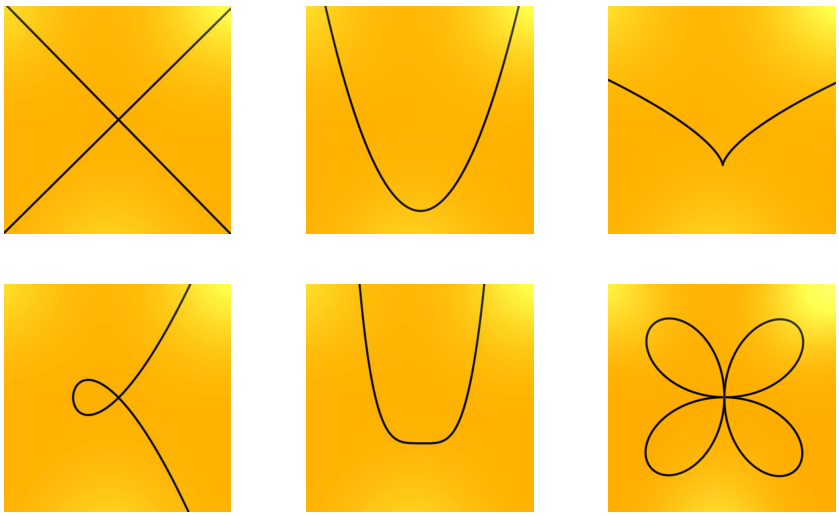
c. $y^2 - x^2 = 0$.

d. $x^2 - y^3 = 0$.

e. $y^2 - x^2 - x^3 = 0$.

f. $(x^2 + y^2)^3 - 4x^2y^2 = 0$.

Which image belongs to which equation?



Exercise 1.2

How do the solution sets in $\mathbb{R}^3$ of the following systems of equations look like?

a. $x^2 - y^3 = 0$.

b. $z - x^2 - y^2 = 0$.

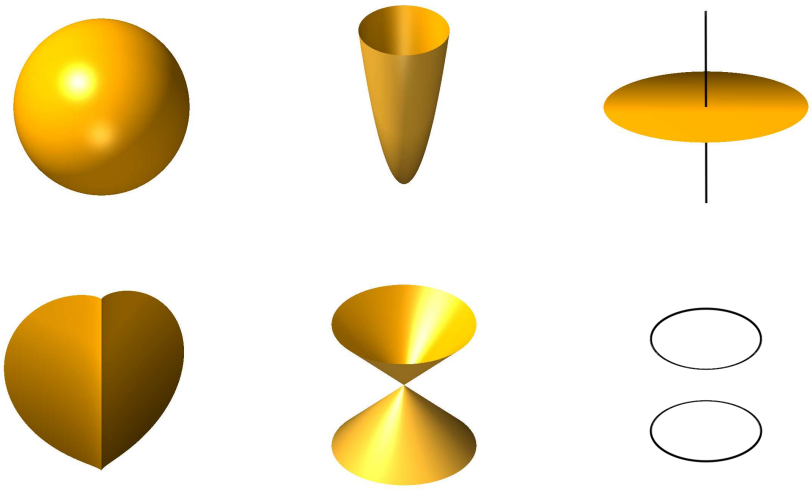
c. $z^2 - x^2 - y^2 = 0$.

d. $z^2 - x^2 - y^2 = 0$ and $z^2 - 1 = 0$.

e. $xz = 0$ and $yz = 0$.

f. $x^2 + y^2 + z^2 - 1 = 0$.

Which image belongs to which system of equations?



2 Affine algebraic varieties

Studying solution sets of systems of polynomial equations does not in general mean that we want to write down all solutions. That is in general impossible even if the number is finite which in general is not the case. Instead we want to study the structure of these sets, properties such as its dimension or if it decomposes into several components or how it looks locally at some point. The latter is important if one wants to control parameters which allow to travel along the solution set without fearing any sudden and instable behaviour. In this section we will introduce the basic notions needed to deal with the theory of the solution sets of finite systems of polynomial equations, called affine algebraic varieties.

Throughout these lecture notes K will always denote a field.

For computations it is best to choose $K = \mathbb{Q}$ or $K = \mathbb{Z}/p\mathbb{Z}$ for some prime number p; for visualisations it is best to choose $K = \mathbb{R}$; for applications one usually needs $K = \mathbb{R}$ or some finite field of size p^n for some large prime number p and some large integer n; finally, the theory works best when K is algebraically closed like $K = \mathbb{C}$.

We will always choose whatever field is best for the purpose at hand!

A) **Basic definitions**

Definition 2.1 (The polynomial ring)
We define the *polynomial ring* in n indeterminates $x_1, \ldots, x_n$ over K as

$$K[\mathbf{x}] = K[x_1, \ldots, x_n] = \left\{ \sum_{|\alpha|=0}^{d} a_\alpha \cdot \mathbf{x}^\alpha \;\middle|\; d \in \mathbb{N}, a_\alpha \in K \right\},$$

where we use the notation

- $\mathbf{x} = (x_1, \ldots, x_n)$,
- $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$,
- $\mathbf{x}^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$,
- $|\alpha| = \alpha_1 + \ldots + \alpha_n$.

This set is a commutative ring with one via the addition

$$\sum_\alpha a_\alpha \cdot \mathbf{x}^\alpha + \sum_\alpha b_\alpha \mathbf{x}^\alpha = \sum_\alpha (a_\alpha + b_\alpha) \cdot \mathbf{x}^\alpha$$

and the multiplication

$$\sum_\alpha a_\alpha \cdot \mathbf{x}^\alpha \cdot \sum_\beta b_\beta \mathbf{x}^\beta = \sum_\gamma \sum_{\alpha+\beta=\gamma} a_\alpha \cdot b_\beta \cdot \mathbf{x}^\gamma.$$

The number

$$\deg\left(\sum_{\alpha} a_\alpha \cdot x^\alpha\right) = \sup\left\{|\alpha| \;\middle|\; a_\alpha \neq 0\right\} \in \mathbb{N} \cup \{-\infty\}$$

is the *degree* of the polynomial $\sum_\alpha a_\alpha \cdot x^\alpha$.

Remark 2.2 (Properties of the polynomial ring)
Let us gather some easy and well known facts about the polynomial ring $K[x]$.

a. For $f, g \in K[x]$ we have:
   - $\deg(f \cdot g) = \deg(f) + \deg(g)$,
   - f is a unit $\iff f \in K \setminus \{0\} \iff \deg(f) = 0$.

b. A non-constant polynomial f is said to be *irreducible* if it does not factor properly, i.e. $f = g \cdot h$ implies that either g or h is a unit. Otherwise it is said to be *reducible*.

c. An *ideal* in $K[x]$ is a non-empty subset I of $K[x]$ which is closed under addition and scalar multiplication, i.e.

$$f + g, r \cdot f \in I \quad \forall\, f, g \in I, r \in K[x].$$

   We denote this by $I \trianglelefteq K[x]$.

d. If $M \subseteq K[x]$ is any subset, then

$$\langle M\rangle := \bigcap_{M \subseteq I \trianglelefteq K[x]} I = \left\{ \sum_{i=1}^{k} r_i \cdot f_i \;\middle|\; r_i \in K[x], f_i \in M, k \in \mathbb{N} \right\}$$

   is the ideal *generated* by M. It is the smallest ideal containing M, and its elements are precisely the finite linear combinations of elements in M.

Definition 2.3 (Affine algebraic varieties)

a. We call $\mathbb{A}^n_K = K^n = \{(a_1, \ldots, a_n) \mid a_i \in K\}$ the *affine n-space*.

b. For any subset $M \subseteq K[x]$ we define the *vanishing set* of M as

$$V(M) = \{p \in \mathbb{A}^n_K \mid f(p) = 0 \;\forall\, f \in M\}.$$

c. A subset $X \subseteq \mathbb{A}^n_K$ is an *affine algebraic variety* if it is the vanishing set $X = V(M)$ of some (not necessarily finite) set M of polynomials in $K[x]$.

Remark 2.4 (First properties of affine algebraic varieties)

a. $\mathbb{A}^n_K = V(0)$ and $\emptyset = V(1)$ are affine algebraic varieties.

b. If $M \subseteq N \subseteq K[x]$, then $V(M) \supseteq V(N)$.

c. Since the polynomials in $\langle M\rangle$ are the linear combinations of polynomials in M we have

$$V(M) = V\big(\langle M\rangle\big).$$

   That is, every affine algebraic variety is the vanishing set of an ideal.

We started by considering *finite* systems of polynomial equations and their solution sets, that is by considering vanishing sets of finite sets of polynomials. The last remark encourages us to replace any given set of polynomials by the ideal which it generates and which in general will contain *infinitely* many polynomials. The ideals are more suitable for theoretical purposes and Hilbert's Basis Theorem states that we have not enlarged the universe of geometric objects we are interested in.

Theorem 2.5 (Hilbert's Basis Theorem)
Every ideal in $K[\mathbf{x}]$ *is finitely generated, and hence every affine algebraic variety is the solution set of a finite system of polynomial equations!*

Idea of the proof: One proves the statement by induction on the number n of variables. We set $R = K[x_1, \ldots, x_{n-1}]$ and consider the polynomials in $K[\mathbf{x}] = R[x_n]$ as polynomials in the variable x_n with coefficients in R. Given an ideal I in $K[\mathbf{x}]$ the ideal J in R generated by the leading coefficients of the elements in I is finitely generated by induction. That is, there are finitely many polynomials $f_1, \ldots, f_k \in I$ such that there leading coefficients generate J. If d is an upper bound for the degree of these polynomials, then one can show by some kind of division with remainder that I is generated by $f_1, \ldots, f_k$ together with the finite generating set of the ideal $\langle 1, x_n, x_n^2, \ldots, x_n^{d-1} \rangle_R \cap J$ in R. □

Example 2.6
It is easy to see that the set

$$I = \{f \in K[x, y] \mid f(1, 1) = 0\} \subset K[x, y]$$

is an ideal. By Hilbert's Basis Theorem it must be finitely generated, and it is actually just

$$I = \langle x - 1, y - 1 \rangle.$$

B) **Gröbner bases**

Remark 2.7
Suppose we are given a system of polynomials $f_1, \ldots, f_k$ and we are interested in their vanishing set $X = V(f_1, \ldots, f_k)$. Remark 2.4 then says that we may replace the given set of generators of $I = \langle f_1, \ldots, f_k \rangle$ by any other set of generators of I without changing the vanishing set X. The idea is now to change to a set of generators which reveals more information on the vanishing set.

Definition 2.8 (Monomial orderings)
A total ordering $>$ on the set

$$\mathrm{Mon}(\mathbf{x}) = \left\{ \mathbf{x}^\alpha \;\middle|\; \alpha \in \mathbb{N}^n \right\}$$

of *monomials* in the indeterminates $x_1, \ldots, x_n$ is a *monomial ordering* if and only if

$$\mathbf{x}^\alpha > \mathbf{x}^\beta \quad \Longrightarrow \quad \mathbf{x}^\alpha \cdot \mathbf{x}^\gamma > \mathbf{x}^\beta \cdot \mathbf{x}^\gamma \quad \forall\, \gamma \in \mathbb{N}^n.$$

That is, the ordering is compatible with the obvious semi group structure on $\mathrm{Mon}(\mathbf{x})$. A monomial ordering is called *global* if $1 < \mathbf{x}^\alpha$ for all $\alpha \neq 0$, and it is called *local* if $1 > \mathbf{x}^\alpha$ for all $\alpha \neq 0$.

Example 2.9 (Monomial orderings on $\mathrm{Mon}(x)$)
On $\mathrm{Mon}(x) = \{x^n \mid n \in \mathbb{N}\}$ there are exactly two monomial orderings defined by either

$$x^n > x^m \iff n > m$$

or

$$x^n > x^m \iff n < m.$$

The first one is global, the second one is local.

Example 2.10 (Global monomial orderings)

a. Define the *global lexicographical ordering* $>_{lp}$ on $\mathrm{Mon}(\mathbf{x})$ by $\mathbf{x}^\alpha >_{lp} \mathbf{x}^\beta$ if

$$\exists\, i \in \{1, \ldots, n\} : \alpha_1 = \beta_1, \ldots, \alpha_{i-1} = \beta_{i-1}, \text{ and } \alpha_i > \beta_i.$$

b. Define the *global degree reverse lexicographical ordering* $>_{dp}$ on $\mathrm{Mon}(\mathbf{x})$ by $\mathbf{x}^\alpha >_{dp} \mathbf{x}^\beta$ if

$$\deg\left(\mathbf{x}^\alpha\right) > \deg\left(\mathbf{x}^\beta\right),$$

or $\deg\left(\mathbf{x}^\alpha\right) = \deg\left(\mathbf{x}^\beta\right)$, but then

$$\exists\, i \in \{1, \ldots, n\} : \alpha_n = \beta_n, \ldots, \alpha_{i+1} = \beta_{i+1}, \text{ and } \alpha_i < \beta_i.$$

c. E.g., $x_1^2 >_{lp} x_1x_2^3$, but $x_1x_2^3 >_{dp} x_1^2$.

Definition 2.11 (Leading monomial)
Let $>$ be a monomial ordering on $\mathrm{Mon}(\mathbf{x})$ and $0 \neq f = \sum_{|\alpha|=0}^{d} a_\alpha \cdot \mathbf{x}^\alpha \in K[\mathbf{x}]$.

a. $\mathrm{lm}_>(f) = \max\left\{\mathbf{x}^\alpha \mid a_\alpha \neq 0\right\}$ is the *leading monomial* of f w.r.t. $>$.

b. $\mathrm{lc}_>(f) = a_\alpha$, if $\mathrm{lm}_>(f) = \mathbf{x}^\alpha$, is the *leading coefficient* of f w.r.t. $>$.

c. $\mathrm{lt}_>(f) = \mathrm{lc}_>(f) \cdot \mathrm{lm}_>(f)$ is the *leading term* of f w.r.t. $>$.

For the sake of completeness we define

$$\mathrm{lm}_>(0) := 0, \quad \mathrm{lt}_>(0) := 0, \quad \mathrm{lc}_>(0) := 0.$$

Example 2.12
We consider $\mathrm{Mon}(x, y)$ with the lexicographical respectively the degree reverse lexicographical ordering and the polynomial $f = 2x^2 + 5xy^3 - y$, then

$$\mathrm{lm}_{>_{lp}}(f) = x^2, \ \mathrm{lc}_{>_{lp}}(f) = 2, \ \mathrm{lt}_{>_{lp}}(f) = 2x^2,$$

while

$$\mathrm{lm}_{>_{dp}}(f) = xy^3, \ \mathrm{lc}_{>_{dp}}(f) = 5, \ \mathrm{lt}_{>_{dp}}(f) = 5xy^3.$$

Remark 2.13 (SINGULAR)
Below you find the SINGULAR commands for the last example. It is important to note that in SINGULAR one always has to specify the polynomial ring one is working with. The command

```
ring r=0,(x,y),dp;
```

does so. It attributes the name `r` to the ring, specifies the base field to be the rational numbers by stating the characteristic to be `0`, introduces the indeterminates `x` and `y`, and finally specifies the ordering to be `dp`, i.e. the global degree reverse lexicographical ordering. The remaining parts should be more or less self explaining.

```
                     SINGULAR                                 /
 A Computer Algebra System for Polynomial Computations       /   version 3-1-1
                                                           0<
     by: G.-M. Greuel, G. Pfister, H. Schoenemann            \   Feb 2010
FB Mathematik der Universitaet, D-67653 Kaiserslautern        \
> ring r=0,(x,y),dp;
> poly f=3*x^2+5*x*y^3-y;
> leadmonom(f);
xy3
> leadcoef(f);
5
> lead(f);
5xy3
```

Remark 2.14 (Standard representations for global monomial orderings)
If the leading monomial x^α of a polynomial f is divisible by the leading monomial x^β of a polynomial g (i.e. $\beta_i \leq \alpha_i$ for all i), then we can cancel out the leading term of f by the leading term of g, i.e.

$$h = f - \frac{lt_>(f)}{lt_>(g)} \cdot g$$

has a leading monomial which is strictly smaller than that of f. That is, we can write f as a monomial multiple of g plus some polynomial with a strictly smaller leading term:

$$f = \frac{lt_>(f)}{lt_>(g)} \cdot g + h.$$

Replacing f by h and continuing like this we can finally write f as a polynomial multiple of g plus some remainder r whose leading term is no longer divisible by $lm_>(g)$ (the termination of this process is guarantueed for global orderings by Exercise 2.44):

$$f = q \cdot g + r.$$

The same works if we replace g by several polynomials $g_1, \ldots, g_k$, i.e. successively canceling out leading terms by the leading term of some g_i we get a representation

$$f = q_1 \cdot g_1 + \ldots q_k \cdot g_k + r \tag{1}$$

where the leading monomial of r is no longer divisible by the leading monomial of any g_i and where by construction the leading monomial of $q_i \cdot g_i$ is never larger than $\mathrm{lm}_>(f)$. Such a representation of f is called a *standard representation* or a *division with remainder* of f with respect to G and $>$.

Definition 2.15 (Leading ideal)
Let $>$ be a monomial ordering on $\mathrm{Mon}(\boldsymbol{x})$.

a. For a subset $M \subseteq K[\boldsymbol{x}]$ we call the ideal

$$L_>(M) = \langle \mathrm{lm}_>(f) \mid f \in M \rangle \trianglelefteq K[\boldsymbol{x}]$$

generated by the leading monomials of the elements in M its *leading ideal*.

b. A finite subset $G = \{f_1, \ldots, f_k\}$ of an ideal $I \trianglelefteq K[\boldsymbol{x}]$ is a *Gröbner basis* or *standard basis* of I w.r.t. $>$, if the leading monomials of the f_i generate the leading ideal $L_>(I)$ of I.

Example 2.16
If $I = \langle x-1, y-1 \rangle$ and $>$ is any global monomial ordering on $\mathrm{Mon}(x, y)$, then $G = \{x-1, y-1\}$ is a Gröbner basis of I since $L_>(I) = \langle x, y \rangle$.

```
> ring r=0,(x,y),dp;
> ideal I=x-1,y-1;
> groebner(I);
_[1]=y-1
_[2]=x-1
> leadmonom(groebner(I));
_[1]=y
_[2]=x
```

Gröbner bases are *good generating systems*, as the following proposition shows.

Proposition 2.17 (Ideal membership test)
Let I *be an ideal in* $K[\boldsymbol{x}]$, $G = \{g_1, \ldots, g_k\} \in K[\boldsymbol{x}]$ *and* $>$ *a monomial ordering.*

a. *If* G *is a Gröbner basis of* I*, then* G *generates* I*, i.e.* $I = \langle G \rangle$.

b. *If* G *is a Gröbner basis of* I *and* $f = \sum_{i=1}^{k} q_i \cdot g_i + r$ *is a standard representation, then*

$$f \in I \iff r = 0.$$

Idea of the proof: If $f \in I$ and $f = \sum_{i=1}^{k} q_i \cdot g_i + r$ is a standard representation, then $r = f - \sum_{i=1}^{k} q_i \cdot g_i$ is in I and thus its leading monomial is divisible by the leading monomial of some g_i if G is a Gröbner basis. Thus r must be zero and the g_i generate I. □

Example 2.18
Let $I = \langle x - y^2, x + y^2 \rangle$ and consider the lexicographical ordering $>_{lp}$ on $\mathrm{Mon}(x, y)$. Then

$$y^2 = 0 \cdot (x - y^2) + 0 \cdot (x + y^2) + y^2$$

is a standard representation with $r = y^2 \neq 0$, however

$$y^2 = \frac{1}{2} \cdot (x + y^2) - \frac{1}{2} \cdot (x - y^2) \in I.$$

This shows that $G = \{x - y^2, x + y^2\}$ is not a Gröbner basis of I.

Example 2.19 (Ideal membership and syzygies)
Suppose we want to find a standard representation of the polynomial $f = x^4 + y^4$ with respect to $G = \{x - 1, y - 1\}$ and to $>_{dp}$ on $\mathrm{Mon}(x, y)$. The command

```
reduce(f,I);
```

gives us the remainder `r`. And in the `Singular` code below we see, how the command `syz` can be used to find q_1 and q_2.

```
> ring r=0,(x,y),dp;
> ideal I=x-1,y-1;
> poly f=x4+y4;
> reduce(f,I);
2
> ideal J=I,f-reduce(f,I);
> print(syz(J));
-y+1,x3+x2+x+1,
x-1, y3+y2+y+1,
0,   -1
```

The last column of the matrix that we get, has a -1 as its last entry. Thus the entries above are $q_1 = x^3 + x^2 + x + 1$ in the first row and $q_2 = y^3 + y^2 + y + 1$ in the second row. This comes from the fact that each column (a_1, a_2, a_3) represents polynomials such that

$$a_1 \cdot (x - 1) + a_2 \cdot (y - 1) + a_3 \cdot (f - r) = 0.$$

One calls such a vector (a_1, a_2, a_3) a *syzygy* of the polynomials $x - 1, y - 1, f - r$.

C) **Finite solution sets and Gröbner bases**

Proposition 2.20 (Finite affine algebraic varieties)
Let I *be an ideal in* $K[x]$. *Then* $V(I)$ *is finite if and only if for each* $i = 1, \ldots, n$

$$I \cap K[x_i] = \langle f_i \rangle \neq \{0\}.$$

Moreover, then $V(I) \subseteq \{(p_1, \ldots, p_n) \mid f_i(p_i) = 0, i = 1, \ldots, n\}$.

Idea of the proof: If $V(I)$ is finite and f_i is the polynomial that has all the i-th coordinates of points in $V(I)$ as zero, then one expects some power of f_i in I. $\square$

Remark 2.21
We now want to explain how one can easily compute $I \cap K[x_i]$ using Gröbner bases. Reorder the variables $x_1, \ldots, x_n$ in such a way, that x_i is the last one and call the new variables $y_1, \ldots, y_n$. Then compute a Gröbner basis G of I with respect to the lexicographical ordering on $\mathrm{Mon}(y_1, \ldots, y_n)$. There will be at least one polynomial in G which only depends on the variable $y_n = x_i$, and among those only depending on that variable the one of lowest degree will be a generator of $I \cap K[x_i]$.
There is actually a simple SINGULAR command, `eliminate`, which allows to compute the generators directly. We will come back to this command further down.

Example 2.22 (Elimination of variables)
Let us consider the ideal $I = \langle x^2 + y^2 - 1, xy \rangle$ in $K[x, y]$. Then the following SINGULAR commands can be used to find generators for $I \cap K[x]$ and $I \cap K[y]$:

```
> ring r=0,(x,y),lp;
> ideal i=x2+y2-1,xy;
> groebner(i);
_[1]=y3-y
_[2]=xy
_[3]=x2+y2-1
> ring r=0,(y,x),lp;
> ideal i=x2+y2-1,xy;
> groebner(i);
_[1]=x3-x
_[2]=yx
_[3]=y2+x2-1
```

We thus see that

$$I \cap K[x] = \langle x^3 - x \rangle$$

and

$$I \cap K[y] = \langle y^3 - y \rangle.$$

This implies that

$$V(I) \subseteq V(x^3 - x) \times V(y^3 - y) = \left\{ (a, b) \;\middle|\; a, b \in \{-1, 0, 1\} \right\},$$

but since $V(I)$ is the intersection of the unit circle with the coordinate axes we know that indeed

$$V(I) = \{(0, 1), (0, -1), (1, 0), (-1, 0)\}.$$

It would of course be an easy task to verify which of the above (a, b) is actually in $V(I)$ by just evaluating the generators of I at each (a, b).

