Toric Geometry and Sage

VOLKER BRAUN

Dublin Institute for Advanced Studies
10 Burlington Road
Dublin 4, Ireland

Email: vbraun@stp.dias.ie

I want to see, retrace and comprehend what my math software does on the inside.

www.sagemath.org

Abstract

I will give a pedagogical introduction to toric geometry without requiring previous knowledge in algebraic geometry. The lecture series will be based on the toric geometry package in the open-source Sage mathematics software system. Various examples relevant to string theory are used to illustrate the techniques. Each lecture will contain exercises to be solved in the accompanying computer lab.

Contents

Part I
Local Structure

1 Affine Varieties

1.1 Affine Algebraic Varieties

Toric varieties are a special kind of algebraic variety (often we say just "variety"), which themselves are a special kind of topological space. So before we get to *toric* part, lets consider just algebraic varieties.

Algebraic geometry (the theory of algebraic varieties) is a wide field with broad applications to geometry, algebra, and number theory. We will actually see applications in these apparently unrelated fields. Unfortunately, its terminology is sometimes unintuitive precisely because it draws from such disparate fields. So instead of starting with the most general definition of an algebraic variety, lets start with the following working definition:

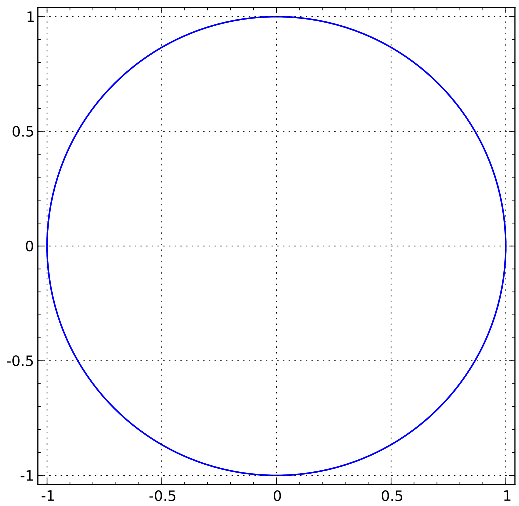


Figure 1: *The 1-dimensional real algebraic variety $x^2 + y^2 = 1$.*

Definition 1 (Affine algebraic variety)**.** *An affine algebraic variety (over the field $\mathbb{F}$) is the zero set $V(p_1, \dots, p_k) \subset \mathbb{F}^n$ of a finite number of polynomials $p_1, \dots, p_k \in \mathbb{F}[x_1, \dots, x_n]$.*

Note the special case where there is no polynomial, $k = 0$. In this case $V(\{\}) = \mathbb{F}^n$, so in particular affine space is an algebraic variety. The non-trivial *affine* algebraic varieties are the subvarieties of affine space cut out by polynomial equations, hence the name. We will almost exclusively consider the case where the field $\mathbb{F} = \mathbb{C}$ are the complex numbers, and this will always be understood in the following whenever we do not specify the field explicitly. However, because they are complex spaces it is generally impossible to draw them on a piece of paper. For graphing purposes, it is rather convenient to use the real numbers $\mathbb{F} = \mathbb{R}$ as the base field. For example, Figure 1 is a picture of a real algebraic variety. But note that the real picture is often misleading. For example, affine algebraic varieties over $\mathbb{C}$ always "run off to infinity":

Exercise 1. *Suppose that that $X = V(p_1, \dots, p_k)$ is generated by $k < n$ non-constant polynomials and that the base field $\mathbb{F}$ is algebraically closed. Show that X admits a surjective map $f : X \to \mathbb{F}^{n-k}$. Conclude that X cannot be compact.*

Another class of base fields is very important for computations, namely finite fields. By definition, a finite field is a field with a finite number of elements. For example, $\mathbb{F}_3 =$

$\{0, 1, 2\}$ with addition and multiplication being the usual addition and multiplication modulo 3 is a field:

- Closed under addition and multiplication
- Multiplication is distributive over addition.
- Both addition and multiplication are associative and commutative.
- Existence of additive identity and inverse.
- Existence of multiplicative identity and inverse.

Usually the last point, that is the existence of a multiplicative inverse for all $x \neq 0$, is the tricky part. Here, we note that $\frac{1}{1} = 1$ and $\frac{1}{2} = 2$ in $\mathbb{F}_3$. This construction generalizes to $\mathbb{F}_p = \mathbb{Z}_p$ for any prime number p.