We could have computed the generator of $I \cap K[x]$ by eliminating the variable y with the aid of the command `eliminate`:

```
> ring r=0,(x,y),lp;
> ideal i=x2+y2-1,xy;
> eliminate(i,y);
_[1]=x3-x
```

Remark 2.23 (The commands `factorize` and `solve`)
The idea that one can find the solutions of $V(I)$ as above if there are only finitely many, depends of course on whether we can find the roots of the univariate polynomials f_i, which in general is impossible.

However, if the solutions happen to be all rational numbers, then one can factorise the polynomial.

```
> ring r=0,x,dp;
> poly f=(x-1)*(x-4)*(x-5)^2*(x+3)^3*(x^2+1);
> f;
x9-6x8-28x7+162x6+314x5-1254x4-1412x3+1278x2-1755x+2700
> factorize(f);
[1]:
   _[1]=1
   _[2]=x+3
   _[3]=x-5
   _[4]=x-1
   _[5]=x2+1
   _[6]=x-4
[2]:
   1,3,2,1,1,1
```

The command `factorize` applied to a univariate polynomial factorises this polynomial into its irreducible factors. The output is a list containing a constant coefficient and the irreducible factors together with a vector of integers containing the corresponding multiplicities of the factors. E.g. the second list entry $x + 3$ has as multiplicity the second entry 3 in the vector.

The command works also if the base field is a field of type $\mathbb{Z}/p\mathbb{Z}$, e.g. in $\mathbb{Z}/2\mathbb{Z}[x]$ we have $x^8 + 1 = (x + 1)^8$.

```
> ring s=2,x,lp;
> poly f=x8+1;
> factorize(f);
[1]:
   _[1]=1
   _[2]=x+1
[2]:
   1,8
```

One can also try to approximate the solutions if they are not rational using the command `solve` from the library `solve.lib`. A library is loaded using the command `LIB` as in the following example.

```
> LIB "solve.lib";
> ring r=0,x,dp;
> poly g=(x^2+1)*(x^2-2)*(x+5);
> g;
x5+5x4-x3-5x2-2x-10
> solve(g);
[1]:
   -5
[2]:
   -1.41421356
[3]:
   1.41421356
[4]:
   -i
[5]:
   i
```

D) **Hilbert's Nullstellensatz**

Definition 2.24 (Radicals, prime ideals and vanishing ideals)

a. A strict ideal $\mathsf{P} \lhd \mathsf{K}[\mathbf{x}]$ is a *prime ideal* if and only if $\mathsf{K}[\mathbf{x}]/\mathsf{P}$ is an integral domain, i.e. $\mathfrak{a} \cdot \mathfrak{b} \in \mathsf{P}$ implies $\mathfrak{a} \in \mathsf{P}$ or $\mathfrak{b} \in \mathsf{P}$.
We denote by $\mathrm{Spec}(\mathsf{K}[\mathbf{x}])$ the set of all prime ideals of $\mathsf{K}[\mathbf{x}]$ and call this the *spectrum* of $\mathsf{K}[\mathbf{x}]$.

b. A strict ideal $\mathfrak{m} \lhd \mathsf{K}[\mathbf{x}]$ is a *maximal ideal* if and only if $\mathsf{K}[\mathbf{x}]/\mathfrak{m}$ is a field, i.e. there is no further ideal between $\mathfrak{m}$ and $\mathsf{K}[\mathbf{x}]$.
We denote by $\mathrm{Max}(\mathsf{K}[\mathbf{x}])$ the set of all maximal ideals of $\mathsf{K}[\mathbf{x}]$ and call this the *maximal spectrum* of $\mathsf{K}[\mathbf{x}]$.

c. If $I \trianglelefteq K[\mathbf{x}]$ is an ideal, we denote the *radical* of I by

$$\sqrt{I} = \bigcap_{I \subseteq P \in \mathrm{Spec}(K[\mathbf{x}])} P = \{f \in K[\mathbf{x}] \mid \exists\, n \in \mathbb{N} \,:\, f^n \in I\}.$$

d. For any subset $X \subseteq \mathbb{A}^n_K$ we call the radical ideal

$$I(X) = \{f \in K[\mathbf{x}] \mid f(p) = 0 \;\forall\, p \in X\} \trianglelefteq K[\mathbf{x}]$$

the *vanishing ideal* of X.

Remark 2.25
Since $f^n(p) = 0$ if and only if $f(p) = 0$, for any ideal $I \trianglelefteq K[\mathbf{x}]$ we have

$$V(I) = V\big(\sqrt{I}\big).$$

Thus, every affine algebraic variety is the vanishing set of a radical ideal.
Moreover, it is obvious that

$$\sqrt{I} \subseteq I\big(V(I)\big).$$

Exercise 2.26
Find a maximal ideal $I \trianglelefteq \mathbb{R}[x]$ such that $\sqrt{I} \neq I\big(V(I)\big)$ and such that I is not of the form $\langle x - a\rangle$.

Something like this cannot happen for $K = \mathbb{C}$. There life is much better as the following theorem states.

Theorem 2.27 (Hilbert's Nullstellensatz)
Let K be an algebraically closed field.

a. *For any ideal $I \trianglelefteq K[\mathbf{x}]$*

$$I\big(V(I)\big) = \sqrt{I}.$$

Hence there is a one-to-one correspondence between the affine algebraic varieties and the radical ideals

$$\{X \subseteq \mathbb{A}^n_K \mid X \textit{ aff. alg. var.}\} \underset{I}{\overset{V}{\underset{\longleftarrow}{\overset{\longrightarrow}{1:1}}}} \{I \trianglelefteq K[\mathbf{x}] \mid I = \sqrt{I}\}$$

b. *The maximal ideals in $K[\mathbf{x}]$ are precisely the ideals of the form*

$$\mathfrak{m}_p = \langle x_1 - p_1, \ldots, x_n - p_n\rangle$$

for $p = (p_1, \ldots, p_n) \in \mathbb{A}^n_K$. Thus there is a one-to-one correspondence

$$\mathbb{A}^n_K \xrightarrow{1:1} \mathrm{Max}(K[\mathbf{x}]) : p = (p_1, \ldots, p_n) \mapsto \mathfrak{m}_p = \langle x_1 - p_1, \ldots, x_n - p_n\rangle.$$

Proof: One first shows that if $K[\mathbf{x}]/\mathfrak{m}$ is a field then $\dim_K(K[\mathbf{x}]/\mathfrak{m})$ is finite using results from commutative algebra on finite ring extensions and localisation. In particular $K[\mathbf{x}]/\mathfrak{m}$ is an algebraic extension of K, but if K is algebraically closed then $K[\mathbf{x}]/\mathfrak{m}$ must coincide with K, i.e. the map

$$K \longrightarrow K[\mathbf{x}]/\mathfrak{m} : a \mapsto \overline{a}$$

is an isomorphism. Thus, there is some $a_i \in K$ such that $x_i \equiv a_i (\mathrm{mod}\ \mathfrak{m})$, and hence $\mathfrak{m} = \langle x_1 - a_1, \ldots, x_n - a_n \rangle$. For part a. one then shows that

$$I(V(I)) = \bigcap_{p \in V(I)} I(V(p)) = \bigcap_{p \in V(I)} \mathfrak{m}_p = \sqrt{I},$$

i.e. the radical of I is already the intersection of all maximal ideals containing I. □

Example 2.28 (Radical)
We can compute the radical of an ideal with SINGULAR with the aid of the command `radical` from the library `primdec.lib`. This may help to see what the vanishing set is. E.g. in the following example the ideal I describes just the line $x = y = z$.

```
> ring r=0,(x,y,z),dp;
> ideal I=x2-2xy+2y2-2yz+z2,x2-2xy+2yz-z2;
> LIB "primdec.lib";
> radical(I);
_[1]=y-z
_[2]=x-z
```

E) **Decomposition into irreducible components**

Definition 2.29 (Irreducible varieties)
An affine algebraic variety X is *irreducible* if it cannot be decomposed as $X = Y \cup Z$ for two strictly smaller affine algebraic varieties Y and Z.

Remark 2.30
It is straight forward to see that for ideals $I, J \trianglelefteq K[\mathbf{x}]$ we have

$$V(I) \cup V(J) = V(I \cap J) = V(I \cdot J).$$

One can compute the intersection of two ideals in SINGULAR using the command `intersect`. In the following example we compute an ideal defining the union of a double cone and a line (see Figure 11).

```
> ring r=0,(x,y,z),dp;
> ideal I=x2+y2-z2;
> ideal J=x-1,z-2;
> intersect(I,J);
_[1]=x2z+y2z-z3-2x2-2y2+2z2
_[2]=x3+xy2-xz2-x2-y2+z2
```

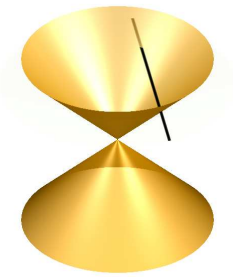


FIGURE 11. A double cone and a line.

Proposition 2.31 (Irreducible $\hat{=}$ prime)
A non-empty affine algebraic variety X *is irreducible if and only if* $I(X)$ *is prime.*

Idea of the proof: $X = X_1 \cup X_2 \iff I(X) = I(X_1) \cap I(X_2) \supseteq I(X_1) \cdot I(X_2)$. □

Proposition 2.32 (Lemma of Gauß + primary decomposition)

a. *The polynomial ring* $K[\mathbf{x}]$ *is factorial, i.e. every non-constant polynomial factorises uniquely as a product of irreducible polynomials.*

b. *Every radical ideal* $I = \sqrt{I}$ *in* $K[\mathbf{x}]$ *factorises uniquely as an intersection of finitely many prime ideals*

$$I = P_1 \cap \ldots \cap P_k \tag{2}$$

such that none of the P_i *is superfluous. We call the decomposition* (2) *the* minimal primary decomposition *of* I *and we call the prime ideals in*

$$\mathrm{Ass}(I) = \{P_1, \ldots, P_k\}$$

the associated prime ideals *of* I.

Corollary 2.33 (Irreducible decomposition)
Every affine algebraic variety X *decomposes uniquely as a union*

$$X = X_1 \cup \ldots \cup X_k$$

of irreducible affine algebraic varieties, non of which is superfluous. We call these the irreducible components *of* X.

Idea of the proof: Let $I(X) = P_1 \cap \ldots \cap P_k$ be the minimal primary decomposition of $I(X)$, then

$$X = V(I(X)) = V(P_1 \cap \ldots \cap P_k) = V(P_1) \cup \ldots \cup V(P_k),$$

and none of these irreducible varieties is superfluous since none of the P_i is so. □

Example 2.34 (Decomposition of a hypersurface)
If an affine algebraic variety X is defined by the vanishing of a single polynomial f, we call it a *hypersurface* in $\mathbb{A}^n_k$, moreover, if $n = 2$ we simply call X a *plane curve*, and if $n = 3$ we call X a *surface*. In that case the primary decomposition $I(X)$ can be computed by decomposing f into its irreducible factors, and the irreducible factors

thus define the irreducible components of X.
E.g. in the following example the curve $V(f)$ decomposes as a union of the three lines (see Figure 12).

```
> ring s=0,(x,y),dp;
> poly f=x^3-x*y^2;
> factorize(f);
[1]:
   _[1]=-1
   _[2]=-x+y
   _[3]=x+y
   _[4]=x
[2]:
   1,1,1,1
```

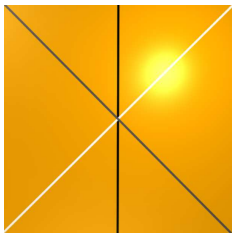


FIGURE 12. $V(x^3 - xy^2) = V(x) \cup V(x - y) \cup V(x + y)$.

Example 2.35 (Associated primes and irreducible decomposition)
We can compute the associated primes of an ideal via `minAssGTZ` and we can thus compute the irreducible components of an affine algebraic variety.

```
> ring r=0,(x,y,z),dp;
> LIB "primdec.lib";
> ideal I=xz,yz;
> minAssGTZ(I);
[1]:
   _[1]=z
[2]:
   _[1]=y
   _[2]=x
```

This shows that the affine algebraic variety defined by $xz = yz = 0$ decomposes as the union of the xy-plane $V(z)$ and the z-axis $V(x, y)$ (see Figure 13).

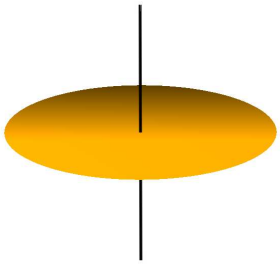


FIGURE 13. The affine algebraic variety $V(xz, yz) = V(z) \cup V(x, y)$.

F) **The coordinate ring and the dimension of a variety**

Remark 2.36
It is easy to see that affine algebraic varieties $X, Y \subseteq \mathbb{A}^n_K$ satisfy

$$X \subsetneqq Y \quad \Longleftrightarrow \quad I(X) \supsetneqq I(Y).$$

We then call X a *subvariety* of Y.

This motivates the following definition of the dimension of an affine algebraic variety.

Definition 2.37 (Dimension)
Let $X \subseteq \mathbb{A}^n_K$ be an affine algebraic variety and $I \trianglelefteq K[\mathbf{x}]$ an ideal.

a. The *dimension* of X is one less than the maximal length of a strictly descending chain of irreducible subvarieties of X, i.e.

$$\dim(X) = \max\{d \mid X \supseteq X_0 \supsetneqq X_1 \supsetneqq \ldots \supsetneqq X_d, X_i \text{ irred. aff. alg. var.}\}.$$

b. The *Krull dimension* $\dim(R)$ of a ring R is one less than the maximal length of a strictly ascending chain of prime ideals in R. Thus the *Krull dimension* of $K[\mathbf{x}]/I$ is

$$\dim(K[\mathbf{x}]/I) = \max\{d \mid I \subseteq P_0 \subsetneqq P_1 \subsetneqq \ldots \subsetneqq P_d, P_i \in \operatorname{Spec}(K[\mathbf{x}])\}.$$

c. We call the ring $K[X] = K[\mathbf{x}]/\,I(X)$ the *coordinate ring* of X.

Proposition 2.38 (Dimension)
If $X \subseteq \mathbb{A}^n_K$ is an affine algebraic variety and K is algebraically closed, then

$$\dim(X) = \dim\big(K[X]\big) = \max\{\dim(K[\mathbf{x}]/P) \mid P \in \operatorname{Ass}(I)\}.$$

The dimension of X is the maximum of the dimensions of its irreducible components.

Idea of the proof: Y is an irreducible subvariety of X if and only if $I(Y)$ is a prime ideal containing $I(X)$. Moreover, by Hilbert's Nullstellensatz two different prime ideals define different varieties. □

Theorem 2.39 (Krull's Principle Ideal Theorem)
Let K be an algebraically closed field.

a. $\dim \mathbb{A}^n_K = \dim K[\mathbf{x}] = n$.

b. *If* $f \in K[\mathbf{x}] \setminus K$, *then* $\dim V(f) = \dim K[\mathbf{x}]/\langle f\rangle = n-1$.

c. *If* $I = \langle f_1, \ldots, f_k\rangle$ *with* $f_i \in K[\mathbf{x}] \setminus K$, *then* $\dim V(I) \geq n-k$.

If the equality in c. holds, we call $V(I)$ *a* complete intersection *and we say that* $V(I)$ *has the* expected dimension.

Idea of the proof: One shows by induction on k that a prime ideal which is minimal such that it contains polynomials $f_1, \ldots, f_k$ contains no chain with more than $k+1$ prime ideals. The hard part is the case $k = 1$. But then Hilbert's Nullstellensatz shows that the dimension of $K[\mathbf{x}]$ is at most n since the maximal ideals are all generated by n polynomials, and there is also a chain with $n+1$ prime ideals

$$\langle 0\rangle \subsetneqq \langle x_1\rangle \subsetneqq \langle x_1, x_2\rangle \subsetneqq \ldots \subsetneqq \langle x_1, x_2, \ldots, x_n\rangle.$$

□

Example 2.40
The dimension of $X = V(xz, yz) = V(z) \cup V(x, y)$ is

$$\begin{aligned}\dim(X) &= \max\{\dim V(z), \dim V(x,y)\}\\ &= \max\{\dim K[x,y,z]/\langle z\rangle, \dim K[x,y,z]/\langle x,y\rangle = \max\{2,1\} = 2.\end{aligned}$$

Remark 2.41 (How to compute the dimension.)
The dimension of $K[\mathbf{x}]/I$ can easily be computed as "dim(I)" with SINGULAR using the command `dim`. The reason for that is that for any global monomial ordering $>$

$$\dim\big(K[\mathbf{x}]/I\big) = \dim\big(K[\mathbf{x}]/L_>(I)\big),$$

and since the latter is a monomial ideal one only has to check how many variables at most are algebraically independent modulo the monomial generators. That is rather simple. Note, that SINGULAR simply computes the leading terms of the generators and proceeds with these. Thus the ideal I should already be given by a Gröbner basis.
E.g. in the following example the ideal I defines a curve in space, for which three equations are needed. It is an example of an irreducible affine algebraic variety which is not a complete intersection.

```
> ring r=0,(x,y,z),dp;
> ideal I=y2-xz,x2y-z2,x3-yz;
> dim(groebner(I));
1
> ideal J=lead(groebner(I));
> J;
J[1]=y2
J[2]=x2y
J[3]=x3
```

```
> dim(groebner(J));
1
> LIB "primdec.lib";
> size(minAssGTZ(I));
1
```

By the last command we show that I has only *one* associated prime ideal, which shows that $V(I)$ is irreducible. One can indeed easily compute that I is prime. (See Figure 14.)

FIGURE 14. A space curve drawn on a shadow of the surface $V(y^2 - xz)$.

G) **Exercises**

Exercise 2.42
Let $M, N \subseteq K[\boldsymbol{x}]$.

a. Show, if $M \subseteq N$, then $V(M) \supseteq V(N)$.

b. Show that $V(M) = V(\langle M \rangle)$ for $M \subseteq K[\boldsymbol{x}]$.

Exercise 2.43
Show that the monomial orderings in Example 2.9 are global monomial orderings.

Exercise 2.44 (Global monomial orderings)
Show that for a total ordering $>$ on $\mathrm{Mon}(\boldsymbol{x})$ which is compatible with the semigroup structure on $\mathrm{Mon}(\boldsymbol{x})$ the following are equivalent:

a. $>$ is a monomial ordering.

b. $1 < x_i$ for $i = 1, \ldots, x_n$.

c. If $\alpha_i \leq \beta_i$ for $i = 1, \ldots, n$, then $\boldsymbol{x}^\alpha \leq \boldsymbol{x}^\beta$.

d. $>$ is a well-ordering (i.e. every non-empty set of monomials contains a minimal element).

Exercise 2.45
Show that the monomial orderings in Example 4.9 are local monomial orderings.

Exercise 2.46
Find an example of a monomial ordering which is neither local nor global.

Exercise 2.47
Compute the leading monomial and the leading coefficient of the polynomial $f = x_1^3x_2 - 4x_1x_2^5 + x_1^2 - x_2^2$ with respect to $>_{lp}$, $>_{dp}$, $>_{ls}$, and $>_{ds}$ by hand and using SINGULAR. (For the definition of $>_{ls}$ and $>_{ds}$ see Example 4.9).

Exercise 2.48
Compute a standard representation of the polynomial $f = x^2yz^2 + 3z^3$ with respect to $g_1 = x^2y + y$, $g_2 = z^2 - x$ and the ordering $>_{dp}$ using SINGULAR.

Exercise 2.49
Compute the leading ideal of $I = \langle x^3y + 2x, x^2yz + 3xy, z^3 - x^2\rangle$ with respect to $>_{dp}$ with SINGULAR.

Exercise 2.50
Check with SINGULAR if the polynomial $x^3y^4 - 3yz^2 - x^2z$ is contained in the ideal $I = \langle x^3y + 2x, x^2yz + 3xy, z^3 - x^2\rangle$.

Exercise 2.51
Show that the set of solutions of

$$x^2 + y^2 - 8 = x^2 - y^2 = 0$$

is finite and compute the solutions with the aid of `minAssGTZ` as well as with the aid of `solve`. Moreover, find a non-zero polynomial in $I \cap K[x]$ for $I = \langle x^2+y^2-8, x^2-y^2\rangle$.

Exercise 2.52
Consider the three plane curves C_i in $\mathbb{A}^2_{\mathbb{C}}$ given by the equations $f_i = 0$, $i = 1, 2, 3$, where

$$f_1 = y^2 - 5x^2 - x^3, \quad f_2 = x^4 + y^4 - 2, \quad \text{respectively} \quad f_3 = y^2 + 5x^2 + x^3.$$

How many intersection points do C_1 and C_2 respectively C_1 and C_3 have? How many of these points are real? You may use SINGULAR for the computations. Verify the real points by drawing the curves using `surf`.

Exercise 2.53
Factorize the polynomial $f = -x^2y^4z - xy^5z - xy^4z^2 + x^4yz + x^3y^2z + x^3yz^2 + 2x^2y^3 + 2xy^4 + 2xy^3z - 2x^4 - 2x^3y - 2x^3z$ using Singular.

Exercise 2.54
Show that $V(I)$ is finite if and only if $\dim_K K[\mathbf{x}]/I$ is finite.

Exercise 2.55
Find a maximal ideal $I \trianglelefteq \mathbb{R}[x]$ such that $\sqrt{I} \neq I\big(V(I)\big)$ and such that I is not of the form $\langle x - a\rangle$.

Exercise 2.56
Compute the vanishing ideal of intersection of $V(x^2 - y^3)$ and $V(x - y^2)$ in $\mathbb{C}[x, y]$ using SINGULAR.

Exercise 2.57
Prove that $V(I \cdot J) = V(I \cap J) = V(I) \cup V(J)$ for ideals $I, J \trianglelefteq K[\mathbf{x}]$.

Exercise 2.58
Show that $I(X) = \sqrt{I(X)}$ for any $X \subseteq \mathbb{A}^n_K$.

Exercise 2.59
Let X be the union of the three coordinate axes in $\mathbb{A}^3_{\mathbb{C}}$. Compute the vanishing ideal of X in $K[x, y, z]$. Is X a complete intersection?

Exercise 2.60
Consider the surface $V(f) \subset \mathbb{A}^3_{\mathbb{C}}$ defined by the polynomials $f = x^2 + y^2 + xyz$ and consider the planes $H_c = V(x - c) \subset \mathbb{A}^3_{\mathbb{C}}$ for $c \in \mathbb{C}$ arbitrary. Determine $I(X_c)$ for $X_c = H_c \cap V(f)$.

Exercise 2.61
Compute the irreducible components of $V(x^3 + x^2y + x^2z - xyz - y^2z - yz^2)$ and visualise them with `surfex`.

Exercise 2.62
Compute first the vanishing ideal of $X = V(x^2 - z, xy - z^2) \cap V(x + y + z)$ and compute then its irreducible components. Visualise the irreducible components with `surfex` and compute their dimension. What is the dimension of X?

Exercise 2.63
Consider the ideal I generated by the 2×2-minors of the matrix

$$A = \begin{pmatrix} x_0 & x_1 & x_2 & x_3 & x_4 \\ x_5 & x_6 & x_7 & x_8 & x_9 \end{pmatrix}$$

in $\mathbb{C}[x_0, \ldots, x_9]$ and the variety $X = V(I)$. Compute the dimension of X. You may use the SINGULAR command `minor`.

Exercise 2.64
Let $X = V\big(yz^2 - yz, xz^2 - xz, xyz + xz - y^2z - yz, y^3z - yz\big) \subseteq \mathbb{A}^3_{\mathbb{C}}$. Compute the irreducible components of the variety X as well as the dimension of X and of each of its irreducible components.

3 Regular functions and morphisms

Affine algebraic varieties carry more structure than mere sets. They are topological spaces with additional structure. In this section we first introduce the *Zariski topology* on affine algebraic varieties, and then we want to study morphisms between affine algebraic varieties, i.e. maps which respect their structure. For that we introduce first the notion of *regular functions*, i.e. admissible maps on open subsets of affine algebraic varieties which take values in the base field K. They will locally be given as rational functions. The regular functions are the additional structure, and they are gathered in the so called *structure sheaf*. A *morphism* then should respect these regular functions via composition. One of the main results will be that only polynomial maps do so.

A) The Zariski topology

Definition 3.1 (Topology)
Let X be a set.

a. A *topology* on X is a set $\mathcal{T}$ of subsets of X which is closed with respect to finite unions and arbitrary intersections and which contains X and $\emptyset$. We call the elements of $\mathcal{T}$ the *closed subsets* of X, and we call X together with $\mathcal{T}$ a *topological space*.

b. A subset of a topological space is *open* if its complement is closed.

c. A map between topological spaces is *continuous* if the preimage of closed subsets is closed or, equivalently, if the preimage of open subsets is open.

Proposition 3.2 (Zariski topology on $\mathbb{A}^n_K$)
The collection of all affine algebraic varieties in $\mathbb{A}^n_K$ *is the* Zariski *topology on* $\mathbb{A}^n_K$.

Idea of the proof:

- $\mathbb{A}^n_K = V(0)$ and $\emptyset = V(1)$ are closed.
- A finite union of closed subsets $\bigcup_{i=1}^k V(I_i) = V\left(\bigcap_{i=1}^k I_i\right)$ is closed.
- An arbitrary intersection of closed subsets $\bigcap_{\lambda\in\Lambda} V(I_\lambda) = V\left(\bigcup_{\lambda\in\Lambda} I_\lambda\right)$ is closed.

□

Example 3.3 (Zariski topology on $\mathbb{A}^1_K$)
In $\mathbb{A}^1_K$ only finite sets and $\mathbb{A}^1_K$ are closed. The Zariski topology is a rather coarse topology.

Remark 3.4 (Zariski topology on X)
Every subvariety $X \subseteq \mathbb{A}^n_K$ of the affine n-space is a topological space with respect to the subspace topology. That is, a subset of X is closed if and only if it is an affine algebraic variety which is contained in X.

Exercise 3.5
If $f \in K[\mathbf{x}]$ we call $X_f = X \setminus V(f)$ a *basic* open subset of X. Show that every open subset of X is a union of finitely many basic open subsets. One says that the basic open subsets form a basis of the Zariski topology.

Exercise 3.6
Let X be an irreducible affine algebraic variety and $U \subseteq X$ a non-empty open subset of X. Then U is dense in X, i.e. its topological closure is all of X.

We will consider all algebraic varieties as topological spaces w.r.t. the Zariski topology. We write $K = \overline{K}$ to indicate that K is algebraically closed.

B) **Regular functions**

Definition 3.7 (Regular functions)
Let X be an affine algebraic variety in $\mathbb{A}^n_K$ and $U \subseteq X$ be open. A function $f : U \longrightarrow K$ is called *regular* if it locally is a quotient of two polynomials, i.e.

$$\forall\, p \in U\ \exists\, p \in V \subseteq U \text{ open and } g, h \in K[\mathbf{x}] \text{ s.t. } \forall\, q \in V\ :\ f(q) = \frac{g(q)}{h(q)}.$$

By $\mathcal{O}_X(U)$ we denote all regular functions on U, and we call the functions in $\mathcal{O}_X(X)$ *global* regular functions. Note that with the usual operations $\mathcal{O}_X(U)$ is a K-algebra.

Exercise 3.8
If $X = V(x_1x_2 - x_3x_4) \subset \mathbb{A}^4_{\mathbb{C}}$ and $U = X \setminus V(x_1, x_3)$ then the function

$$f : U \longrightarrow \mathbb{C} : (x_1, x_2, x_3, x_4) \mapsto \begin{cases} \frac{x_2}{x_3}, \text{ if } x_3 \neq 0, \\ \frac{x_4}{x_1}, \text{ if } x_1 \neq 0 \end{cases}$$

is well-defined and regular. However, it is impossible to write f as a quotient of two polynomials on the whole of U!

Example 3.9 (Elements in the coordinate ring as regular functions)
Every polynomial $f \in K[\mathbf{x}]$ defines a regular function

$$f : X \mapsto K : p \mapsto f(p)$$

on an affine algebraic variety X, and two polynomials f and g define the same function on X if their difference is in $I(X)$. One may thus in a natural way identify the elements of the coordinate ring of X with regular functions on X.

It turns out that indeed there are no other regular functions on all of X.

Theorem 3.10 (Global regular functions)
Let X be an affine algebraic variety and $K = \overline{K}$, then $\mathcal{O}_X(X) \cong K[X]$ as K-algebras. In particular, every global regular function is globally defined by a polynomial.

Idea of the proof: Considering the elements in $K[X]$ as regular functions gives an injective K-algebra homomorphism from $K[X]$ to $\mathcal{O}_X(X)$. It remains to show that every regular function on X is globally given by a polynomial. For that we note that for any given regular function $f : X \longrightarrow K$ there is a finite covering of X by basic open subsets $X_{h_1}, \ldots, X_{h_k}$ such that f coincides on X_{h_i} with a quotient $\frac{g_i}{h_i}$. But then $\langle h_1, \ldots, h_k \rangle + I(X) = K[\boldsymbol{x}]$ and thus

$$1 \equiv \sum_{i=1}^{k} f_i \cdot h_i \pmod{I(X)},$$

and since $g_i \cdot h_j \equiv g_j \cdot h_i$ we get

$$g_j \equiv \sum_{i=1}^{k} f_i \cdot h_i \cdot g_j \equiv \sum_{i=1}^{k} f_i \cdot h_j \cdot g_i \equiv h_j \cdot g$$

for $g = \sum_{i=1}^{k} f_i \cdot g_i \in K[\boldsymbol{x}]$. This shows that the regular function $f \equiv \frac{g_j}{h_j} \equiv g$ coincides with g on each X_{h_i} and hence on X. □

This result generalises to basic open sets.

Remark 3.11 (Localisation at f)
Let $X \subseteq \mathbb{A}^n_K$ be an affine algebraic variety and $0 \neq f \in K[X]$. We call the K-algebra

$$K[X]_f = \left\{ \frac{g}{f^m} \;\middle|\; g \in K[X], m \in \mathbb{N} \right\}$$

the *localisation of* $K[X]$ *at* f.

Theorem 3.12 (Regular functions on basic open sets)
Let X *be an affine algebraic variety,* $K = \overline{K}$ *and* $f \in K[\boldsymbol{x}] \setminus I(X)$*, then* $\mathcal{O}_X(X_f) \cong K[X]_f$*. In particular, every regular function on a basic open set is globally a quotient of two polynomials.*

Proof: This is proved as Theorem 3.10 with X replaced by X_f. □

Remark 3.13 (The structure sheaf $\mathcal{O}_X$)
For every open subset U of an affine algebraic variety X we have the K-algebra $\mathcal{O}_X(U)$ of regular functions on U, and for two open subsets $V \subseteq U \subseteq X$ of X the restriction map

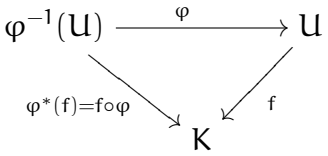
$$\mathrm{res}_{U,V} : \mathcal{O}_X(U) \longrightarrow \mathcal{O}_X(V) : f \mapsto f_{|V}$$

which restricts a regular function on U to the subset V is a K-algebra homomorphism. Moreover, the restriction maps behave nicely, i.e. $\mathrm{res}_{U,U} = \mathrm{id}_{\mathcal{O}_X(U)}$ and $\mathrm{res}_{V,W} \circ \mathrm{res}_{U,V} = \mathrm{res}_{U,W}$. This all amounts to the fact that the collection of K-algebras $\mathcal{O}_X(U)$ and restriction maps $\mathrm{res}_{U,V}$, where U and V run over all open subsets of X such that $V \subseteq U$, forms a *sheaf of* K*-algebras*. We call it the *structure sheaf* of X and denote it by $\mathcal{O}_X$.

In modern algebraic geometry the language of sheaves is vital. In this minicourse, however, we will avoid it since it is rather technical and not that important for computational questions.

C) **Morphisms**

Definition 3.14 (Morphisms)
Let $X \subseteq \mathbb{A}^n_K$ and $Y \subseteq \mathbb{A}^m_K$ be two affine algebraic varieties. A *morphism* from X to Y is a continuous map $\varphi : X \longrightarrow Y$ such that regular functions pull back to regular functions, i.e. for any open subset $U \subseteq Y$ and $f \in \mathcal{O}_Y(U)$ the function $\varphi^*(f) = f \circ \varphi \in \mathcal{O}_X\big(\varphi^{-1}(U)\big)$:



We denote by $\mathrm{Mor}(X, Y)$ the set of morphisms from X to Y. A morphism is an *isomorphism* if it is bijective and its inverse in a morphism as well.

Example 3.15 (Polynomial maps are morphisms.)
If $f_1, \ldots, f_m \in K[X]$ then we get a morphism

$$X \longrightarrow \mathbb{A}^m_K : p \mapsto \big(f_1(p), \ldots, f_m(p)\big)$$

by just taking the f_i as component functions. This works since composing a rational function with a polynomial gives a rational function.

The following theorem states that actually this is the only way to get a morphism.

Theorem 3.16 (Morphisms are polynomial maps.)
If $X \subseteq \mathbb{A}^n_K$ and $Y \subseteq \mathbb{A}^m_K$ are affine algebraic varieties and $K = \overline{K}$, then the pull-back

$$\mathrm{Mor}(X, Y) \longrightarrow \mathrm{Hom}_{K-alg}(K[Y], K[X]) : \varphi \mapsto \varphi^*$$

is a bijection.
In particular, if $\varphi : X \to Y$ is a morphism, then the components of φ are polynomials.

Idea of the proof: The inverse map is given by assigning to a K-algebra homomorphism $\psi : K[Y] \longrightarrow K[X]$ the morphism $X \longrightarrow Y$ with component functions $\psi(y_1), \ldots, \psi(y_m) \in K[X]$. □

Corollary 3.17 (Isomorphisms)
Let $\varphi : X \longrightarrow Y$ be a morphism of affine algebraic varieties over $K = \overline{K}$ and let $\varphi^* : K[Y] \longrightarrow K[X]$ be its pull back. Then φ is an isomorphism of varieties if and only if φ^* is an isomorphism of K-algebras.

Exercise 3.18 (Bijective morphisms need not be isomorphisms.)
Show that the morphism $\varphi : \mathbb{A}^1_{\mathbb{C}} \longrightarrow \mathbb{A}^2_{\mathbb{C}} : t \mapsto (t^2, t^3)$ is bijective onto its image $Y = V(x^3 - y^2)$, but it is not an isomorphism, since the pull back

$$\varphi^* : \mathbb{C}[Y] = \mathbb{C}[x, y]/\langle x^3 - y^2\rangle \longrightarrow \mathbb{C}[t] : x \mapsto t^2, y \mapsto t^3$$

is not surjective — t is not in its image.

Remark 3.19 (Morphisms on open subsets of X)
If one replaces in Theorem 3.16 $K[X]$ and $K[Y]$ by $\mathcal{O}_X(X)$ and $\mathcal{O}_Y(Y)$ respectively, then the assumption $K = \overline{K}$ can be dropped.
One can of course define morphisms on open subsets of X in the same way, and if one then replaces in Theorem 3.16 X by some open subset U of X and $K[X]$ by $\mathcal{O}_X(U)$ the statement holds still true. However, this is not the case for Corollary 3.17. For $X = \mathbb{A}^2_{\mathbb{C}}$ and $U = X \setminus \{(0,0)\}$ one can show that $\mathcal{O}_X(U) = K[x,y]$ and the inclusion $i : U \hookrightarrow X$ induces the isomorphism

$$i^* = \mathrm{id} : K[Y] = K[x,y] \longrightarrow K[x,y] = \mathcal{O}_X(U)$$

without being an isomorphism itself.

Remark 3.20 (Morphisms in SINGULAR)
One can define a ring homomorphism between two polynomial rings in SINGULAR and then map polynomials or ideals with this homomorphism from the previous ring to the new ring. E.g. $R = K[x,y,z]$ and $S = K[s,t]$ with

$$\varphi : R \longrightarrow S : x \mapsto s^2, y \mapsto st, z \mapsto t^2$$

and we would like to compute the image of $x^2+y^2-z^2$. We have to define a variable of type `map` by specifying the domain of definition and the polynomials to which the coordinates in the domain of definition should be mapped.

```
> ring R=0,(x,y,z),dp;
> poly f=x^2+y^2-z^2;
> ring S=0,(s,t),dp;
> map phi=R,s2,st,t2;
> phi(f);
s4+s2t2-t4
```

There are also predefined maps which can be accessed by the commands `imap` and `fetch`. `imap` works like the identity and does not change anything, `fetch` simply maps the i-th variable of the domain of definition to the i-th variable of the new ring. For more details on how to use these one should consult the manual.

D) **Computing the image of a morphism**

Example 3.21 (Projections)
The simplest type of a morphism is a projection which simply forgets some components. If $n \geq m$ then we get the projection

$$\mathrm{pr}_{n,m} : \mathbb{A}^n_K \longrightarrow \mathbb{A}^m_K : (p_1, \ldots, p_n) \mapsto (p_1, \ldots, p_m).$$

Unfortunately, the image of an affine algebraic variety under a projection need no longer be an affine algebraic variety. E.g. the image of $V(xy-1) \subset \mathbb{A}^2_K$ under the

projection to the x-axis is $\mathbb{A}^1_K \setminus \{0\}$. However, it is never far from being a variety and it is safe to compute its *topological closure*, i.e. the smallest affine algebraic variety in which it is contained. This can be done as follows.

If $X = V(I) \subseteq \mathbb{A}^n_K$ is an affine algebraic variety then the closure of $\pi_{n,m}(X)$ in $\mathbb{A}^m_K$ is

$$\overline{\pi_{n,m}(X)} = V\big(I \cap K[x_1, \ldots, x_m]\big).$$

That is, we can compute the closure of the image of an affine algebraic variety under a projection by intersecting a defining ideal with a subalgebra. This can be done by computing a Gröbner basis similar to Remark 2.21 using the concept of block orderings. In SINGULAR one can instead use the built-in command `eliminate` which computes the intersection of an ideal with a subalgebra by *eliminating* the variables one wants to get rid of.

```
> ring r=0,(x,y,z),dp;
> ideal I=y2-xz,x2y-z2,x3-yz;
> eliminate(I,x);
_[1]=y5-z4
```

The example shows that if we project the space curve $V(I)$ to the yz-plane we get a curve with the equation $y^5 - z^4 = 0$ (see Figure 15).

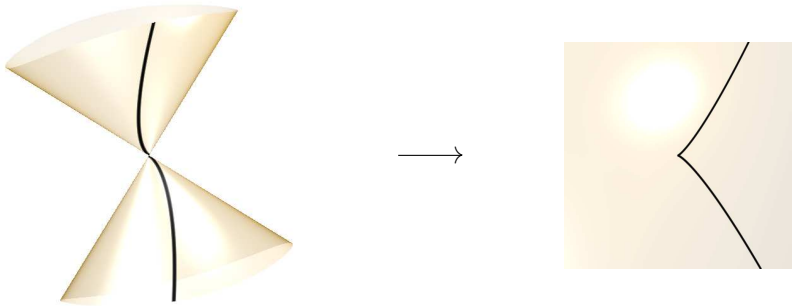


FIGURE 15. A projection of a space curve

Remark 3.22 (Computing the image of a morphism by projecting its graph.) If $\varphi : X \longrightarrow \mathbb{A}^m_K$ with $X \subseteq \mathbb{A}^n_K$ is any morphism, then its *graph* is the set

$$\mathrm{Graph}(\varphi) = \big\{\big(p, \varphi(p)\big) \in \mathbb{A}^{n+m}_K \;\big|\; p \in X\big\}.$$

It actually is an affine algebraic variety, namely the one defined by the ideal

$$\langle f_1, \ldots, f_k, y_1 - g_1, \ldots, y_m - g_m\rangle \trianglelefteq K[x_1, \ldots, x_n, y_1, \ldots, y_m]$$

if $X = V(f_1, \ldots, f_k)$ and $\varphi = (g_1, \ldots, g_m)$.

Moreover, it is clear that

$$\varphi(X) = \pi_{n,m}\big(\operatorname{Graph}(\varphi)\big),$$

and we can thus compute the closure of the image of φ as the projection of the graph of φ.

E.g. consider the morphism

$$\varphi : \mathbb{A}^1_{\mathbb{R}} \longrightarrow \mathbb{A}^2_{\mathbb{R}} : t \mapsto \left(\frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2}\right)$$

and its graph

$$\operatorname{Graph}(\varphi) = \big\{\big(t, \varphi(t)\big) \mid t \in \mathbb{R}\big\} = \left\{\left(t, \frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2}\right) \;\middle|\; t \in \mathbb{R}\right\}.$$

The graph of φ is a curve in space which lies on the surface of a cylinder. The coordinates in space are given by t, x and y, and the axis of the cylinder is the t-axis. If we project the curve into the xy-plane we get a circle. This is the image of the morphism. The morphism, its graph and the projection are displayed in Figure 16.

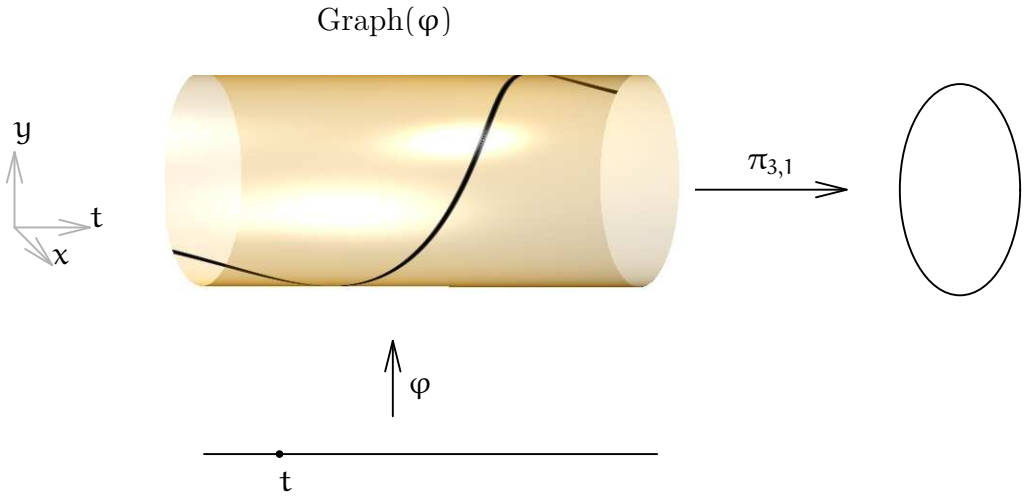


FIGURE 16. The black curve on the cylinder is the graph of φ

Example 3.23
If we consider the plane curve $X = V(y^2 - x^2 - x^3)$, the so called *Newton node*, and the map $\varphi : \mathbb{A}^2_k \longrightarrow \mathbb{A}^2_k : (x, y) \mapsto (x - 1, x + y)$, then we can compute $\varphi(X)$:

```
> ring r=0,(x,y,X,Y),dp;
> ideal I=y2-x2-x3,X-x+1,Y-x-y;
> eliminate(I,xy);
_[1]=X3+3X2+2XY-Y2+3X+2Y+1
```

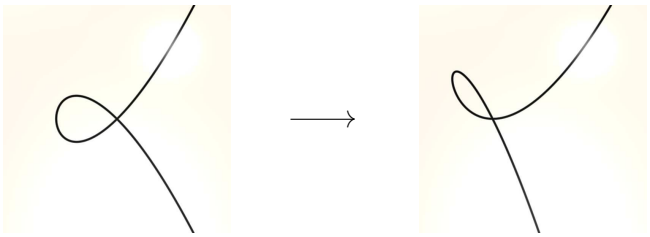


FIGURE 17. The image of the Newton node

Remark 3.24 (Parametrisations)
A particularly interesting case is the image of morphism

$$\varphi : \mathbb{A}^n_K \longrightarrow \mathbb{A}^m_K$$

on the whole of $\mathbb{A}^n_K$ when the morphism is nearly everywhere injective and its image is closed. We then say that φ *parametrises* its image.
E.g. the morphism

$$\varphi : \mathbb{A}^1_K \longrightarrow \mathbb{A}^3_K : t \mapsto \left(t^3, t^4, t^5\right)$$

is a parametrisation of the space curve considered in Remark 2.41 as the following computation basically shows:

```
> ring R=0,(x,y,z,t),dp;
> ideal I=x-t3,y-t4,z-t5;
> eliminate(I,t);
_[1]=y2-xz
_[2]=x2y-z2
_[3]=x3-yz
```

E) **Exercises**

Exercise 3.25
If $f \in K[\mathbf{x}]$ we call $X_f = X \setminus V(f)$ a *basic* open subset of X. Show that every open subset of X is a union of finitely many basic open subsets. One says that the basic open subsets form a basis of the Zariski topology.

Exercise 3.26
Let X be an irreducible affine algebraic variety and $U \subseteq X$ a non-empty open subset of X. Then U is dense in X, i.e. its topological closure is all of X.

Exercise 3.27
Show that any two non-empty open subsets of $\mathbb{A}^n_K$ have a non-empty intersection.

Exercise 3.28
If $X = V(x_1x_2 - x_3x_4) \subset \mathbb{A}^4_{\mathbb{C}}$ and $U = X \setminus V(x_1, x_3)$ then the function

$$f : U \longrightarrow \mathbb{C} : (x_1, x_2, x_3, x_4) \mapsto \begin{cases} \frac{x_2}{x_3}, \text{ if } x_3 \neq 0, \\ \frac{x_4}{x_1}, \text{ if } x_1 \neq 0 \end{cases}$$

is well-defined and regular. However, it is impossible to write f as a quotient of two polynomials on the whole of U!

Exercise 3.29
Let $U = \mathbb{A}^2_{\mathbb{C}} \setminus V(y^3 - y^2, y^2 + y - 2)$. Are all regular functions on U globally given by rational functions? If so, explain why, if not, give a counter example.

Exercise 3.30
Show that a regular function is continuous w.r.t. the Zariski topology.

Exercise 3.31
Let $X \subseteq \mathbb{A}^n_K$ be an affine algebraic variety with irreducible decomposition $X = X_1 \cup \ldots \cup X_k$ and let $f \in K[\mathbf{x}]$. Show that the residue class of f is a zero-divisor in $K[X]$ if and only if f vanishes identically on some X_i.

Exercise 3.32
Let K be algebraically closed and $U = \mathbb{A}^2_K \setminus \{0\}$. Show that $\mathcal{O}_X(U) = K[x, y]$, i.e. each regular function on U extends to a regular function on all of $\mathbb{A}^2_K$.

Exercise 3.33 (Bijective morphisms need not be isomorphisms.)
Show that the morphism $\varphi : \mathbb{A}^1_{\mathbb{C}} \longrightarrow \mathbb{A}^2_{\mathbb{C}} : t \mapsto (t^2, t^3)$ is bijective onto its image $Y = V(x^3 - y^2)$, but it is not an isomorphism, since the pull back

$$\varphi^* : \mathbb{C}[Y] = \mathbb{C}[x, y]/\langle x^3 - y^2 \rangle \longrightarrow \mathbb{C}[t] : x \mapsto t^2, y \mapsto t^3$$

is not surjective — t is not in its image.