Exercise 2. *Construct a field with* 4 *elements, that is, construct an addition and multiplication table on* $\{0, 1, x, y\}$.

As the exercise shows, there are more finite fields than just $\mathbb{F}_p$. The general theory of finite fields can be summarized as

Theorem 1 (Structure of finite fields)**.** *The number of elements of a finite field is of the form* p^n*, where* p *is a prime and* $n \geq 1$*. The field of size* p^n *exists and is unique, and is usually denoted by* $\mathbb{F}_{p^n}$*.*

Physicists are generally not interested in finite fields for obvious reasons. But they are an important tool in computations, because one can often devise much faster algorithms for dealing with polynomials over finite fields. For example, see [1, Section 4.6.2] for ways to factorize polynomials over finite fields. Many computations in algebraic geometry are only possible by judiciously replacing the problem over $\mathbb{C}$ with another problem over a finite field, and we will see some examples of this in the following. Note that for any finite field $\mathbb{F}_{p^n}$,

- There are finitely many maps $(\mathbb{F}_{p^n})^k \to \mathbb{F}_{p^n}$.
- There are infinitely many polynomials (in k variables) over $\mathbb{F}_{p^n}$.
- A variety over $\mathbb{F}_{p^n}$ consists of a finite number of points.

1.2 The Ideal of a Variety

While a set of polynomials uniquely determines an algebraic variety, the converse is not true: You cannot uniquely recover the polynomials from the variety. For starters, if p and q are polynomials, then

$$p(\bar{x})f(\bar{x}) + q(\bar{x})g(\bar{x}), \qquad f, g \in \mathbb{C}[x_1, \ldots, x_n] = \mathbb{C}[\bar{x}] \tag{1}$$

also vanishes on $V(p, q)$. So, first of all, we should only think of the variety as depending on the ideal generated by the polynomials, that is, the subset of the whole polynomial ring that is generated by linear combinations with polynomial coefficients:

Definition 2 (Ideal)**.** *The ideal* $\langle r_1, r_2 \ldots, \rangle$ *generated by elements* $r_1, r_2, \cdots \in R$ *of a ring*[1] R *is the set of all* R*-linear combinations*

$$\langle r_1, r_2, \ldots \rangle = \left\{ \sum s_i r_i \middle| s_i \in R \right\} \subset R \tag{2}$$

One key fact about ideals in polynomial rings is that they are finitely generated:

Theorem 2 (Hilbert's basis theorem)**.** *An ideal* $I \subset \mathbb{C}[\bar{x}]$ *is always finitely generated, that is, of the form* $I = \langle p_1(\bar{x}), \ldots, p_k(\bar{x}) \rangle$ *for some* $k \in \mathbb{Z}$*.*

There is some rather unintuitive nomenclature associated; For the record let me mention that

- A ring is called *noetherian* if every ideal is finitely generated; Examples are multivariate polynomial rings over fields.
- A ring[2] is a *principal ideal domain* (PID) if every ideal can be generated by a single element.

Exercise 3. *Given univariate polynomials* f*,* $g \in \mathbb{C}[x]$*, show that* $\langle f, g \rangle = \langle \gcd(f, g) \rangle$*. Conclude that univariate polynomials rings over fields are PIDs.*

We can define the ideal generated by all polynomials vanishing on a variety without having to enumerate any particula set of generators:

Definition 3 (Ideal of a variety)**.** *Given an affine algebraic variety* $V \subset \mathbb{C}^n$*, let*

$$I(V) = \left\{ p \in \mathbb{C}[\bar{x}] \;\middle|\; p(\bar{x}) = 0 \;\forall\; \bar{x} \in V \right\} \tag{3}$$

be the ideal generated by all polynomials vanishing on V*.*

Finding the variety defined by an ideal and the vanishing ideal of a variety are almost, but not quite, inverse operations. What is true is that, for any affine variety V,

$$V\big(I(V)\big) = V. \tag{4}$$

To better understand $I\big(V(I)\big)$, consider the ideal $I = \langle x^2 \rangle \subset \mathbb{C}[x]$. Its variety is $V(I) = \{0\}$, and its vanishing ideal is $\langle x \rangle$. Hence we get a larger ideal than the one we started with; Essentially, if $p^k \in I$ then we have to add p to the generators of $I\big(V(I)\big)$. This construction is called the radical of I and written $\sqrt{I}$, though it involves more than just square-roots. To summarize,

$$I\big(V(I)\big) = \sqrt{I}. \tag{5}$$

[1] By ring, I will always mean a commutative ring.