Exercise 3.34
Let K be algebraically closed. Show that $V(y - x^2) \subseteq \mathbb{A}^2_K$ is isomorphic to $\mathbb{A}^1_K$.

Exercise 3.35
Which of the following algebraic sets are isomorphic?

$\mathbb{A}^1_{\mathbb{C}}$ $\quad$ $V(xy) \subseteq \mathbb{A}^2_{\mathbb{C}}$ $\quad$ $V(x^2 + y^2) \subseteq \mathbb{A}^2_{\mathbb{C}}$

$V(x^2 - y^3) \subseteq \mathbb{A}^2_{\mathbb{C}}$ $\quad$ $V(y - x^2, z - x^3) \subseteq \mathbb{A}^3_{\mathbb{C}}$.

Exercise 3.36
Define in SINGULAR a the map

$$\varphi : \mathbb{Q}[x, y] \longrightarrow \mathbb{Q}[a, b, c] : x \mapsto a^2 - bc, y \mapsto abc - 2$$

and compute the image of $f = x^2 - y^2$.

Exercise 3.37
Compute the image of the morphism $\mathbb{A}^2_K \longrightarrow \mathbb{A}^3_K : (s, t) \mapsto (s, t, s^2 + t^2)$ and visualise it with `surfex`.

Exercise 3.38
Compute the image of the morphism $\mathbb{A}^2_K \longrightarrow \mathbb{A}^3_K : (s,t) \mapsto (t^2 - st, s^2 - st, t^2 - s^2)$ and visualise it with `surfex`.

Exercise 3.39
Consider the variety X from exercise 2.63 and compute the vanishing ideal of its projection under $\pi_{10,7}$.

Exercise 3.40
Let $g_1, \ldots, g_m \in K[\mathbf{x}]$. Show that the topological closure of the image of

$$\varphi : \mathbb{A}^n_K \longrightarrow \mathbb{A}^m_K : p \mapsto (g_1(p), \ldots, g_k(p))$$

is the vanishing set of

$$\langle y_1 - g_1, \ldots, y_m - g_m \rangle \cap K[y_1, \ldots, y_m].$$

Exercise 3.41
Find a parametrisation of the Newton node $V(y^2 - x^2 - x^3)$.

Exercise 3.42
Compute the vanishing ideal of the image of the map

$$\varphi : \mathbb{A}^1_{\mathbb{R}} \longrightarrow \mathbb{A}^2_{\mathbb{R}} : t \mapsto \left(\frac{t^3 + 1}{t^4 + 1}, \frac{t^4 + t}{t^4 + 1} \right).$$

How can we deal with the denominator $t^4 + 1$? Visualise the resulting plane curve with `surfex`.

4 Local properties of algebraic varieties

In this section we want to consider the behaviour of an algebraic variety locally at some point. This leads naturally to the notion of the *tangent space* at a point.

A) The local ring of X at p

Remark 4.1 (Germs of regular functions)
Let X be an affine algebraic variety and $p \in X$. We call two regular functions $f : U \longrightarrow K$ and $g : V \longrightarrow K$ which are defined in two open neighbourhoods of p *equivalent* if they coincide on some possibly smaller open neighbourhood of p. This defines an equivalence relation on the set of all regular functions which are defined on some open neighbourhood of p.
The equivalence classes are called *germs of regular functions at* p, and they are represented by some regular function which is defined on an arbitrarily small neighbourhood of p.
The set of germs of regular functions at p is denoted by $\mathcal{O}_{X,p}$. If we define operations on $\mathcal{O}_{X,p}$ via representatives then it is straight forward to see that $\mathcal{O}_{X,p}$ is a local K-algebra and its unique maximal ideal consists of those germs whose representatives vanish at the point p. We call $\mathcal{O}_{X,p}$ the *local ring of* X *at* p.

Remark 4.2 (Localisation at p)
Let $X \subseteq \mathbb{A}^n_K$ be an affine algebraic variety and $p \in X$. We call the local K-algebra

$$K[X]_{\mathfrak{m}_p} = \left\{ \frac{g}{h} \;\middle|\; g, h \in K[X], h(p) \neq 0 \right\}$$

the *localisation of* $K[X]$ *at* $\mathfrak{m}_p = \langle x_1 - p_1, \ldots, x_n - p_n \rangle$ or at p. By abuse of notation we call the unique maximal ideal in $K[X]_{\mathfrak{m}_p}$ again $\mathfrak{m}_p$. It is generated by the $x_i - p_i$.

The next theorem shows that the elements in $K[X]_{\mathfrak{m}_p}$ are precisely the germs of regular functions at p.

Theorem 4.3 (The local ring of X at p)
Let X *be an affine algebraic variety over* $K = \overline{K}$ *and* $p \in X$*, then* $\mathcal{O}_{X,p} \cong K[X]_{\mathfrak{m}_p}$.

Idea of the proof: We consider the K-algebra homomorphism

$$K[X]_{\mathfrak{m}_p} \longrightarrow \mathcal{O}_{X,p} : \frac{f}{g} \mapsto \frac{f}{g}$$

which assigns to a rational function its germ at the point p. If $\frac{f}{g}$ is the zero germ, then f vanishes in some open neighbourhood of p, but since f is continuous and any open neighbourhood of p intersects all irreducible components of X, which contain p, in a dense set it follows that f vanishes identically on each irreducible component which passes through p. Thus $\frac{f}{g}$ is zero in $K[X]_{\mathfrak{m}_p}$ and the homomorphism is injective. Moreover, it is obviously surjective since every regular function locally in p is a rational function. $\square$

Example 4.4
The local ring of affine n-space at the origin is

$$\mathcal{O}_{\mathbb{A}^n_K,0} \cong K[\mathbf{x}]_{\langle x_1,\ldots,x_n\rangle} = \left\{ \frac{f}{g} \;\middle|\; f,g \in K[\mathbf{x}], g(0) \neq 0 \right\}.$$

Definition 4.5
Let X be an affine algebraic variety with irreducible components $X_1,\ldots,X_k$ and let $p \in X$. We define the *dimension of* X *locally at* p as

$$\dim(X,p) = \max\{\dim(X_i) \mid p \in X_i\}$$

the maximal dimension of an irreducible component of X containing p.

Proposition 4.6 (The dimension locally at a point)
If X *is an affine algebraic variety over* $K = \overline{K}$ *and* $p \in X$*, then*

$$\dim(X,p) = \dim \mathcal{O}_{X,p} = \dim K[X]_{\mathfrak{m}_p},$$

the dimension of X *locally at* p *is the Krull dimension of the local ring of* X *at* p*.*

Idea of the proof: When localising at p all components which do not pass through p are lost. □

Example 4.7
Consider $X = V(xz,yz) = V(z) \cup V(x,y)$ and $p = (0,0,1) \in V(x,y)$. Since only the component $V(x,y)$ contains p the dimension of X locally at p is 1. Moreover,

$$K[X]_{\mathfrak{m}_p} \cong K[x,y,z]_{\langle x,y,z-1\rangle}/\langle xz,yz\rangle = K[x,y,z]_{\langle x,y,z-1\rangle}/\langle x,y\rangle \cong K[z]_{\langle z-1\rangle}$$

since after localising at $\langle x,y,z-1\rangle$ the element z becomes a unit. The ring on the right hand side has also dimension one. (See Figure 18.)

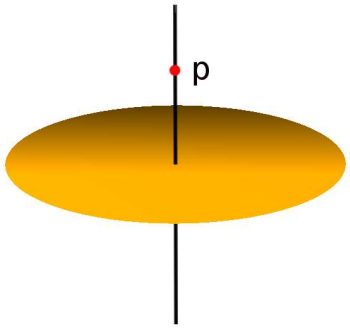


FIGURE 18. Dimension locally at p.

B) **Standard bases in local rings**

Remark 4.8 (Computing in local rings)
We can compute in the ring $K[x]_{\langle x_1,\ldots,x_n\rangle}$ as we did in $K[x]$ provided that we work with a *local* monomial ordering instead of a global one. The algorithms behind the scenes are somewhat more involved and some notions have to be slightly adjusted, but we will not do so here. The philosophy is that everything works basically in the same way as for global orderings. In local rings one rather uses the notion of *standard basis* than *Gröbner basis*.

Example 4.9 (Local monomial orderings)

a. Define the *local lexicographical ordering* $>_{ls}$ on $\mathrm{Mon}(x)$ by $x^\alpha >_{ls} x^\beta$ if

$$\exists\, i \in \{1,\ldots,n\} : \alpha_1 = \beta_1, \ldots, \alpha_{i-1} = \beta_{i-1}, \text{ and } \alpha_i < \beta_i.$$

b. Define the *local degree reverse lexicographical ordering* $>_{ds}$ on $\mathrm{Mon}(x)$ by $x^\alpha >_{ds} x^\beta$ if

$$\deg\left(x^\alpha\right) < \deg\left(x^\beta\right),$$

or $\deg\left(x^\alpha\right) = \deg\left(x^\beta\right)$, but then

$$\exists\, i \in \{1,\ldots,n\} : \alpha_n = \beta_n, \ldots, \alpha_{i+1} = \beta_{i+1}, \text{ and } \alpha_i < \beta_i.$$

c. E.g., $x_1^2 >_{ds} x_1x_2^3$, but $x_1x_2^3 >_{ls} x_1^2$.

Example 4.10 (Computing dimensions locally at a point)
If one wants to compute the dimension of an affine algebraic variety locally at a point p, then one should first move the point p to the origin by substituting $x_i + p_i$ for x_i in the defining polynomials and then one can compute the dimension of the ideal in the local ring $K[x]_{\langle x_1,\ldots,x_n\rangle}$.
E.g. $X = V(xz, yz)$ and $p = (0,0,1)$ then the following SINGULAR commands compute the dimension of X locally at p, where in the definition of the ring we use the local ordering `ls`. Note also that we have to replace the command `groebner` by the command `std` to compute a standard basis:

```
> ring r=0,(x,y,z),ds;
> ideal I=xz,yz;
> I=subst(I,z,z+1);
> I;
I[1]=x+xz
I[2]=y+yz
> dim(std(I));
1
```

C) **The tangent space of X at p**

Remark 4.11
If X is an affine algebraic variety, $p \in X$ and $\mathfrak{m}_p$ is the maximal ideal of the local ring $K[X]_{\mathfrak{m}_p}$ of X at p, then $\mathfrak{m}_p/\mathfrak{m}_p^2$ is a finite dimensional K-vector space generated by $x_1 - p_1, \ldots, x_n - p_n$, so that

$$\dim_K \mathfrak{m}_p/\mathfrak{m}_p^2 \leq n.$$

We call $\mathfrak{m}_p/\mathfrak{m}_p^2$ the *Zariski cotangent space* of X at p.

Remark 4.12 (Tangent space to a hypersurface)
It is known from basic courses in calculus that the tangent hyperplane at a point p to the level space $f^{-1}(c)$ of a function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ has the gradient of f at p as its normal vector. In our terminology this mean that the tangent hyperplane to the hypersurface $X = V(f - c)$ at a point p is given by

$$H = V\left(\frac{\partial f}{x_1}(p) \cdot (x_1 - p_1) + \ldots + \frac{\partial f}{x_n}(p) \cdot (x_n - p_n)\right).$$

Note that this is indeed a hyperplane unless the gradient of f at p vanishes identically.

It is custom to move the tangent hyperplane H to the origin by subtracting p in order to end up with a K-vector space. We thus call the K-vector space

$$T_p(X) = V\left(\frac{\partial f}{x_1}(p) \cdot x_1 + \ldots + \frac{\partial f}{x_n}(p) \cdot x_n\right) = \operatorname{Ker}\left(\frac{\partial f}{x_1}(p), \ldots, \frac{\partial f}{x_n}(p)\right)$$

the *tangent space* of X.

This leads us to the following generalisation of the notion of tangent space.

Theorem 4.13 (The tangent space of X at p)
Let $X \subseteq \mathbb{A}_K^n$ be an affine algebraic variety with $I(X) = \langle f_1, \ldots, f_k \rangle$ and let $p \in X$. Then

$$\mathfrak{m}_p/\mathfrak{m}_p^2 \cong \operatorname{Ker}\big(Df(p)\big)$$

as K-vector spaces, where

$$Df(p) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(p) & \ldots & \frac{\partial f_1}{\partial x_n}(p) \\ \vdots & & \vdots \\ \frac{\partial f_k}{\partial x_1}(p) & \ldots & \frac{\partial f_k}{\partial x_n}(p) \end{pmatrix}$$

is the Jacobian matrix *of $f = (f_1, \ldots, f_k)$ at p.*
In particular, the vector space

$$T_p(X) = \operatorname{Ker}\big(Df(p)\big)$$

is independent of the chosen generators of $I(X)$. We call it the tangent space of X at p.

Example 4.14
Consider the affine algebraic variety $X = V(x^2 + y^2 - z)$ and $p = (1, 0, 1)$.

```
> ring r=0,(x,y,z),dp;
> ideal I=x2+y2-z;
> matrix J[1][3]=jacob(I);
> print(J);
2x,2y,-1
> LIB "poly.lib";
> J=substitute(J,x,1,y,0,z,1);
> print(J);
2,0,-1
> print(syz(J));
0,1,
1,0,
0,2
```

We have thus computed the tangent space of X at p to be

$$T_p(X) = \mathrm{Ker}(2 \;\; 0 \;\; -1) = \left\langle \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix} \right\rangle_K = V(2x - z).$$

In the SINGULAR code above we have loaded the library `poly.lib` to have the command `substitute` at hand which allows to substitute values for the variables `x`, `y` and `z` all at once. Moreover, we have used the command `jacob` in order to compute the Jacobian matrix of the generators of I and we have used the command `syz` in order to compute generators of the kernel of the Jacobian matrix. In Figure 19 we have translated the tangent space by the point p, and we have marked p by a small green sphere.

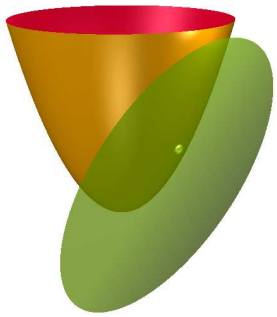


FIGURE 19. A tangent space.

D) **Regular and singular points**

Proposition 4.15 (Local version of Krull's Principle Ideal Theorem)
If X *is an affine algebraic variety over* $K = \overline{K}$ *and* $p \in X$*, then*

$$\dim(X, p) = \dim K[X]_{\mathfrak{m}_p} \leq \dim_K \mathfrak{m}_p/\mathfrak{m}_p^2 = \dim_K T_p(X).$$

Idea of the proof: The dimension of $\mathfrak{m}_p/\mathfrak{m}_p^2$ is by Nakayama's Lemma the minimal number of generators of $\mathfrak{m}_p$ and by Krull's Principle Ideal Theorem this is an upper bound for the length of a chain of prime ideals ending at $\mathfrak{m}_p$, which is the dimension of $K[X]_{\mathfrak{m}_p}$. □

One expects of course that the tangent space to a geometric object has the same dimension as the object itself. That is the *regular* behaviour, everything else is *irregular* or *singular*.

Definition 4.16 (Regular and singular points)
Let X be an affine algebraic variety and $p \in X$. We call p *regular* if $\dim(X, p) = \dim_K T_p(X)$, i.e. the dimension of X locally at p coincides with the dimension of the tangent space to X at p. Otherwise we call p *singular* or a *singularity*, and that means that $\dim(X, p) < \dim_K T_p(X)$. We denote by $\mathrm{Reg}(X)$ all regular points of X and by $\mathrm{Sing}(X)$ all singular points of X.

Remark 4.17
If $f_1, \ldots, f_k$ generate $I(X)$ and $f = (f_1, \ldots, f_k)$, then p is regular if and only if

$$\dim(X, p) = \dim_K T_p(X) = n - \operatorname{rank}\big(Df(p)\big).$$

This can be generalised even if we do not know that the f_i generate the vanishing ideal of X.

Theorem 4.18 (Jacobian Criterion)
Let $X = V(f_1, \ldots, f_k) \subseteq \mathbb{A}^n_K$ *with* $K = \overline{K}$ *and* $f = (f_1, \ldots, f_k)$*.*
Then $p \in X$ *is regular if and only if*

$$\dim(X, p) \geq n - \operatorname{rank}\big(Df(p)\big).$$

Idea of the proof: Choose polynomials $g_1, \ldots, g_l$ such that the vanishing ideal is $I(X) = \langle f_1, \ldots, f_k, g_1, \ldots, g_l \rangle$. Setting $F = (f_1, \ldots, g_l)$ we have by definition

$$\dim_K T_p(X) = n - \operatorname{rank}\big(DF(p)\big) \leq n - \operatorname{rank}\big(Df(p)\big).$$

□

Corollary 4.19 (Jacobian Criterion for hypersurfaces)
If $X = V(f) \subseteq \mathbb{A}^n_K$ *with* $K = \overline{K}$ *is a hypersurface and* $f \in K[\mathbf{x}]$ *is squarefree, then* p *is a singular point of* X *if and only if*

$$f(p) = \frac{\partial f}{\partial x_1}(p) = \ldots = \frac{\partial f}{\partial x_n}(p) = 0.$$

Idea of the proof: The hypersurface X has dimension $n-1$ locally at each point, so that a point on X is singular if and only if the Jacobian matrix of f at p has rank zero, i.e. all partial derivatives vanish at p. That $p \in X$ requires $f(p) = 0$. □

Corollary 4.20 (The regular locus is open and dense in X.)
If X *is an affine algebraic variety, then* $\mathrm{Reg}(X)$ *is open and dense in* X.
In particular, $\mathrm{Sing}(X)$ *is an affine algebraic variety.*

Idea of the proof: Let $I(X) = \langle f_1, \ldots, f_k \rangle$ and $f = (f_1, \ldots, f_k)$. That the rank of the Jacobian matrix $Df(p)$ is smaller than a certain value can be expressed by the vanishing of certain minors of the matrix. Thus $\mathrm{Sing}(X)$ has these minors and the f_i as equations. □

Example 4.21
Newton's node $X = V(y^2 - x^2 - x^3) \subseteq \mathbb{A}^2_K$ is an irreducible curve and has thus dimension one locally at each point. The singular points are thus those points *on the curve* where the rank of the Jacobian is zero, i.e. where the Jacobian vanishes.

```
> ring r=0,(x,y),dp;
> ideal I=y2-x2-x3;
> ideal J=I,jacob(I);
> LIB "primdec.lib";
> minAssGTZ(J);
[1]:
   _[1]=y
   _[2]=x
```

Thus the origin $p = (0, 0)$ is the only singular point of X, i.e. it is the only point where we have difficulties to say what the tangent line should be (see Figure 20).

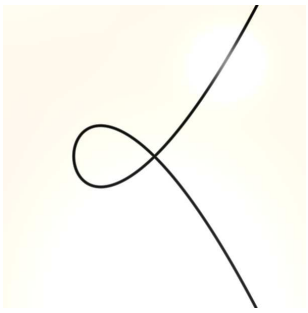


FIGURE 20. Newton node $V(y^2 - x^2 - x^3)$

Example 4.22
Consider the affine algebraic variety $X = V(x^2 - y^3 + z^2, z - y^2) \subseteq \mathbb{A}^2_K$. We want to compute the dimension of X as well as its singular locus.

```
> ring r=0,(x,y,z),dp;
> ideal I=x2-y3+z2,z-y2;
> size(minAssGTZ(I));
1
> dim(groebner(I));
1
> matrix J[2][3]=jacob(I);
> ideal JJ=I,minor(J,2);
> LIB "primdec.lib";
> minAssGTZ(JJ);
[1]:
   _[1]=-y2+z
   _[2]=y
   _[3]=x
```

We have first checked that X is indeed an irreducible space curve. Then we defined the Jacobian matrix of the given two equations, which is a 2×3-matrix. The Jacobian Criterion says that a point p is singular if and only if the rank of the Jacobian matrix is strictly smaller than $n - \dim(X, p) = 3 - 1 = 2$. That is the case if and only if the 2×2-minors of the Jacobian vanish. Thus the singular locus of X is an algebraic variety which has the two equations in I together with the 2×2-minors of the Jacobian as entries. We computed this ideal as JJ above. Then we computed the associated primes of `JJ`, and it turns out, that the irreducible space curve X has only the origin as singular point (see Figure 22).
SINGULAR offers a short cut for the computation of `JJ` via the command `slocus` from the library `sing.lib`.

```
> ring r=0,(x,y,z),dp;
> ideal I=x2-y3+z2,z-y2;
> LIB "sing.lib";
> ideal JJ=slocus(I);
> LIB "primdec.lib";
> minAssGTZ(JJ);
[1]:
   _[1]=-y2+z
   _[2]=y
   _[3]=x
```

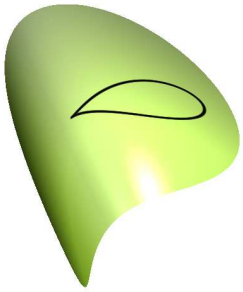


FIGURE 21. The singular space curve $V(x^2 - y^3 + z^2, z - y^2)$.

E) **Intersection multiplicity of two plane curves at a point**

Definition 4.23 (Intersection multiplicity)
Let $X = V(f)$ and $Y = V(g)$ be two plane curves given by squarefree polynomials $f, g \in K[x, y] \setminus K$, and let $p \in X \cap Y$.

a. We define the *intersection multiplicity* of X and Y at p as

$$\mathrm{mult}_p(X \cap Y) = \dim_K K[x, y]_{\mathfrak{m}_p}/\langle f, g\rangle.$$

Unless the two curves share a common component at p this is a positive integer.

b. We say that X and Y meet *non-transversally* at p if one of the tangent spaces $T_p(X)$ respectively $T_p(Y)$ is contained in the other, i.e. either p is a singular point of one of the two curves or they have the same tangent line at p.

Proposition 4.24 (Intersection multiplicity)
Let $f, g \in K[x, y] \setminus K$ *be squarefree,* $X = V(f)$ *and* $Y = V(g)$ *and* $p \in X \cap Y$.
Then X *and* Y *meet non-transversally at* p *if and only if* $\mathrm{mult}_p(X \cap Y) \geq 2$.

Idea of the proof: If the curves X and Y meet transversally at p then f and g generate the maximal ideal in $K[x, y]_{\mathfrak{m}_p}$. □

Example 4.25
We can again compute the intersection multiplicity. For that we should move the point p in question to the origin, since we want to compute in $K[x, y]_{\langle x,y\rangle}$.
E.g. we can compute the intersection multiplicity of the circle $X = V(x^2 + y^2 - 1)$ and the line $Y = V(x - 1)$ at the intersection point $p = (1, 0)$ as follows, where the command `vdim` computes the vector space dimension of the $K[x, y]_{\langle x,y\rangle}$ modulo the given ideal:

```
> ring r=0,(x,y),ds;
> poly f=x2+y2-1;
```

```
> poly g=x-1;
> ideal I=f,g;
> I=subst(I,x,x+1);
> vdim(std(I));
2
```

Thus the intersection multiplicity is

$$\mathrm{mult}_p(X \cap Y) = 2,$$

and the line meets the circle non-transversally, which is not surprising since it is the tangent line to the circle at this point.

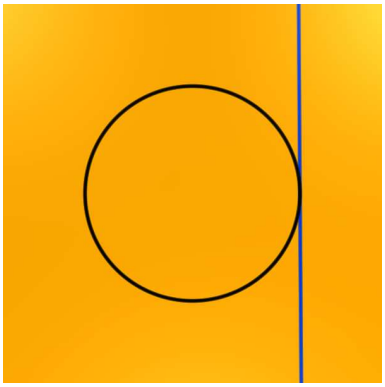


FIGURE 22. The intersection of the two curves is non-transversal.

Definition 4.26 (The multiplicity of a curve at a point)
For a squarefree $f = \sum_{i,j} a_{ij} \cdot x^i \cdot y^j \in K[x,y]$ and $X = V(f)$ we call

$$\mathrm{mult}_p(X) = \mathrm{ord}(f) = \inf\{i+j \mid a_{ij} \neq 0\}$$

the *multiplicity* of X at the origin. The multiplicity at an arbitrary point p is defined by first moving the point to the origin and then computing the multiplicity there.

Example 4.27
The multiplicity of $V(f)$ at the origin is easy to compute with SINGULAR via the command `mult`, but one has to use a *local* ordering, since one should rather think of f as a power series (see also Remark 4.29).

```
> ring r=0,(x,y),ds;
> poly f=x3y-5x2y2+7x4y5;
> mult(std(f));
4
```

Proposition 4.28 (Multiplicity versus intersection multiplicity)
If X *and* Y *are two plane curves with* $p \in X \cap Y$*, then*

$$\mathrm{mult}_p(X \cap Y) \geq \mathrm{mult}_p(X) \cdot \mathrm{mult}_p(Y).$$

Idea of the proof: We consider just the case that $Y = V(y)$ and $p = (0,0)$. From the definition it is clear that

$$\mathrm{mult}_p(f) \cdot \mathrm{mult}_p(y) = \mathrm{mult}_p(f) \leq \mathrm{mult}_p\big(f(x,0)\big).$$

Moreover, for $m = \mathrm{mult}_p\big(f(x,0)\big)$ we have

$$f(x,0) = x^m \cdot u$$

for some unit $u \in K[x]_{\langle x\rangle}$, so that

$$\begin{aligned}\mathrm{mult}_p(X \cap Y) &= \dim_K K[x,y]_{\langle x,y\rangle}/\langle f, y\rangle \\ &= \dim_K K[x,y]_{\langle x,y\rangle}/\langle f(x,0), y\rangle \\ &= \dim_K K[x]_{\langle x\rangle}/\langle f(x,0)\rangle \\ &= \dim_K K[x]_{\langle x\rangle}/\langle x^m\rangle = m.\end{aligned}$$

It thus follows

$$\mathrm{mult}_p(f) \cdot \mathrm{mult}_p(y) \leq m = \mathrm{mult}_p(X \cap Y).$$

□

F) **Local parametrisations**

Remark 4.29 (The power series ring)
If $g \in K[x,y]$ is a polynomial with a non-zero constant term $c = g(0,0) \neq 0$, then

$$\frac{1}{g} = \frac{1}{c} \cdot \frac{1}{1 - \frac{c-g}{c}} = \frac{1}{c} \cdot \sum_{k=0}^{\infty} \left(\frac{c-g}{c}\right)^k \in K[[x,y]]$$

can be written as a formal power series, i.e. it is an element of the *ring of formal power series*

$$K[[x,y]] = \left\{ \sum_{i+j=0}^{\infty} a_{ij} \cdot x^i \cdot y^j \;\middle|\; a_{ij} \in K \right\}.$$

This shows that the local ring

$$K[x,y]_{\langle x,y\rangle} \subset K[[x,y]]$$

is a subring of the ring of formal power series. Moreover, if $f, g \in K[x,y]$ then

$$\dim_K K[x,y]_{\langle x,y\rangle}/\langle f, g\rangle = \dim_K K[[x,y]]/\langle f, g\rangle,$$

that is, it does not make any difference if we compute intersection multiplicities in the local ring $K[x,y]_{\langle x,y\rangle}$ or in the power series ring.
Moreover, one can even define local monomial orderings and standard bases for the formal power series ring in the same way as for $K[x,y]_{\langle x,y\rangle}$, and a finite set of *polynomials* will be a standard basis in $K[x,y]_{\langle x,y\rangle}$ if and only if it is one in the power

series ring. Thus, as long as we start with polynomial generators we can compute in the power series ring $K[[x,y]]$.
There is, however, one remarkable difference between the rings $K[x,y]_{\langle x,y\rangle}$ and $K[[x,y]]$. A polynomial f which is irreducible in $K[x,y]_{\langle x,y\rangle}$ may very well decompose into several irreducible factors in $K[[x,y]]$. These are then called the *branches* of the plane curve locally at the origin. E.g. the polynomial $f = y^2 - x^2 - x^3$ is irreducible in $\mathbb{R}[x,y]_{\langle x,y\rangle}$, but it decomposes in $\mathbb{R}[[x,y]]$ as

$$f = \left(y - x \cdot \sqrt{1+x}\right) \cdot \left(y + x \cdot \sqrt{1+x}\right)$$

since $\sqrt{1+x}$ can be expanded into a power series using the Taylor formula. This reflects the fact that in a *small Euclidean neighbourhood* of the origin the curve $V(y^2 - x^2 - x^3)$ has two components, its branches (see Figure 23).

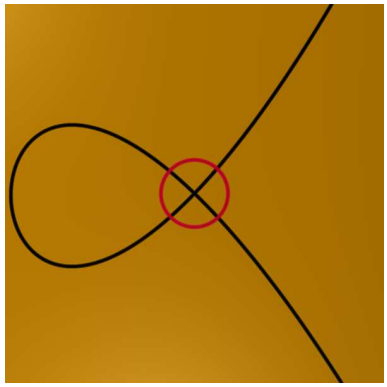


FIGURE 23. A plane curve with two branches locally at the origin

Remark 4.30 (Parametrisations of plane curve singularities)
A global parametrisation of a plane curve $\subseteq \mathbb{A}^2_K$ is a surjective polynomial map

$$\varphi : \mathbb{A}^1_K \longrightarrow X,$$

but only so called rational curves admit such parametrisations and rational curves are rare.
For many questions it suffices to have *local parametrisations*, i.e. parametrisations of the branches of a curve locally in a point. Suppose that $X = V(f) \subseteq \mathbb{A}^2_K$ is an algebraic plane curve through the origin and that $X(t), Y(t) \in K[[t]]$ are two power series such that $f\big(X(t), Y(t)\big) = 0$, then we call the assignment

$$t \mapsto \big(X(t), Y(t)\big)$$

a *local parametrisation* of a branch of X, and we call a local parametrisation *primitive* if it is not derived from another local parametrisation by replacing t by some power of t. E.g.

$$t \mapsto \left(t, t \cdot \sqrt{1+t}\right)$$

is a primitive local parametrisation of the branch $y - x \cdot \sqrt{1+x}$ of the plane curve $V(y^2 - x^2 - x^3)$.

Such local parametrisation can be computed via *Puiseux expansion* in characteristic zero or *Hamburger-Noether expansion* in arbitrary characteristic. We will not explain the theory of those here, but rather show how to compute primitive local parametrisations in SINGULAR instead. The command `hnexpansion` is used to compute a Hamburger-Noether expansion of each branch, and the command `param` can then be used to compute the parametrisation for each branch up to some finite order — we can of course not compute all infinitely many terms of a power series. Both commands belong to the library `hnoether.lib`.

```
> LIB "hnoether.lib";
> ring r=0,(x,y),ds;
> poly f=y2-x2-x3;
> list HNE=hnexpansion(f);
> size(HNE);
2
> param(HNE[1]);
// ** Warning: result is exact up to order 2 in y !
_[1]=x
_[2]=x+1/2x2
> param(HNE[2]);
// ** Warning: result is exact up to order 2 in y !
_[1]=x
_[2]=-x-1/2x2
```

For the example of the curve given by $f = y^2 - x^2 - x^3$ compute two branches and their local parametrisations up to order 2. If we want to have a parametrisation up a higher order we can extend the Hamburger-Noether expansion by the command `extdevelop`.

```
> list L=extdevelop(HNE[1],5);
> param(L);
// ** Warning: result is exact up to order 5 in y !
_[1]=x
_[2]=x+1/2x2-1/8x3+1/16x4-5/128x5
```

Local parametrisations can now be used to compute intersection multiplicities. For this we recall that the order of a power series $h = \sum_{k=0}^{\infty} a_k \cdot t^k \in K[[t]]$ is

$$\mathrm{ord}_t(h) = \inf\{k \mid a_k \neq 0\}.$$

Proposition 4.31 (Intersection multiplicities via local parametrisations)
Let $f, g \in K[x,y] \setminus K$ *and suppose that* $\big(X_1(t), Y_1(t)\big), \ldots, \big(X_k(t), Y_k(t)\big)$ *are primitive local parametrisations of the* k *branches of* $V(g)$ *at the origin* $p = (0,0)$*, then*

$$\operatorname{mult}_p\big(V(f) \cap V(g)\big) = \sum_{i=1}^{k} \operatorname{ord}_t f\big(X_i(t), Y_i(t)\big).$$

In particular, the intersection multiplicity behaves additive w.r.t. the branches.

Idea of the proof: Let us prove the statement in the case where $V(g)$ is a line. We may assume that $g = y - ax$ so that

$$\langle f, g\rangle = \langle f(x, ax), y - ax\rangle \lhd K[[x,y]].$$

If we now apply the coordinate transformation

$$\varphi : K[[x,y]] \xrightarrow{\cong} K[[x,y]] : x \mapsto x, y \mapsto y - ax$$

we see that

$$\begin{aligned} K[[x,y]]/\langle f, g\rangle =& K[[x,y]]/\langle f(x, ax), y - ax\rangle \\ \cong& K[[x,y]]/\langle f(x, ax), y\rangle \cong K[[x]]/\langle f(x, ax)\rangle. \end{aligned}$$

Note that $t \mapsto (t, at)$ is a primitive parametrisation of $V(g)$ and if $m = \operatorname{ord}_t(f(t, at))$ is the order of $f(t, at)$, then

$$f(x, ax) = x^{\operatorname{ord}_x(f(x,ax))} \cdot u = x^m \cdot u$$

for some unit $u \in K[[x]]$, so that

$$K[[x]]/\langle f(x, ax)\rangle = K[[x]]/\langle x^m\rangle.$$

But then

$$\begin{aligned} \operatorname{mult}_p\big(V(f) \cap V(g)\big) =& \dim_K K[[x,y]]/\langle f, g\rangle \\ =& \dim_K K[[x]]/\langle x^m\rangle = m = \operatorname{ord}_t\big(f(t, at)\big) \end{aligned}$$

as required. □

Example 4.32 (Intersection multiplicities and local parametrisations)
Let $f = x^2 - 2xy + y^2 - x - y$, $g = y^2 - x^2 - x^3$ and $p = (0,0)$.

```
> LIB "hnoether.lib";
> ring r=0,(x,y),ds;
> poly f=x2-2xy+y2-x-y;
> poly g=y2-x2-x3;
> list HNE=hnexpansion(g);
> ideal XY=param(HNE[1]);
> ord(substitute(f,x,XY[1],y,XY[2]));
1
```

```
> XY=param(HNE[2]);
> ord(substitute(f,x,XY[1],y,XY[2]));
2
> vdim(std(ideal(f,g)));
3
```

The computation shows that the first branch of g gives order 1 and the second branch of g gives order 2 so that the intersection multiplicity of f and g in the origin is $1+2$. This was verified by computing the intersection multiplicity directly.

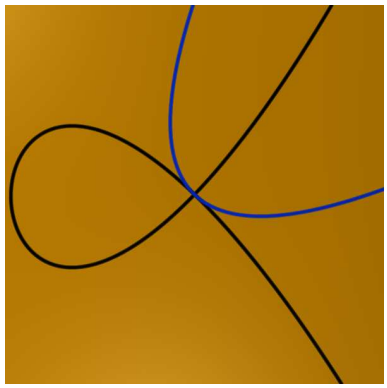


FIGURE 24. An intersection of multiplicity 3

G) **Exercises**

Exercise 4.33
Let $X = V(I) \subseteq \mathbb{A}^3_{\mathbb{C}}$ where

$$\begin{aligned} I = \langle & x^2y + xy^2 + xyz - xz^2 - yz^2 - z^3, \\ & x^3 + 2x^2y + xy^2 + x^2z + xyz - xz^2 - yz^2 - z^3 - xz - yz - z^2, \\ & x^2z^2 + xyz^2 + xz^3 - xyz - y^2z - yz^2 \rangle. \end{aligned}$$

Show that the points $p = (0,0,0)$ and $q = (0,1,0)$ are contained in X and compute the dimension of X locally at these points.

Exercise 4.34
Consider the space curve X given by the parametrisation

$$\mathbb{A}^1_{\mathbb{C}} \longrightarrow \mathbb{A}^3_{\mathbb{C}} : t \mapsto (t^3, t^4, t^5).$$

Compute the tangent space of X at the point $(1,1,1)$. Visualise the space curve and the tangent line at $(1,1,1)$ with `surfex`.

Exercise 4.35
Check if the origin is a singular point of the variety X in Exercise 4.33.

Exercise 4.36
Compute the singular locus $\mathrm{Sing}(X)$ for $X = V(x^2 - y^3, x - y^2 - z^2 + 1)$.

Exercise 4.37

Compute for the following polynomials f and g the intersection multiplicity of $X = V(f)$ and $Y = V(g)$ at $p = (0, 0)$:

a. $f = x^3 + 4xy^4$ and $g = x + y$.

b. $f = \left(x^2 + y^2\right)^2 + 3x^2y - y^3$ and $g = x + 2y$.

c. $f = \left(x^2 + y^2\right)^2 + 3x^2y - y^3$ and $g = y$.

d. $f = \left(x^2 + y^2\right)^2 + 3x^2y - y^3$ and $g = y - x^2$.

e. $f = \left(x^2 + y^2\right)^2 + 3x^2y - y^3$ and $g = y^2 - x$.

f. $f = \left(x^2 + y^2\right)^2 + 3x^2y - y^3$ and $g = y^2 - x^2$.

In which of the cases do we have

$$\mathrm{mult}_p(X \cap Y) = \mathrm{mult}_p(X) \cdot \mathrm{mult}_p(Y)?$$

Visualise X and Y locally at $p = (0, 0)$ with `surfex`.

Exercise 4.38

Compute local parametrisations at the origin of the plane curves in Exercise 4.37 and compute the intersection multiplicities there with the aid of these local parametrisations.

5 Projective plane curves

The following section is a first introduction into aspects of projective geometry.

A) **The projective plane**

Remark 5.1 (Defects of the affine plane)
If we consider the lines

$$L_1 = V(x - y - 1)$$

and

$$L_2 = V(x + y - 1)$$

in the affine plane $\mathbb{A}^2_{\mathbb{R}}$, we find that they intersect in a point

$$L_1 \cap L_2 = \{(1, 0)\},$$

while the line L_1 and the line

$$L_3 = V(x - y + 1)$$

do not intersect at all, they are parallel. This distinction is rather unsatisfactory,

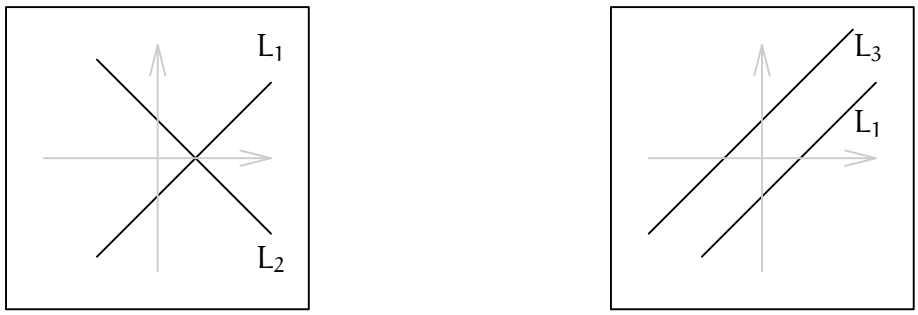


FIGURE 25. Intersection types of lines in the affine plane $\mathbb{A}^2_{\mathbb{R}}$

and projective geometry is a way to get around this problem by adding points, as we say, at infinity.

Definition 5.2 (The projective plane)
We define the *projective plane* $\mathbb{P}^2_K$ to be the set of lines through the origin in affine 3-space $\mathbb{A}^3_K$. We denote the line through the origin determined by a non-zero point $0 \neq p = (p_0, p_1, p_2) \in \mathbb{A}^3_K$ by

$$P = (p_0 : p_1 : p_2) = \{\lambda \cdot p \mid \lambda \in K\} \in \mathbb{P}^2_K$$

and call the p_i the *homogeneous coordinates* of P.

Remark 5.3 ($\mathbb{A}^2_K$ as a subset of $\mathbb{P}^2_K$)
Let us consider the plane E in affine 3-space parallel to the xy-plane through the point $(0, 0, 1)$, which can be viewed as a copy of $\mathbb{A}^2_K$ (see Figure 26).

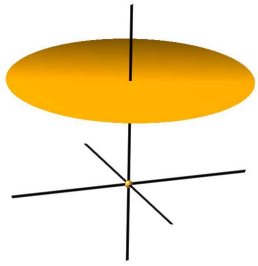


FIGURE 26. The plane $E \triangleq \mathbb{A}^2_K$ in affine 3-space

Each point $P = (p_0 : p_1 : p_2) \in \mathbb{P}^2_{\mathbb{R}}$ is a line through the origin in affine 3-space. The line intersects E if and only if it is not contained in the xy-plane. Let us denote the set of lines in the xy-plane by

$$\mathbb{P}^1_K = \{(p_0 : p_1 : 0) \mid (p_0, p_1) \neq (0, 0)\}$$

and call it the *line at infinity*, then with $U_z = \mathbb{P}^2_K \setminus \mathbb{P}^1_K \triangleq E$ we have

$$\mathbb{P}^2_{\mathbb{R}} \triangleq U_z \cup \mathbb{P}^1_K \triangleq E \cup \mathbb{P}^1_K \triangleq \mathbb{A}^2_K \cup \mathbb{P}^1_K.$$

Thus $\mathbb{P}^2_K$ is the affine plane together with some additional points — we will explain in Example 5.9, why we call $\mathbb{P}^1_K$ a *line*.

B) **Projective plane curves**

Remark 5.4 (Evaluating a polynomial at a point in $\mathbb{P}^2_K$)
Note that the homogeneous coordinates of a point in $\mathbb{P}^2_K$ are only determined up to a common scalar $0 \neq \lambda \in K$. That makes it difficult to evaluate a polynomial $f \in K[x, y, z]$ at P by inserting the coordinates. E.g. let $P = (1 : 2 : 1) = (2 : 4 : 2)$ and $f = 2xz - y$ then

$$f(1, 2, 1) = 0 \neq 4 = f(2, 4, 2).$$

We see that it is in general not even possible to say whether a point in $\mathbb{P}^2_K$ is a zero of a polynomial. However, the latter is possible, if we restrict to homogeneous polynomials, where a polynomial $F \in K[x, y, z]$ of degree d is called *homogeneous* if for $0 \neq \lambda \in K$

$$F(\lambda \cdot x, \lambda \cdot y, \lambda \cdot z) = \lambda^d \cdot F(x, y, z), \tag{3}$$

or equivalently if all monomials of F have degree d. E.g. $F = x^2 - 3xy + z^2$ is homogeneous while $f = x^2 - 3xy + 1$ is not.
Note that (3) implies that

$$F(p) = 0 \iff F(\lambda \cdot p) = 0 \quad \forall\, 0 \neq \lambda \in K,$$

and we may define $F(P) = 0$ for $P \in \mathbb{P}^2_K$ if $F(p) = 0$ for some non-zero point p on P. We will in the sequel use capital letters (e.g. F) for homogeneous polynomials in $K[x, y, z]$ and lower case letters (e.g. f) for not necessarily homogeneous polynomials in $K[x, y]$ or $K[x, y, z]$.

The SINGULAR command `homog` can be used to check if a polynomial is homogeneous. The return value `1` means `TRUE` and the return value `0` means `FALSE`.

```
> ring r=0,(x,y,z),dp;
> homog(x2-xy+z2);
1
> homog(x2-xyz);
0
```

Definition 5.5
A *projective plane curve* is the zero locus

$$V(F) = \{P \in \mathbb{P}^2_K \mid F(P) = 0\} \subset \mathbb{P}^2_K$$

of a non-constant homogeneous polynomial $F \in K[x, y, z] \setminus K$. If F is squarefree we call $\deg(F)$ the *degree* of the projective plane curve and denote it by $\deg(V(F))$. Note that by abuse of notation we have used the same notation for the surface

$$V(F) = \{p \in \mathbb{A}^3_K \mid F(p) = 0\} \subset \mathbb{A}^3_K$$

which we call the *cone* over the corresponding projective curve. It should always be clear from the context what $V(F)$ actually means.

Remark 5.6 (The irreducible components of a curve)
A squarefree homogeneous polynomial $F \in K[x, y, z]$ factorises into a product $F = F_1 \cdots F_k$ of irreducible homogeneous polynomials and

$$V(F) = V(F_1) \cup \ldots \cup V(F_k).$$

We call the $V(F_i)$ the *irreducible components* of $V(F)$.

```
> ring r=0,(x,y,z),dp;
> poly F=-x2y3+y5+x4z-x2y2z;
> homog(F);
1
> factorize(F);
[1]:
   _[1]=1
   _[2]=x-y
   _[3]=x+y
   _[4]=-y3+x2z
[2]:
   1,1,1,1
```

Remark 5.7 (The Zariski topology on $\mathbb{P}^2_K$)
It is easy to see that

$$\{X \subset \mathbb{P}^2_K \mid X \text{ finite}\} \cup \{V(F) \mid F \text{ homogeneous}\} \cup \{\mathbb{P}^2_K\}$$

are the closed sets of a topology on $\mathbb{P}^2_K$, the *Zariski topology*.

Remark 5.8 (Homogenisation and projective closure)
If $f = \sum_{i,j} a_{i,j} \cdot x^i \cdot y^j \in K[x,y]$ is a non-constant polynomial of degree d, then we define its *homogenisation* as

$$F = \sum_{i,j} a_{i,j} \cdot x^i \cdot y^j \cdot z^{d-i-j} \in K[x,y,z].$$

It is a homogeneous polynomial and thus defines a projective plane curve $V(F)$. If we consider the homeomorphism

$$\varphi : \mathbb{A}^2_K \xrightarrow{\cong} U_z \subset \mathbb{P}^2_K : (p_0, p_1) \mapsto (p_0 : p_1 : 1),$$

then it turns out that

$$\varphi\big(V(f)\big) = V(F) \cap U_z$$

and $V(F)$ is the *topological closure* of $V(f)$ which we get by just adding points at infinity. We call it also the *projective closure* of $V(f)$.
The SINGULAR command `homog` can also be used to homogenise a polynomial.

```
> ring r=0,(x,y,z),dp;
> poly f=y2-x2-x3;
> homog(f,z);
-x3-x2z+y2z
```

Example 5.9 (Lines in $\mathbb{P}^2_K$)
A projective *line* in the projective plane $\mathbb{P}^2_K$ is the projective plane curve defined by a homogeneous *linear* polynomial $F = ax + by + cz$. Note that the line $V(F)$ is thus the projective closure of the line $V(xy + by + c)$ in the affine plane, and note also that the *line at infinity*

$$\mathbb{P}^1_K = V(z)$$

is indeed a projective line in the projective plane. It is the only line which is *not* the projective closure of a line in the affine plane.
If we consider the affine cone of $V(xy + by + cz)$, then this is a plane E_L in affine 3-space through the origin. Its intersection with the plane E representing $\mathbb{A}^2_K$ is the line $L = \varphi\big(V(ax + by + c)\big)$ (see Figure 27).
Note that any two distinct lines P and Q through the origin in 3-space span a unique plane E_L through the origin in 3-space. Translating this to the projective plane means that through any two distinct points P and Q there is a unique line L as was the case in the affine plane. (See Figure 27.)

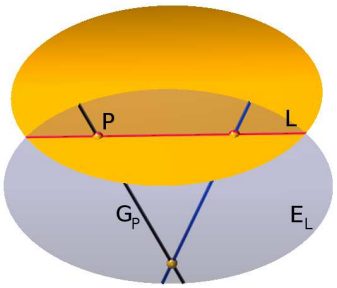


FIGURE 27. The line L through the points P and Q

Moreover, any two planes E_L and $E_{L'}$ in affine 3-space through the origin intersect in a line Q through the origin. Translating this to the projective plane means that any two lines L and L' in the projective plane intersect in a point Q. There are no parallel lines as in the affine plane! The defect of the affine plane is resolved. (See Figure 28.)

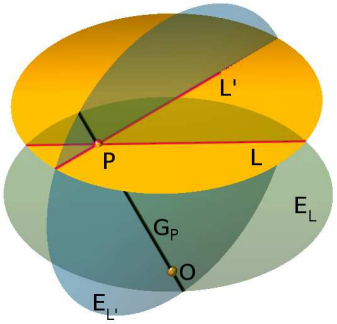


FIGURE 28. Two projective lines intersecting in a point

Remark 5.10 (Affine charts)
We have seen that the set $U_z = \mathbb{P}^2_K \setminus V(z)$ is an open and dense set of $\mathbb{P}^2_K$ which is homeomorphic to $\mathbb{A}^2_K$. Similarly, the sets $U_x = \mathbb{P}^2_K \setminus V(x)$ and $U_y = \mathbb{P}^2_K \setminus V(y)$ are open and dense in $\mathbb{P}^2_K$ and homeomorphic to $\mathbb{A}^2_K$, and moreover

$$\mathbb{P}^2_K = U_x \cup U_y \cup U_z,$$

that is they form an open cover of the projective plane. We call U_x, U_y and U_z the *affine charts* of $\mathbb{P}^2_K$, and whenever we want to study a *local* property of the projective plane or of projective curves at a point P, we can restrict to an affine chart which contains the point P. But then we are in the affine setting and the affine theory as explained in Section 4 applies. In particular, we can talk of tangent spaces, of singularities and of intersection multiplicities $\mathrm{mult}_P(X \cap Y)$ for two projective plane curves X and Y at a point $P \in X \cap Y$.

C) **Visualising the projective plane $\mathbb{P}^2_{\mathbb{R}}$**

Remark 5.11 (Visualising the affine cone of a projective plane curve)
A first method of visualising a projective plane curve in $\mathbb{P}^2_{\mathbb{R}}$ would be by visualising its affine cone. E.g. in Figure 29 we show the affine cone of the projective plane curve $V(xyz + x^2y - y^3)$.

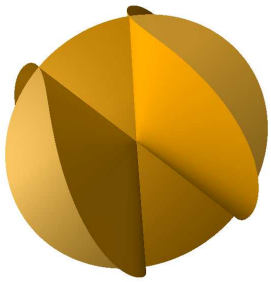


FIGURE 29. The affine cone of $V(xyz + x^2y - y^3)$.

This is, however, somewhat unsatisfactory since one expects to see a curve, i.e. a one dimensional object, rather than a surface. One can now intersect this surface with the plane E in order to see a curve (see Figure 30.), but that would only be the affine part of the curve, which again is somewhat unsatisfactory.

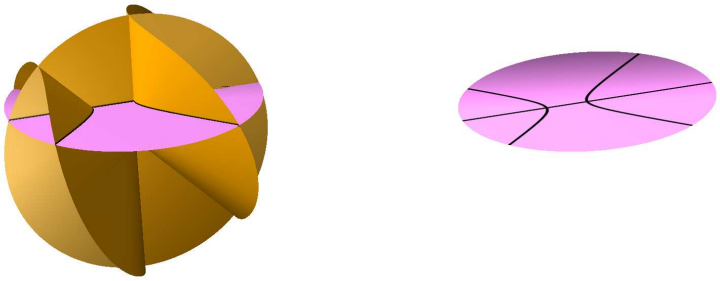


FIGURE 30. The affine cone and part of the curve $V(xyz + x^2y - y^3)$

In the following remark we explain how we can actually visualise the global picture.

Remark 5.12 (The sphere as a model for visualising $\mathbb{P}^2_{\mathbb{R}}$)
Any line through the origin meets the unit sphere in exactly two antipodal points. We may thus identify the projective plane $\mathbb{P}^2_{\mathbb{R}}$ with sets of antipodal pairs of points on the unit-sphere. Moreover, the points on the upper hemisphere (omitting the equator) are in one-to-one correspondence with the points in $E \triangleq U_z$, i.e. with the points in the affine part of the projective plane, and the equator is the line at infinity, where we have to identify antipodal points (see Figure 31).

This allows us to visualise a projective curve like $V(xyz + x^2y - y^3) \subseteq \mathbb{P}^2_{\mathbb{R}}$ (see Figure 32). The affine plane curve in Figure 30 consists of a line and two branches of a hyperbola. If we compare this to the curve in the upper hemisphere in Figure 32,

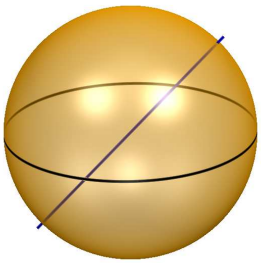


FIGURE 31. The unit sphere with the equator and a pair of antipodal points

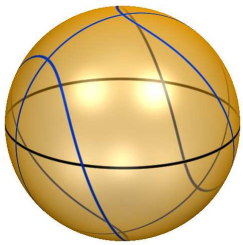


FIGURE 32. A model of $\mathbb{P}^2_{\mathbb{R}}$ with the curve $V(xyz + x^2y - y^3)$

we find there as well the line an the two branches of the hyperbola. However, the line intersects the equator in two antipodal points which have to be identified, that is, the open ends of the line are closed up by a point. Similarly, the two branches of the hyperbola intersect the equator in two sets of antipodal points which means that one end of one of the branches is connected to one end of the other branch. Topologically the hyperbola thus becomes an ellipse. Altogether this reflects the fact that projective curves or more generally projective algebraic varieties are *compact*.

D) The Theorem of Bézout for projective plane curves

The following theorem counts the number of intersection points of two projective plane curves with multiplicity and claims that the result does only depend on the degree of the two curves.

Theorem 5.13 (Bézout)
Let X *and* Y *be two projective plane curves without a common component and* $K = \overline{K}$*, then*

$$\sum_{P \in X \cap Y} \mathrm{mult}_P(X \cap Y) = \deg(X) \cdot \deg(Y).$$

In particular, X *and* Y *must intersect.*

Idea of the proof: Let us prove the statement in the special case where $Y = V(y)$ is the x-axis and $X \cap Y$ has no point on the line at infinity. We thus have $\deg(Y) = 1$ and if $X = V(F)$ for some squarefree homogeneous polynomial F, then

$$\deg(X) = \deg(F) = \deg(f) = \deg\big(f(x, 0)\big)$$

where $f = F(x, y, 1) \in K[x, y]$.Moreover, we can compute the intersection multiplicities in the affine chart $U_z \cong \mathbb{A}^2_K$. Note that

$$\langle f, y\rangle = \langle f(x, 0), y\rangle \lhd K[x, y]$$

and that the polynomial $f(x, 0) \in K[x]$ factorises into linear factors

$$f(x, 0) = c \cdot (x - c_1)^{m_1} \cdots (x - x_k)^{m_k}$$

since K is algebraically closed. But the intersection of X and Y is then

$$X \cap Y = \{(c_1 : 0 : 1), \ldots, (c_k : 0 : 1)\}.$$

If we want to compute the intersection multiplicity of $V(f)$ and $V(y)$ in $p_i = (c_i, 0)$ we first of all apply the coordinate change

$$\varphi : x \mapsto x + c_i, y \mapsto y$$

so that

$$f(x, 0) \mapsto f(x + c_i, 0) = c \cdot x^{m_i} \cdot \prod_{j\neq i}(x + c_i - c_j)^{m_j}.$$

Setting $P_i = (c_i : 0 : 1)$ and $p_i = (c_i, 0)$ we compute the intersection multiplicity

$$\begin{aligned}
\mathrm{mult}_{P_i}(X \cap Y) &= \mathrm{mult}_{p_i}\big(V(f) \cap V(y)\big) \\
&= \dim_K K[x, y]_{\langle x-c_1, y\rangle}/\langle f, y\rangle \\
&= \dim_K K[x, y]_{\langle x,y\rangle}/\langle f(x + c_i, 0), y\rangle \\
&= \dim_K K[x]_{\langle x\rangle}/\Big\langle c \cdot x^{m_i} \cdot \prod_{j\neq i}(x + c_i - c_j)^{m_j}\Big\rangle \\
&= \dim_K K[x]_{\langle x\rangle}/\langle x^{m_i}\rangle = m_i.
\end{aligned}$$

But then we have

$$\deg(X) \cdot \deg(Y) = \deg(X) = \deg\big(f(x, 0)\big) = m_1 + \ldots + m_k = \sum_{P\in X\cap Y} \mathrm{mult}_P(X \cap Y).$$

□

Example 5.14
Let us consider the two plane quadrics

$$C_t = V\big(x^2 + t \cdot y^2 - z^2\big) \subset \mathbb{P}^2_{\mathbb{R}}$$

and

$$C' = V\big(4xy - z^2\big) \subset \mathbb{P}^2_{\mathbb{R}},$$

where the parameter t varies in the interval $[1, 4]$. In order to compute the points of intersection of C_t and C' we insert $z^2 = 4xy$ into the equation $x^2 + t \cdot y^2 - z^2 = 0$. This leads to

$$x^2 + t \cdot y^2 - 4xy = 0,$$

or alternatively

$$(x-2y)^2 = (4-t)\cdot y^2.$$

Taking square roots on both sides we get two solutions

$$x = \left(2 \pm \sqrt{4-t}\right)\cdot y.$$

Plugging these into the equation $4xy - z^2 = 0$ we get

$$z^2 = \left(8 \pm 4\cdot\sqrt{4-t}\right)\cdot y^2,$$

and taking square roots on both sides gives the four solutions

$$z = \pm\sqrt{8 \pm 4\sqrt{4-t}}\cdot y.$$

In order to compute the projective coordinates of the points of intersection we choose $y = 1$ and get

$$\begin{aligned}
P_1 &= \left(2-\sqrt{4-t} : 1 : \sqrt{8-4\cdot\sqrt{4-t}}\right),\\
P_2 &= \left(2+\sqrt{4-t} : 1 : \sqrt{8+4\cdot\sqrt{4-t}}\right),\\
P_3 &= \left(2-\sqrt{4-t} : 1 : -\sqrt{8-4\cdot\sqrt{4-t}}\right),\\
P_4 &= \left(2+\sqrt{4-t} : 1 : -\sqrt{8+4\cdot\sqrt{4-t}}\right).
\end{aligned}$$

Note that only for $t < 4$ the points are pairwise different. If $t = 4$ then the points P_1 and P_2 coincide as do the points P_3 and P_4. That means we only get two points of intersection. Note also that in the latter case the tangents to C_4 and C' in P_1 coincide and the same holds in P_3. The common tangent of C_4 and C' in P_1 is $V(4x+8y-16z)$. The intersection multiplicity of C_4 and C' in P_1 as well as in P_3 is two, so that the statement of the Theorem of Bézout works out in this case as well even though the base field $\mathbb{R}$ is not algebraically closed. (See Figure 33.)

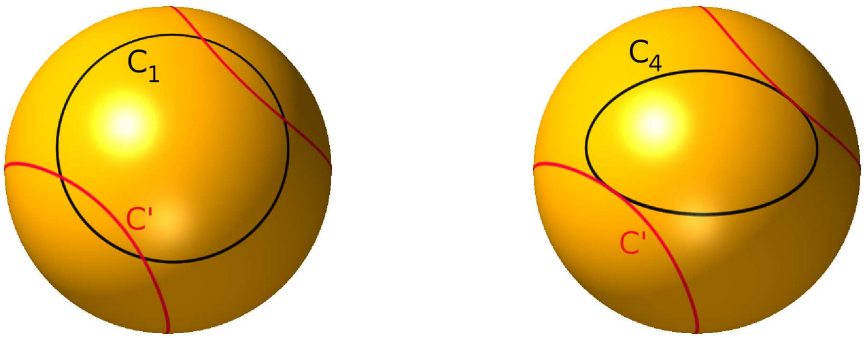


FIGURE 33. $C_1 \cap C'$ und $C_4 \cap C'$

E) **Parametrisations via the Theorem of Bézout**

Proposition 5.15 (Parametrisations via the Theorem of Bézout)
Let X *be a projective algebraic curve of degree* d *and suppose that* $P \in X$ *with* $\mathrm{mult}_P(X) = d-1$*, then there exists a parametrisation* $\varphi : \mathbb{P}^1_K \longrightarrow X$*. The analogous result for affine algebraic curves holds as well.*

Idea of the proof: The lines through the point p are actually a $\mathbb{P}^1_K$. By the Theorem of Bézout each of these lines L intersects X in exactly one further point, since

$$d = \deg(X) \cdot \deg(L) = \mathrm{mult}_P(X \cap L) + \sum_{P \neq Q \in X \cap L} \mathrm{mult}_Q(X \cap L)$$

and $\mathrm{mult}_P(X \cap L) \geq \mathrm{mult}_P(X) = d-1$. Thus we can associate to the line L (considered as a point in $\mathbb{P}^1_K$) this additional intersection point. That gives the parametrisation. □

Remark 5.16
We can actually compute a parametrisation as above using the following SINGULAR procedure if $P = (0:0:1)$.

```
proc parametrise (poly F)
"USAGE:   parametrise(F); F a homogeneous polynomial in three variables
ASSUME:   the multiplicity of f at (0:0:1) is one less than the degree of F;
          x=0 is not a tangent direction at (0:0:1)
RETURN:   a list containing a parametrisation of F"
{
  def BASERING=basering;
  ring S=(0,s,t),(x,y,z),dp;
  poly F=fetch(BASERING,F);
  poly f=subst(F,z,1);
  poly h=substitute(f,y,tx);
  h=h/x^(deg(f)-1);
  poly xnumerator=-leadcoef(h-lead(h));
  poly denominator=leadcoef(h);
  poly ynumerator=t*xnumerator;
  ring R=0,(s,t),dp;
  poly xnumerator=homog(imap(S,xnumerator),s);
  poly ynumerator=homog(imap(S,ynumerator),s);
  poly denominator=homog(imap(S,denominator),s);
  int d=deg(xnumerator);
  if (d<deg(ynumerator)){d=deg(ynumerator);}
  if (d<deg(denominator)){d=deg(denominator);}
  xnumerator=xnumerator*s^(d-deg(xnumerator));
```

```
  ynumerator=ynumerator*s^(d-deg(ynumerator));
  denominator=denominator*s^(d-deg(denominator));
  return(list(string(xnumerator),string(ynumerator),string(denominator)));
}
example
{ "EXAMPLE:";echo = 2;
  ring RING=0,(x,y,z),lp;
  poly F=x3+y3-3xyz;
  parametrisiere(F);
}
```

If we apply the procedure to the Newton node $V(y^2z-x^2z-x^3)$ we get the following parametrisation.

```
> ring r=0,(x,y,z),dp;
> poly F=y2z-x2z-x3;
> parametrise(F);
[1]:
   s3-st2
[2]:
   s2t-t3
[3]:
   -s3
```

F) **Exercises**

Exercise 5.17
Check the Theorem of Bézout for the projective plane curves $V(F)$ and $V(G)$ with

$$F = y^2z - x \cdot (x - z) \cdot (x - 2z)$$

and

$$G = y^2 + 2x^2 - 2xz.$$

Visualise both curves and their intersection in the sphere model of the projective plane.

Exercise 5.18
Compute a parametrisation of the projective plane curve

$$V\left(x^3z - 3x^2yz + 3xy^2z - y^3z + 4x^2y^2\right)$$

and visualise the curve in $\mathbb{P}^2_{\mathbb{R}}$ with `surfex`.

Exercise 5.19

Find more examples of plane curves of degree d with one singular point of multiplicity $d-1$ and compute parametrisations for these.

Exercise 5.20

Let $V(F)$ be a projective plane curve and let P be a regular point on $V(F)$. Show that the projective closure of the tangent line of $V(F)$ at P is

$$V\left(\frac{\partial F}{\partial x}(P)\cdot x+\frac{\partial F}{\partial y}(P)\cdot y+\frac{\partial F}{\partial z}(P)\cdot z\right).$$

6 Projective varieties

In this section we want to generalise the results from Section 5 to arbitrary projective varieties. We will not be able to treat this Section in the lectures of this summer school, however. Consider it as an additional reading which may complete the picture on basic facts about algebraic geometry introduced throughout this course.

In this section we set $\mathbf{x} = (x_0, \ldots, x_n)$ and $K[\mathbf{x}] = K[x_0, \ldots, x_n]$.

A) The projective n-space

Definition 6.1 (Projective n-space)
We define on the set $K^{n+1} \setminus \{0\}$ an equivalence relation by setting

$$p \sim q \iff \exists\, 0 \neq \lambda \in K \,:\, q = \lambda \cdot p.$$

The equivalence class of a point $p = (p_0, \ldots, p_n)$ is denoted by

$$P = (p_0 : \ldots : p_n),$$

and we call the p_i the *homogeneous coordinates* of P. Moreover, the set of equivalence classes

$$\mathbb{P}^n_K = \left\{ (p_0 : \ldots : p_n) \;\middle|\; (p_0, \ldots, p_n) \in K^{n+1} \setminus \{0\} \right\}$$

is the *projective n-space*.

Remark 6.2 (The points of projective n-space)
The points in projective n-space are by definition *lines in* K^{n+1} *through the origin* omitting the origin. We will, however, for simplicity call them lines through the origin.

Definition 6.3 (Affine charts)
For $i = 0, \ldots, n$ we call the injective map

$$\phi_i : \mathbb{A}^n_K \longrightarrow \mathbb{P}^n_K : (p_1, \ldots, p_n) \mapsto (p_1 : \ldots : p_{i-1} : 1 : p_i : \ldots : p_n)$$

the i-*th affine chart* of $\mathbb{P}^n_K$. The image of ϕ_i is denoted by

$$U_i = \mathrm{Im}(\phi_i) = \{(p_0 : \ldots : p_n) \mid p_i \neq 0\}.$$

Remark 6.4 (Affine charts and the decomposition of $\mathbb{P}^n_K$)
The affine charts can be used to identify affine n-space with certain subsets of projective n-space. At the same time it follows from

$$\mathbb{P}^n_K = U_0 \cup \ldots \cup U_n$$

that projective n-space can be covered by $n+1$ copies of affine n-space. Finally, if we identify U_0 with $\mathbb{A}^n_K$ and note

$$\mathbb{P}^n_K \setminus U_0 = \{(0 : p_1 : \ldots : p_n) \mid (p_1 : \ldots : p_n) \in \mathbb{P}^{n-1}_K\} \hat{=} \mathbb{P}^{n-1}_K$$

we see that

$$\mathbb{P}^n_K \widehat{=} \mathbb{A}^n_K \cup \mathbb{P}^{n-1}_K$$

is a disjoint union of a copy of affine n-space and of projective $n-1$-space.

B) **Projective algebraic varieties**

Remark 6.5 (Homogeneous polynomials)
A polynomial $F \in K[\mathbf{x}]$ of degree d is called *homogeneous* if for $0 \neq \lambda \in K$

$$F(\lambda \cdot x_0, \ldots, \lambda \cdot x_n) = \lambda^d \cdot F(x_0, \ldots, x_n), \tag{4}$$

or equivalently if all monomials of F have degree d. Note that (4) implies that

$$F(p) = 0 \iff F(\lambda \cdot p) = 0 \quad \forall\, 0 \neq \lambda \in K,$$

and we may define $F(P) = 0$ for $P \in \mathbb{P}^n_K$ if $F(p) = 0$ for some representative p of P.

Definition 6.6 (Homogeneous ideals)
If $f \in K[\mathbf{x}]$ is any polynomial of degree d then we can write

$$f = f_0 + f_1 + \ldots + f_d$$

where f_i is homogeneous of degree i, by just collecting in f_i all terms of f of degree d. We call the f_i the *homogeneous parts* of f.
An ideal $I \trianglelefteq K[\mathbf{x}]$ is called *homogeneous* if it contains for any polynomial f also its homogeneous parts.

Remark 6.7 (Homogeneous ideals)
It is easy to see that an ideal is homogeneous if and only if it is generated by homogeneous polynomials. Moreover, if the homogeneous generators of a homogeneous ideal vanish at a point $P \in \mathbb{P}^n_K$ then actually all (not necessarily homogeneous) polynomials in the ideal do so as well, independently of the chosen homogeneous coordinates. This justifies the following definition.

Definition 6.8 (Projective algebraic varieties)
For a homogeneous ideal $I \trianglelefteq K[\mathbf{x}]$ we call

$$V(I) = \{P \in \mathbb{P}^n_K \mid f(P) = 0 \ \forall\, f \in I\}$$

the *projective vanishing set* of I. And we call a subset $X \subseteq \mathbb{P}^n_K$ a *projective algebraic variety* if it is the projective vanishing set of some homogeneous ideal.

Remark 6.9 (Cones over projective algebraic varieties)
One should note that for a homogeneous ideal $I \trianglelefteq K[\mathbf{x}]$ the notion $V(I)$ is used for both, the projective algebraic variety

$$V(I) = \{P \in \mathbb{P}^n_K \mid f(P) = 0 \ \forall\, f \in I\}$$

as well as the affine algebraic variety

$$V(I) = \{p \in \mathbb{A}^{n+1}_K \mid f(p) = 0 \ \forall\, f \in I\}.$$

The latter is often called the *affine cone* over the corresponding projective algebraic variety, and one has to understand from the context which one is meant.

Since the intersection and the sum of homogeneous ideals is again homogeneous the Zariski topology generalises to projective n-space.

Proposition 6.10 (Zariski topology on $\mathbb{P}^n_K$)
The collection of all projective algebraic varieties in $\mathbb{P}^n_K$ *is a topology.*

Remark 6.11
Given a polynomial $f \in K[x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$ of degree d we define its *homogenisation* with respect to x_i as

$$f_i^h = x_i^d \cdot f\left(\tfrac{x_0}{x_i}, \ldots, \tfrac{x_{i-1}}{x_i}, \tfrac{x_{i+1}}{x_i}, \ldots, \tfrac{x_n}{x_i}\right) \in K[x_0, \ldots, x_n]$$

which is homogeneous of degree d.
For a homogeneous polynomial $F \in K[x_0, \ldots, x_n]$ we define its *dehomogenisation* with respect to x_i as

$$F_i^d = F(x_0, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) \in K[x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n],$$

which is a not necessarily homogeneous polynomial of degree at most d.
For an ideal I in $K[x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$ we define its *homogenisation* I_i^h w.r.t. x_i to be the ideal generated by the homogenisations of all elements in I, and for a homogeneous ideal I in $K[x_0, \ldots, x_n]$ we define its *dehomogenisation* I_i^d w.r.t. x_i to be the set of dehomogenisations of elements in I.

Proposition 6.12

a. *If* I *is an ideal in* $K[x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$ *and* $X = V(I) \subseteq \mathbb{A}^n_K$*, then the topological closure of* $\phi_i(X)$ *in* $\mathbb{P}^n_K$ *is the vanishing set of* I_i^h*.*

b. *If* I *is a homogeneous ideal in* $K[x_0, \ldots, x_n]$ *and* $X = V(I) \subseteq \mathbb{P}^n_K$*, then* $\phi_i^{-1}(X) = V(I_i^d) \subseteq \mathbb{A}^n_K$*.*

Corollary 6.13
The affine charts $\phi_i : \mathbb{A}^n_K \hookrightarrow \mathbb{P}^n_K$ *are continuous.*

C) **The projective Nullstellensatz**

Remark 6.14
The ideals $\langle x_0, \ldots, x_n \rangle$ and $K[\mathbf{x}]$ are both homogeneous radical ideals and their projective vanishing set is empty. But this is the only ambiguity of this sort when considering a projective version of Hilbert's Nullstellensatz. Note that any homogeneous ideal which is not the whole ring is contained in the ideal $\langle x_0, \ldots, x_n \rangle$.

Analogously to the affine case we can define a *homogeneous vanishing ideal*

$$I(X) = \langle F \in K[\mathbf{x}] \mid F \text{ is homogeneous}, F(P) = 0 \ \forall\ P \in X \rangle$$

for a subset $X \subseteq \mathbb{P}^n_K$, and $I(X)$ will be a radical ideal. Moreover, we call the graded K-algebra

$$K[\mathbf{x}]/I(X)$$

the *coordinate ring* of the projective algebraic variety X, if X is a projective algebraic variety.

Theorem 6.15 (Projective Nullstellensatz)
Let $I \lhd K[\mathbf{x}]$ *be a strict homogeneous ideal and* $K = \overline{K}$ *then*

$$I(V(I)) = \sqrt{I}.$$

In particular, there is a one-to-one correspondence between the projective algebraic varieties in $\mathbb{P}^n_K$ *and the strict homogeneous radical ideals in* $K[\mathbf{x}]$.

Remark 6.16
One defines the notion of *irreducibility* for projective algebraic varieties analogous to the same notion for affine algebraic varieties, and it turns out that a projective algebraic variety is irreducible if and only if its vanishing ideal is prime. Also, the associated prime ideals of $I(X)$ give the *irreducible components* of X as in the affine case. The *dimension* of a projective algebraic variety can be defined analogously to that of an affine algebraic variety, and if $K = \overline{K}$ then $\dim(X) = \dim K[X] - 1$, i.e. the dimension of the variety coincides with that of its coordinate ring minus one. The minus one reflects the fact that a point in projective space is line in affine space.

D) **Regular functions and morphisms on projective algebraic varieties**

Definition 6.17 (Regular functions and morphisms)
Let $X \subseteq \mathbb{P}^n_K$ and $Y \subseteq \mathbb{P}^m_K$ be two projective algebraic varieties and $U \subseteq X$ be open.

a. A function $f : U \longrightarrow K$ is *regular* if and only if f is locally at each point of U the quotient of two homogeneous polynomials of the same degree. We again denote by $\mathcal{O}_U(X)$ the K-algebra of regular functions on U.

b. A morphism from X to Y is a continuous map such that the pull back of a regular function is a regular function.

Remark 6.18 (Regular functions on projective algebraic varieties)
For a projective algebraic variety the elements of $K[X]$ are in general not regular functions, since we cannot evaluate polynomials on projective space as we have seen above. Actually, there are not many *global* regular functions on a projective algebraic variety, as the following theorem shows – the constant functions are the only global regular functions! But open subsets, such as the affine charts $U_i \cong \mathbb{A}^2_K$, may have plenty of regular functions – e.g. $\mathcal{O}_{\mathbb{P}^n_K}(U_i) \cong K[x_0, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$.

Theorem 6.19 (Global regular functions on projective algebraic varieties)
If $X \subseteq \mathbb{P}^n_K$ *is an irreducible projective algebraic variety and* $K = \overline{K}$, *then* $\mathcal{O}_X(X) = K$.

Remark 6.20 (Morphisms of projective algebraic varieties)
While even homogeneous polynomials are no regular functions on projective algebraic varieties, they can be used to define morphisms. If $F_0, \ldots, F_m \in K[x_0, \ldots, x_n]$ are *homogeneous* polynomials of the *same degree* and if $X \subseteq \mathbb{P}^n_K$ is a projective algebraic variety such $V(F_0, \ldots, F_m) \cap X = \emptyset$, then

$$\varphi : \mathbb{P}^n_K \longrightarrow \mathbb{P}^m_K : P \mapsto \big(F_0(P) : \ldots : F_m(P)\big)$$

is a morphism. This is the most common way to write down a morphism on a projective algebraic variety, but we should like to point out that not every morphism of projective algebraic varieties has such a global description.

Theorem 6.21 (Projective morphisms are closed.)
If $\varphi : X \longrightarrow Y$ *is a morphism of projective algebraic varieties, then the image of every closed subset of* X *is closed in* Y*.*

Example 6.22
The analogous statements for affine algebraic varieties is wrong, as the image of $V(xy - 1) \subseteq \mathbb{A}^2_K$ under the projection to the x-axis is not closed in $\mathbb{A}^1_K$.

Example 6.23 (Computing the image of a projective morphism)
Since the image of a projective morphism is closed, the method in Remark 3.22 applied to the projective situation actually computes the image of the projective morphism and no topological closure is needed. E.g. we can compute the image of the 2-uple Veronese embedding of $\mathbb{P}^2_K$

$$\mathbb{P}^2_K \longrightarrow \mathbb{P}^6_K : (x : y : z) \mapsto (x^2, y^2, z^2, xy, xz, yz)$$

which uses the monomials of degree two as component functions.

```
> ring r=0,(x,y,z,a(0..5)),dp;
> ideal I=a(0)-x2,a(1)-y2,a(2)-z2,a(3)-xy,a(4)-xz,a(5)-yz;
> eliminate(I,xyz);
_[1]=a(3)*a(4)-a(0)*a(5)
_[2]=a(1)*a(4)-a(3)*a(5)
_[3]=a(2)*a(3)-a(4)*a(5)
_[4]=a(1)*a(2)-a(5)^2
_[5]=a(0)*a(2)-a(4)^2
_[6]=a(0)*a(1)-a(3)^2
```

E) **The Hilbert polynomial of a projective algebraic variety**

Remark 6.24 (The Hilbert function)
The coordinate ring $K[X] = K[\mathbf{x}]/I(X)$ of a projective algebraic variety X is a K-vector space, and we denote by $K[X]_d$ the subspace generated by the residue classes

of all monomials of degree d. The function

$$H_X : \mathbb{Z} \longrightarrow \mathbb{Z} : d \mapsto \dim_K K[X]_d$$

is the *Hilbert function* of X. E.g. for $X = \mathbb{P}^n_K$ we have

$$H_{\mathbb{P}^n_K}(d) = \binom{d+n}{n}.$$

Theorem 6.25 (Hilbert polynomial)
If X is a projective variety, then there is a polynomial $HP_X \in \mathbb{Q}[t]$ *of degree* $\dim(X)$ *such that*

$$HP_X(d) = H_X(d)$$

for d sufficiently large. We call HP_X *the* Hilbert polynomial *of* X.

Remark 6.26 (The degree of a projective algebraic variety)
The Hilbert polynomial HP_X of a projective algebraic variety X of dimension d is of the form

$$HP_X(t) = a_d \cdot t^d + a_{d-1} \cdot t^{d-1} + \ldots + a_0 \in \mathbb{Q}[t]. \tag{5}$$

We call the number

$$\deg(X) = d! \cdot a_d \in \mathbb{Z}$$

the *degree* of the variety X, and it turns out that this number is always a positive integer. In the case that $X = V(F)$ for some squarefree homogeneous polynomial F, then

$$\deg(X) = \deg(F).$$

Example 6.27 (Computing the Hilbert polynomial)
The Hilbert polynomial of a projective algebraic variety can be computed with the SINGULAR command `hilbPoly`. It returns an integer vector containing the entries

$$d! \cdot a_0, d! \cdot a_1, \ldots, d! \cdot a_d$$

where we use the notation in (5). E.g. the so called *twisted cubic* is the image of the parametrisation

$$\mathbb{P}^1_K \longrightarrow \mathbb{P}^3_K : (s : t) \mapsto (s^3 : s^2t : st^2 : t^3).$$

We want to show that the result is actually a space curve of degree 3 as the name suggests.

```
> ring R=0,(w,x,y,z,s,t),dp;
> ideal J=w-s3,x-s2t,y-st2,z-t3;
> ideal I=eliminate(J,st);
> ring r=0,(w,x,y,z),dp;
> ideal I=imap(R,I);
> I;
I[1]=y2-xz
I[2]=xy-wz
```

```
I[3]=x2-wy
> hilbPoly(I);
1,3
> dim(groebner(I));
2
```

We first compute the ideal of the image of the parametrisation in the ring $K[w,x,y,z,s,t]$ and we then map with the command `fetch` to the ring $K[w,x,y,z]$. There we compute the Hilbert polynomial and get

$$\mathrm{HP}_X = \frac{3}{1!}\cdot t + \frac{1}{1!} = 3t+1.$$

This shows in particular that $X = V(I)$ is a curve in projective 3-space of degree 3. We finally verified that the dimension is 1 by computing the dimension of the coordinate ring which should be one more than the dimension of the projective algebraic variety.

F) **The Theorem of Bézout**

Definition 6.28 (Pure-dimensional)
We call a projective algebraic variety *pure-dimensional* if all its irreducible components have the same dimension.

Remark 6.29 (Intersection multiplicity)
One can generalise the notion of intersection multiplicity of two curves to higher dimensions. If a point p is an irreducible component of the intersection $X\cap Y$ of two *affine* algebraic varieties in $\mathbb{A}^n_K$ then we define

$$\mathrm{mult}_p(X\cap Y) = \dim_K K[x_1,\ldots,x_n]/I(X)+I(Y).$$

In the projective case one passes to some affine chart which contains the point.

Theorem 6.30 (Bézout)
Let $X,Y\subseteq\mathbb{P}^n_K$ be two pure-dimensional projective varieties over $K=\overline{K}$ such that no component of X is contained in Y and vice versa.

a. *If* $\dim(X)+\dim(Y)=n$, *then* $X\cap Y$ *is a finite set and*

$$\sum_{P\in X\cap Y}\mathrm{mult}_P(X\cap Y) = \deg(X)\cdot\deg(Y).$$

b. *If* $\dim(X)+\dim(Y)=n+1$, *then* $X\cap Y$ *is a curve of degree*

$$\deg(X\cap Y) = \deg(X)\cdot\deg(Y).$$

Corollary 6.31 (The degree of a projective algebraic variety)
The degree of a projective algebraic variety X over $K=\overline{K}$ is the number of intersection points with a generic linear space of dimension $n-\dim(X)$.

Example 6.32
Intersecting $V(wz - xy)$ with $V(w^3 + x^3 + y^3 + z^3)$ in $\mathbb{P}^3_{\mathbb{C}}$ we get a space curve of degree 6.

```
> ring r=0,(w,x,y,z),dp;
> poly F=wz-xy;
> poly G=w3+x3+y3+z3;
> LIB "poly.lib";
> hilbPoly(ideal(F,G));
-3,6
```

Intersecting $V(w^3 + x^3 + y^3 + z^3)$ with a generic line in $\mathbb{P}^3_{\mathbb{C}}$ we should find three intersection points. A generic line is given by two random linear polynomials, but in order to use the command `solve` in SINGULAR we have to concentrate on one of the affine charts, e.g. we can add the polynomial $z - 1$ to the ideal — if the linear polynomials were chosen randomly then no intersection point should occur on one of the coordinate hyperplanes.

```
> ring r=0,(w,x,y,z),dp;
> poly F=w3+x3+y3+z3;
> ideal H=3w+x+2y-12z,13w-x-y+21z;
> ideal I=F,H,z-1;
> LIB "solve.lib";
> solve(I);
[1]:
   [1]:
      -1.61509409
   [2]:
      -16.8377286
   [3]:
      16.84150544
   [4]:
      1
[2]:
   [1]:
      (-0.88699321+i*0.12160413)
   [2]:
      (4.27719681+i*3.52651968)
   [3]:
      (5.19189141-i*1.945666)
```

```
   [4]:
      1
[3]:
   [1]:
      (-0.88699321-i*0.12160413)
   [2]:
      (4.27719681-i*3.52651968)
   [3]:
      (5.19189141+i*1.945666)
   [4]:
      1
```

Appendix A Short introduction to Singular

This short introduction to the computer algebra system Singular does not claim to be complete. It introduces step by step basic structures and commands in Singular. The introduction is not written in a strictly systematic manner. Therefore, for a systematical and complete documentation of Singular, we refer to the manual [DGPS10]. Anyone wishing to install Singular on their personal computer can find the sources and the installation instructions on the Singular home page:

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

1) First steps

1.1 *Notations.* The following notations will be used in this introduction:

- Singular input and output as well as set words will be written in typewriter face, e.g. `exit;` oder `help`.
- The symbol $\mapsto$ starts Singular output, e.g.:

  ```
  int i=5;
  i;
  ↦ 5
  ```

- Square brackets mark the parts of the syntax which are optional, that is, can be left out. E.g.

  ```
  pmat(M[,n]);
  ```

  The above command, a procedure of the library `matrix.lib` is used to show a matrix M as a formatted matrix. The optional parameter `n` defines the width of the columns. If this is missing, a standard value will be used.
- Keys are also shown in typewriter face, such as:
  `n` (press the key `n`),
  `RETURN` (press the enter key),
  `CTRL-P` (press the control key and `P` simultaneously).

1.2 *Starting and terminating* Singular. Obviously, the first question is, how does one start the programme and how can it be terminated? Singular is started by using the command

`Singular`

in the command line of the system.

After the start, Singular shows an input prompt, >, and is available to the user for interactive use. As soon as the user no longer wants to use this possibility, it is recommended to terminate the programme. There are three commands available for this: `exit;`, `quit;` or, for very lazy users, $.

Please note that the semicolons in the preceding paragraph are part of the Singular commands.

In general, *every* command in Singular ends with a semicolon!

The semicolon tells the computer that the input is to be *interpreted* and, if this is successful, be *carried out*. The programme comes up with a result (possibly an error notification) followed by a new input prompt. Should the user forget the semicolon, SINGULAR shows this with an input prompt ., in words a dot, and enables further inputs, such as the missing semicolon. In this way it is possible to stretch longer commands over several lines.

1.3 *The online help* `help`. The next most important information after the start and terminate commands is how to find help. Here SINGULAR offers the command `help`, or in short `?`. Using the command `help` followed by a SINGULAR command, a SINGULAR function or procedure name or a SINGULAR library, information to the respective objects are shown. For the libraries one receives a list of the procedures contained therein, for commands, functions and procedures their purpose is explained as well as their syntax and one gets examples.

Examples:

```
help exit;
help standard.lib;
help printf;
```

By default an internet browser will be opened and the help will be displayed. Via self–explanatory buttons the entire handbook is available.

1.4 *Interrupt* SINGULAR. Under Unix–like operating systems and under Windows, it is possible, via the key combination `CTRL-C`, to force an interruption in SINGULAR. SINGULAR reacts with an output of the currently performed command and awaits further instructions. The following options are available:

`a`	SINGULAR carries out the current command and returns then to top level,
`c`	SINGULAR carries on,
`q`	the programme SINGULAR is terminated.

1.5 *Editting inputs.* If a command has been misspelled, or if an earlier command is needed again, it is not absolutely necessary to renew the input. Existing SINGULAR text can be edited. For this, SINGULAR records a history of all commands of a SINGULAR session. Below is a selection of the available key combinations for text editing:

`TAB`	automatic completion of function and file names
←	
`CTRL-B`	moves the cursor to the left
→	
`CTRL-F`	moves the cursor to the right

CTRL-A	moves the cursor to the beginning of the line
CTRL-E	moves the cursor to the end of the line
CTRL-D	deletes the letter under the cursor — never use in an empty line!
BACKSPACE	
DEL	
CTRL-H	deletes the letter in front of the cursor
CTRL-K	deletes all from the cursor to the end of the line
CTRL-U	deletes all from the cursor to the beginning of the line
↓	
CTRL-N	supplies the next line from the history
↑	
CTRL-P	supplies the preceding line from the history
RETURN	sends the current line to the SINGULAR parser

1.6 *Procedures.* The user can create new commands in SINGULAR. These are called procedures. The syntax of a procedure is fairly simple:

```
proc PROCEDURENAME [PARAMETERLIST]
{
  PROCEDUREBODY
}
```

For `PROCEDURENAME`, any not otherwise resevered sequence of letters can be used. The types and names of the arguments which are passed on to the procedure are laid down in the `PARAMETERLIST`. The `PARAMETERLIST` should be encased in round brackets. The `PROCEDUREBODY` contains a sequence of SINGULAR code. If the procedure is to return a result, the result should be stored in a variable `result` and the procedure should terminate with the command `return(result);`.

An example is more useful than thousands of words:

```
proc permcol (matrix A, int c1, int c2)
{
  matrix B=A;
  B[1..nrows(B),c1]=A[1..nrows(A),c2];
  B[1..nrows(B),c2]=A[1..nrows(A),c1];
  return(B);
}
```

The procedure `permcol` should exchange two columns of a matrix. For this three arguments are necessary. The first argument of name `A` is of type `matrix`, the two following arguments `c1` and `c2` are of type `int`. SINGULAR instructions follow and the result is stored in the variable `B` of type `matrix`, which is then returned with `return(B);`. This means, in particular, that the result of the procedure is of type `matrix`.

A procedure can be invoked by entering the procedure name, followed by the arguments in round brackets. E.g.

```
LIB "matrix.lib"; LIB "inout.lib";
ring r=0,(x),lp;
matrix A[3][3]=1,2,3,4,5,6,7,8,9;
pmat(A,2);
↦ 1  2  3
   4  5  6
   7  8  9

matrix B=permcol(A,2,3);
pmat(B,2);
↦ 1  3  2
   4  6  5
   7  9  8
```

Variables, which are defined within a procedure, are only known there and may, therefore, have the same name as objects which are defined outside the procedure.

1.7 *Libraries.* To make procedures available for more than one SINGULAR session, it makes sense to store them in files, which can be loaded as SINGULAR libraries. The library names always have the ending `.lib`. Libraries are read into SINGULAR through the command `LIB` followed by library name enclosed in `"`, such as

```
LIB "123456.lib";
```

(Library names should, if possible, only consist of *eight* letters, to guarantee compatibility with systems such as Dos!) If they are not builtin SINGULAR libraries, then they should be in the subdirectory from which SINGULAR is started.

Of course, a library must conform to certain syntax rules, and procedures, which are stored in libraries, should be extended by two explanatory additions. We show this in an example:

```
////////////////////////////////////////////////////////////////
version="1.0";

info="
     LIBRARY:     linalgeb.lib FIRST STEPS IN LINEAR ALGEBRA
     AUTHOR:      Thomas Markwig, email:  keilen@mathematik.uni-kl.de
     PROCEDURES:
       permcol(matrix,int,int)  permutes columns of the matrix
       permrow(matrix,int,int)  permutes rows of the matrix
";
////////////////////////////////////////////////////////////////
LIB "inout.lib";
```

```
////////////////////////////////////////////////////////////////
proc permcol (matrix A, int c1, int c2)
 "USAGE:   permcol(A,c1,c2); A matrix, c1,c2 positive integers
 RETURN:   matrix, A being modified by permuting column c1 and c2
 NOTE:     space for important remark
           can be stretched over several lines
 EXAMPLE:  example permcol; shows an example"
{
  matrix B=A;
  B[1..nrows(B),c1]=A[1..nrows(A),c2];
  B[1..nrows(B),c2]=A[1..nrows(A),c1];
  return(B);
}
example
{
"EXAMPLE:";
echo = 2;
ring r=0,(x),lp;
matrix A[3][3]=1,2,3,4,5,6,7,8,9;
pmat(A);
pmat(permcol(A,2,3));
}
⋮
```

If a double slash `//` in a line appears, the rest of the line is interpreted as a comment and ignored.

The first section is the head of the library. The first line contains the reserved name `version`, through which the version number of the library is fixed. General information to the library follows the reserved name `info`.

It should be noted that under the item `PROCEDURES:` all procedure names are listed with a one-line description.

SINGULAR shows this part when the help command is called on the library, that is

```
help linalgeb.lib;
```

It should also be noted that strings are allocated to `version` and `info` by means of the sign of equality, `=`, so that the `"` are just as necessary as the semicolon at the end of the line!

Section two serves the loading of further libraries, whose procedures one wants to use. As an example, the library `inout.lib` is loaded, whose procedure `pmat` is used in the `example` part of the procedure `permcol`.

In the third section the procedures follow one by one. (It should be noted that the command `proc` always has to be at the start of a new line!)

It is recommended that the Syntax in section 1.6 is extended by two sections for procedures. A commentary block can be inserted between the procedure head and body, enclosed in ", which contains certain key words followed by the relative information. Under `USAGE:` should be shown how the command is invoked and of which type the arguments are. `RETURN:` should contain information on the type of the return value and, if necessary, further information. `NOTE:` is used to show important comments to the procedure, its use, etc. `EXAMPLE:` shows how an example of the use of the procedure can be displayed in SINGULAR. The commentary block contains the information which is shown when the help command is called for the procedure, e.g. through

```
help permcol;
```

The second additional section at the end of the procedure is initiated through the reserved name `example`, followed by a section in curly brackets which contains the SINGULAR code. The aim is to show an example for the operation of the procedure which explains its use to the user. The user obtains the example by entering `example PROCEDURENAME;`.

1.8 *Write to files / read from files.* The command `write` offers the possibility to store the values of variables or any string in a file. For this, the variable values are converted to strings. The following lines store variable values, resp. a string, in the file `hello.txt`:

```
int a=5;
int b=4;
write("hello.txt",a,b);
write("hello.txt","This is Singular.");
```

Several variables or strings can be stored at a time, separated by commas. The value of each variable is written in a separate line.

Data contained in a file can be read in by the command `read`. They are, however, interpreted as strings, e.g.

```
read("hello.txt");
↦ 5
  4
  This is Singular.
```

Should SINGULAR code, which is read in from a file, be recognised as such, then the `read` command must be passed on to the command `execute`. If the file `hello.txt` contains the following lines,

```
4*5-3;
6/3;
```

then the command

```
execute(read("hello.txt"));
```

leads to the following SINGULAR output:

```
↦  17
   2
```

A short form for `execute(read(...))` is <, e.g.

```
< "hello.txt";
```

Anyone wanting to document a SINGULAR session for security in a file, e.g. `hello.txt`, can do this with the command `monitor`, e.g.

```
monitor("hello.txt","io");
```

The option `"io"` causes input as well as output to be stored. The omission of one of the letters leads to only the input or only the output being stored. The option `monitor` is very helpful when working on an operating system on which SINGULAR is instable.

Please note that `monitor` opens a file, but does not terminate it. This can be done by the following input:

```
monitor("");
```

2) **Types of data in Singular and rings**

In SINGULAR different data types are available, and when introducing a variable first one has to specify the data type of the variable. Most data types in SINGULAR depend on a meta structure, the base ring, over which they exist. Exceptions are `string`, `int`, `intvec` und `intmat`. To perform a computation in SINGULAR it is first absolutely necessary to define the ring over which one is working.

`ring r=0,x(0..2),lp;`	The ring of polynomials in the variables $x(0), x(1), x(2)$ with coefficients in the rational numbers $\mathbb{Q}$ and global lexicographical ordering.
`ring r=(0,a,b),(x,y,z),dp;`	The ring of polynomials in the variables x, y, z, where the coefficients are rational functions in the variables a and b. The global degree reverse lexicographical ordering is used.
`ring r=(real,15),x,ls;`	The localised ring of polynomials in the variables x with coefficients in real numbers $\mathbb{R}$ — for computations with 15 places after the decimal point. The local lexicographical ordering is used.
`ring r=5,x,ds;`	The localised ring of polynomials in the variables x with coefficients in $\mathbb{Z}/5\mathbb{Z}$ and local degree reverse lexicographical ordering.

A list of the available data types in SINGULAR is given below.

`int i=1;`	The data type `int` represents the machine integers (between -2^{31} und $2^{31}-1$). In addition, `boolean` values are represented as `integers`, `0 = FALSE`, `1 = TRUE`.
`string s="Hallo";`	`strings` are chains of letters enclosed by ".
`intvec iv=1,2,3,4;`	A vector of `integers`.
`intmat im[2][3]=1,2,3,4,5,6;`	A matrix with two lines and three columns with `integer` entries.
`ring R=(0,a),(x,y),lp;`	$\mathbb{Q}(a)[x,y]$ with lexicographical order.
`number n=4/6;`	`numbers` are the elements of the base field of the ring. By `ring r=0,x,lp;` the rational numbers, by `ring r=(0,a),x,lp;` also fractions of polynomials in a, e.g. $\frac{a^2+1}{a-1}$.
`list l=n,iv,s;`	A list can contain objects of different types. `l[2]` refers to the second entry of `l`.
`matrix m[2][3]=1,2,3,4,5,6;`	A matrix with two lines and three columns, the entries being of type `poly`,
`vector v=[1,2,3];`	A vector in the module R^3.
`proc`	The data type `procedure` is discussed at length in 1.6.
`poly f=x2+2x+1;`	A polynomial in the indeterminates of the ring with `numbers` as coefficients, here $f = x^2+2x+1$. Note that numbers in front of the monomials are interpreted as coefficients, whereas SINGULAR interprets integers after single variables as exponents.
`ideal i=f,x3;`	The ideal generated by f and x^3.
`qring Q=i;`	The quotient ring `R/i`.
`map g=R,x;`	A map from the ring `R` to the current ring sending the first variable of `R` to x.
`module mo=v,[x,x2,x+1];`	The module generated `v` and $(x, x^2, x+1)$.
`def j;`	In case one does not want to specify the data type yet, one can use the type `def`. The first time a value is assigned to `j` this value determines the data type of `j`.
`link`	For the data type `link`, we refer to the manual [DGPS10].
`resolution`	For the data type `resolution`, we refer to the manual [DGPS10].

At the first glance it might seem as though the matrices `im` and `m` are identical. For SINGULAR that is not the case as they are of different types!

3) **Some elements of the programming language Singular**

3.1 *Allocations.* In SINGULAR the operator `=` is used to assign a value to a variable. It is possible to assign a value at the time of the definition of the variable,

```
int i=1;
```

or later,

```
int i;
⋮
i=2;
```

3.2 *Loops.* There are two types of loops, the `for` and the `while` loop.

The `for` loop is used typically, if a command sequence is to be performed several times and the number of times is known beforehand. E.g.

```
int s=0;
int i;
for (i=1; i<=10; i=i+1)
{
  s=s+i;
}
```

The command sequence in curly brackets are the commands executed when passing the loop. The commands in round brackets determine how often the loop is to be passed. The first entry fixes the control variable and is here of type `int`; the second entry shows the termination condition, i.e. the loop is passed through as long as this condition is fulfilled; the third entry fixes how the control variable should change in each passage. The example computes the sum of the first ten natural numbers.

`while` loops are used, when the number of passages is not a priori clear. E.g.

```
int s=10000;
int i=1;
while (s > 50)
{
  i=i*i;
  s=s-i;
}
```

Again the command sequence is shown in curly brackets, whilst the termination conditions are shown in round brackets. As long as these show the value `TRUE`, the loop is performed.

The termination condition is checked before the first entry into the loop.

3.3 *Branchings.* SINGULAR offers as a branching the `if-else` command, where, however, the `else` part could be missing. E.g.

```
int i=10;
int s=7;
if (i<5 or s<10)
{
  s=5;
}
else
{
  s=0;
}
```

Again the command sequences are shown as a block in curly brackets, whereas the branching conditions are in round brackets.

3.4 *Comparison operators.* In SINGULAR we have the comparison operators `==` and `!=`, with which objects of the same type (e.g. `int`, `string`, `matrix`, etc.) can be compared to one another. `==` tests for equality and supplies the value `1` if the objects are the same and otherwise `0`. `!=` checks for inequality. `<>` has the same effect.

For the data types `int`, `number`, `poly` and `vector`, the operators `<`, `>`, `<=` and `>=` are available. Its significance for `integers` and `monomials` is clear. We refer to the manual for further data types [DGPS10].

3.5 *Some further operators in* SINGULAR. As we have already seen, the operators may depend on the data types.

`boolean:` For `boolean` variables, the connecting operators `and` and `or` as well as the negating operator `not` are defined.

```
not ((1==0) or (1!=0));
↦ 0
```

`int:` For `integers` the operations `+`, `-` and `*` are entirely clear. `^` means raising to some power

```
int i=4;
i^3;
↦ 64
```

The commands `div` and `mod` are more difficult, whereby the first is synonymous to `/`. If, for two integers, a division with remainder is performed, then `mod` supplies the remainder, and `div` the result without remainder. E.g. $7 = 2*3+1$, also

```
7 div 3;
↦ 2
7 mod 3;
↦ 1
```

`list`: The following operators are given for the data type `list`.

- `+` Combines the elements of two lists.
- `delete` Deletes an element from a list, `delete(L,3)` deletes the third element of the list `L`.
- `insert` Inserts an element into a list. `insert(L,4)` inserts the element `4` at the start of the list `L`, `insert(L,4,2)` inserts four into into the second position.

`matrix`: The operators `+`, `-` and `*` are available with their obvious meaning. We show, by examples, how single entries of a matrix, resp. whole lines or columns of a matrix, can by accessed:

```
matrix m[2][3]=1,2,3,4,5,6;
print(m);
↦  1,2,3,
   4,5,6
m[1,2];
↦ 2;
m[1,1..3];
↦ 1 2 3
m[1..2,3];
↦ 3 6
```

4) Some selected functions in Singular

SINGULAR has a quite notable arsenal of functions available, which are, in part, integrated in the SINGULAR core, in part made available via libraries. We only wish to show a small selection of function names, which are useful for computations in linear algebra. Information on their syntax can be found via `help` or in the manual.

4.1 *Functions which are connected to the data type* ***matrix***. `ncols, nrows, print, size, transpose, det`, as functions in the core of SINGULAR. Furthermore there are the functions of the library `matrix.lib`, in particular `permrow, permcol, multrow, multcol, addrow, addcol, concat, unitmat, gauss_row, gauss_col, rowred, colred`. Also the function `pmat` from the library `inout.lib` is interesting.

4.2 *Functions which are connected to the data type* ***int***. `random, gcd, prime` as functions in the core of `Singular`.

5) `ESingular` - or the editor Emacs

There are many editors in which SINGULAR procedures and libraries can be written. On Unix or similar systems the editor emacs (oder Xemacs) should be considered, as it simplifies the entered code through using coloured highlighting of the key words, and they offer many options which simplify editing and error correction.

There is another reason for the recommendation to use Emacs. SINGULAR can be started in a special Emacs mode, as `ESingular`. This means that first the editor Emacs is started and then inside Emacs the programme SINGULAR. The advantage is that apart from the full functionality of the editor Emacs for editing files, a bunch of further options can be made available, which simplify the use — in particular for the inexperienced user, for whom pulldown menu buttons are available. By calling

```
ESingular --emacs=xemacs
```

it is possible to fix the version of Emacs which is to be used, in this case Xemacs.

6) **Exercises**

Exercise A.1
Write a procedure `binomi`, which reads in two natural numbers `n` and `k` and returns the binomial coefficient $\binom{n}{k}$. (Convention, if $k < 0$ or $k > n$, then $\binom{n}{k} = 0$.)

Exercise A.2
Write a procedure `squaresum`, which reads in the natural number `n` and returns the sum of the square numbers $1^2, 2^2, 3^2, \ldots, n^2$.

Exercise A.3
Write a procedure `minimum`, which reads in a vector of natural numbers and returns the minimum of the numbers.

Exercise A.4
Write a procedure `rowsumnorm`, `maximumnorm` and `q_eukl_norm`, which read in a $(m \times n)$ matrix `A` of real numbers and calculate

a. the row summmation norm of `A` (i.e. $\max_{i=1,\ldots,m} \left(\Sigma_{j=1}^{n} |A_{ij}|\right)$),

b. the maximum norm of `A` (i.e. $\max\left(|A_{ij}| \;\middle|\; i = 1, \ldots, m,\ j = 1, \ldots, n\right)$), respectively

c. the square of the euclidian norm of `A` (i.e. $\Sigma_{i,j} |A_{ij}|^2$).

Use the function `abs` from the library `linalg.lib` for the absolute value.

7) **Solutions**

Solution to Exercise A.1

```
proc binomi (int n, int k)
"USAGE: binomi(n,k); int n, int k
RETURN: int, binomial coefficient n over k
EXAMPLE: example binomi; shows an example"
{
  if ((k < 0) or (k > n))
  {
   return(0);
```

```
  }
  else
  {
   int i;
   int denominator,nominator1,nominator2 = 1,1,1;
   for (i=1;i<=n;i++)
   {
    denominator = denominator * i;
   }
   for (i=1;i<=k;i++)
   {
    nominator1 = nominator1 * i;
   }
   for (i=1;i<=n-k;i++)
   {
    nominator2 = nominator2 *i;
   }
   return (denominator / (nominator1 * nominator2));
  }
}
example
{
  "Example:";
  echo = 2;
  binomi(5,2);
  binomi(7,5);
}
```

Solution to Exercise A.2

```
proc squaresum (int n)
"USAGE: squaresum(n); int n
RETURN: int, the sum of the first n square numbers
EXAMPLE: example squaresum; shows an example"
{
  if (n < 0)
   {
    return (0);
   }
   else
   {
    int i;
    int result = 0;
```

```
    for (i=1;i<=n;i++)
    {
     result = result + i*i;
    }
    return (result);
   }
}
example
{
  "Example:";
  echo = 2;
  squaresum(3);
  squaresum(5);
}
```

Solution to Exercise A.3

```
proc minimum (intvec iv)
"USAGE: minimum(iv); iv intvector
RETURN: int, the minimum of the entries in iv
EXAMPLE: example minium; shows an example"
{
  int i;
  int k=size(iv);
  int result=iv[1];
  for (i=2;i<=k;i++)
   {
    if (iv[i] < result)
    {
     result=iv[i];
    }
   }
  return(result);
}
example
{
  "EXAMPLE:";
  echo=2;
  intvec iv=3,2,5,2,1;
  print(iv);
  minimum(iv);
  iv =-3,4,5,3,-6,7;
  print(iv);
```

```
  minimum(iv);
}
```

Solution to Exercise A.4

```
proc rowsumnorm (matrix A)
"USAGE: rowsumnorm(A); matrix A with rational/real entries
RETURN: poly, the row-sum-norm of A
EXAMPLE: example rowsumnorm; shows an example"
{
  int i,j;
  int n,m = ncols(A),nrows(A);
  poly r,s = 0,0;
  for (i=1;i<=m;i++)
  {
   for (j=1;j<=n;j++)
   {
    r = r + abs(A[i,j]);
   }
   if (r > s)
   {
    s = r;
   }
   r = 0;
  }
  return (s);
}
example
{
  "Example:";
  echo = 2;
  ring r=real,x,lp;
  matrix A[3][2]=-3,-2,-1,3,-4,2;
  print(A);
  rowsumnorm(A);
  ring r=0,x,lp;
  matrix B[3][2]=-7,0,0,3,-4,2;
  print(B);
  rowsumnorm(B);
}

proc maximumnorm (matrix A)
"USAGE: maximumnorm(A); matrix A with rational/real entries
RETURN: poly, the maximum norm of A
```

```
EXAMPLE: example maximumnorm; shows an example"
{
  int i,j;
  int n,m = ncols(A),nrows(A);
  poly r = 0;
  for (i=1;i<=m;i++)
  {
   for (j=1;j<=n;j++)
   {
    if (abs(A[i,j]) > r)
    {
     r = abs(A[i,j]);
    }
   }
  }
  return(r);
}
example
{
  "Example:";
  echo = 2;
  ring r=real,x,lp;
  matrix A[3][2]=-3,-2,-1,3,-4,2;
  print(A);
  maximumnorm(A);
  ring r=0,x,lp;
  matrix B[3][2]=-7,0,0,3,-4,2;
  print(B);
  maximumnorm(B);
}

proc q_eukl_norm (matrix A)
"USAGE: q_eukl_norm(A); matrix A with rational/real entries
RETURN: poly, the square of the euclidean norm of A
EXAMPLE: example q_eukl_norm; shows an example"
{
  int i,j;
  int n,m = ncols(A),nrows(A);
  poly r = 0;
  for (i=1;i<=m;i++)
  {
   for (j=1;j<=n;j++)
```

```
    {
     r = r + abs(A[i,j]) * abs(A[i,j]);
    }
   }
   return (r);
}
example
{
   "Example:";
   echo = 2;
   ring r=real,x,lp;
   matrix A[3][2]=-3,-2,-1,3,-4,2;
   print(A);
   q_eukl_norm(A);
   ring r=0,x,lp;
   matrix B[3][2]=-7,0,0,3,-4,2;
   print(B);
   q_eukl_norm(B);
}
```

Appendix B First steps with surfex

The programme surfex (see [HL08]) offers a graphical interface to visualise algebraic surfaces in 3-space and curves on such surfaces. One can either start surfex directly or one can invoke it from within a Singular session. If you should want to invoke surfex from within Singular then you have to load the library surfex.lib first. The Singular command to plot a surface would then be plotRot. We demonstrate this with for the polynomial $f = x^2 - y^2 \cdot (t^2 - y)$.

```
                     SINGULAR                                /
A Computer Algebra System for Polynomial Computations    /   version 3-1-2
                                                        0<
by: W. Decker, G.-M. Greuel, G. Pfister, H. Schoenemann  \   Oct 2010
FB Mathematik der Universitaet, D-67653 Kaiserslautern    \
> ring r=0,(t,x,y),dp;
> poly f=x^2-y^2*(t^2-y);
> LIB "surfex.lib";
> plotRot(f);
```

The command plotRot starts the programme surfex and opens three new windows (see Figure 34). The leftmost window contains the equation of the surface as well as most of the control buttons to direct the programme; the upper right window

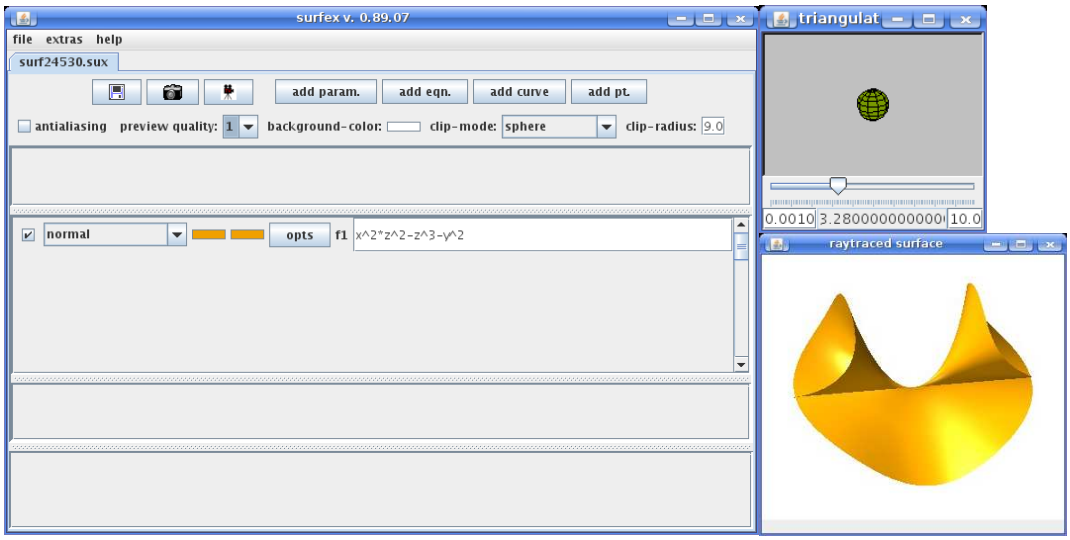


FIGURE 34. Surfex

allows to rotate the surface and to zoom in and out; the third window displays the surface.

If you start `surfex` from within SINGULAR your SINGULAR session is interrupted as long as `surfex` is running. In order to continue in SINGULAR you have to shut down `surfex`. Moreover, you should be aware that your SINGULAR variables will be renamed, so that the first variable becomes x, the second y and the third z. In the above example the variable t becomes x, x becomes y and y becomes z. The polynomial $x^2 - y^2 \cdot (t^2 - y)$ is thus transformed into $y^2 - z^2 \cdot (y^2 - z)$.

As mentioned above one can use `surfex` in order to cut two surfaces and to draw the intersection curve on one of the two surfaces. We show this for the surface $V(y^2 - z^2 \cdot (y^2 - z))$ and the plane $V(x - 1)$. In Figure 35 we added besides the polynomial $y^2 - z^2 \cdot (y^2 - z)$, that could already be seen in Figure 34 the polynomial $x - 1$ by using the button `add eqn`. Both polynomials can be seen in the third part of the control window, and both surfaces are displayed in the left part of Figure 36. There you can also see the intersection curve in black. In order to produce this we have used the button `add curve`. By this we got the option to choose some numbers in the fourth part of the control window. We chose the numbers 1 and 2. This means that on the surface given by the polynomial `f1` the intersection curve that we get by intersecting this surface with the surface given by the polynomial `f2` should be drawn. In the right part of Figure 36 only the surface $V\big(y^2 - z^2 \cdot (y^2 - z)\big)$ and the intersection curve can be seen. This is achieved by removing the hook in front of the polynomial $x - 1$ in the control window of `surfex` (see Figure 35). For further details on the use of `surfex` we refer to the manual [HL08] and to the `surfex` web page:

`http://www.surfex.algebraicsurface.net`

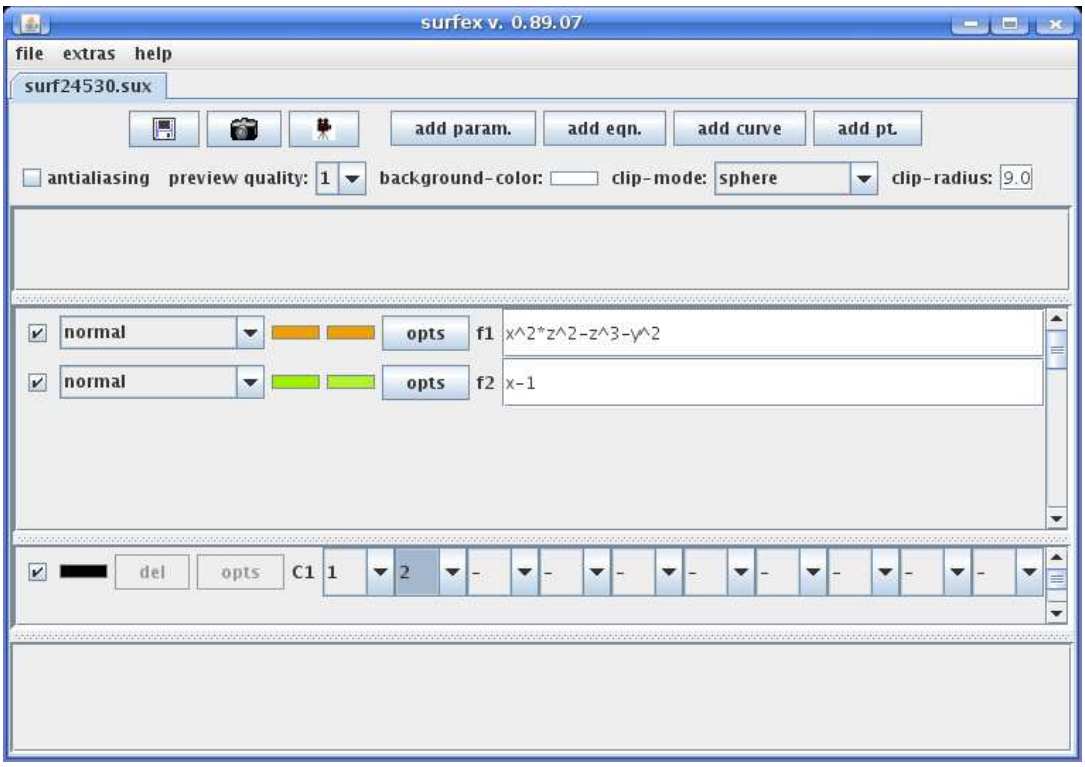


FIGURE 35. The intersection of $V(y^2 - z^2 \cdot (y^2 - z))$ and $V(x - 1)$

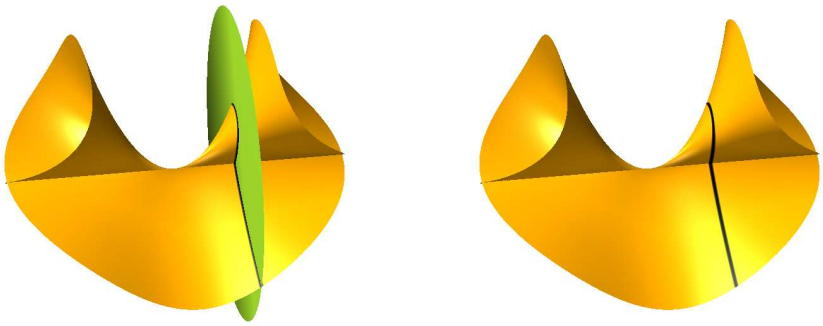


FIGURE 36. The intersection of $V(y^2 - z^2 \cdot (y^2 - z))$ and $V(x - 1)$

REFERENCES

[CLO97] David Cox, John Little, and Donal O'Shea, *Ideals, varieties, and algorithms*, 2nd ed., Springer, 1997.

[DGPS10] Wolfram Decker, Gert-Martin Greuel, Gerhard Pfister, and Hans Schönemann, SINGULAR *3-1-1 — A computer algebra system for polynomial computations*, Tech. report, Centre for Computer Algebra, University of Kaiserslautern, 2010, `http://www.singular.uni-kl.de`.

[Eis96] David Eisenbud, *Commutative algebra with a view toward algebraic geometry*, Graduate Texts in Mathematics, no. 150, Springer, 1996.

[Gat03] Andreas Gathmann, *Algebraic geometry*, Lecture Notes, 2003.

[GP08] Gert-Martin Greuel and Gerhard Pfister, *A* SINGULAR *introduction to commutative algebra*, 2nd ed., Springer, 2008.

[Har77] Robin Hartshorne, *Algebraic geometry*, Springer, 1977.

[Har92] Joe Harris, *Algebraic geometry, a first course*, Graduate Texts in Mathematics, no. 133, Springer, 1992.

[HL08] Stephan Holzer and Oliver Labs, SURFEX 0.90, Tech. report, University of Mainz, University of Saarbrücken, 2008, `www.surfex.AlgebraicSurface.net`.