[2] Some authors also require a PID to be without zero-divisors.

1.3 Dimension

In order to perform any computation, we need to rephrase questions about the geometry of a variety in terms of an algebraic question about the defining ideal. For example, consider the variety defined by the ideal $\langle xy\rangle \subset \mathbb{C}[x, y]$. Clearly, $xy = 0$ if either $x = 0$ or $y = 0$, so the variety is the union of the two coordinate hyperplanes,

$$V\big(\langle xy\rangle\big) = V\big(\langle x\rangle\big) \cup V\big(\langle y\rangle\big) \tag{6}$$

A variety that can be written as the union is called a *reducible* variety. As the example shows, if you can find polynomials not in the ideal but such that their product is in the ideal, then the variety is reducible. This motivates the definition

Definition 4 (Prime ideal)**.** *An ideal $I \subsetneq R$ is prime if $f \cdot g \in I \Rightarrow f \in I$ or $g \in I$.*

The weird name stems from the fact that if you take $R = \mathbb{Z}$, then the ideal $\langle k\rangle$ is a prime ideal if and only if k is a prime number.

$$\boxed{I(V) \text{ is a prime ideal}} \Leftrightarrow \boxed{V \text{ is an irreducible variety}}$$

Perhaps the most basic property is the dimension of the variety. One might be tempted to define the dimension of the variety as (# of variables) − (# of equations), but as we will see in Subsection 2.4 this fails horribly. The next best guess would be to define it geometrically by the tangent plane at a suitable non-singular point, but what does that mean for varieties over finite fields? Instead, we use the following definition of the dimension of an ideal I. First, note that if $J \subsetneq I$ and J is a prime ideal, then $V(J)$ must have strictly larger dimension[3] than $V(I)$. Hence, we are led to define

Definition 5 (Krull dimension)**.** *The dimension of $V(I)$, also written as* $\dim(I)$*, is the maximal length of a chain of prime ideals*

$$P_0 \subsetneq P_1 \subsetneq \cdots \subsetneq P_{\dim V(I)} \subsetneq \mathbb{C}[\bar{x}]/I \tag{7}$$

The simplest example is affine subspace $\mathbb{C}^r = V(\langle x_{r+1}, \ldots, x_n\rangle) \subset \mathbb{C}^n$. The quotient $\mathbb{C}[x_1, \ldots, x_n]/I \simeq \mathbb{C}[x_1, \ldots, x_r]$ has the following maximal chain of prime ideals

$$\{0\} \subsetneq \langle x_1\rangle \subsetneq \langle x_1, x_2\rangle \subsetneq \cdots \subsetneq \langle x_1, \ldots, x_r\rangle \subsetneq \mathbb{C}[\bar{x}]/I, \tag{8}$$

in accordance with the expected dimension of $\mathbb{C}^r$.

1.4 Gröbner Bases

How can we actually compute the dimension? Actually, a number of algorithms have been proposed in the literature, for example see [2, 3]. One common fact about all of them is that they rely on Gröbner bases. Instead of computing the dimension in general, I will only consider the question whether the dimension is zero, which is technically a bit easier but uses the same ingredients as the general case.

[3]If we would not have required J to be prime then we would still have $V(J) \supsetneq V(I)$, but this might just be because $V(J)$ contains another irreducible component in addition to $V(I)$.

Theorem 3 (Theorem 6.54 of [2]). *For an ideal $I \subset \mathbb{C}[\bar{x}]$, the following are are equivalent:*

1. $\dim(I) = 0$.
2. $\mathbb{C}[\bar{x}]/I$ *is a finite-dimensional* $\mathbb{C}$*-vector space.*
3. *For every variable x_i there is some non-zero univariate polynomial*

$$b_i(x_i) = x_i^{m_i} + a_{m_i-1}x^{m_i-1} + \cdots + a_0 \ \in I \cap \mathbb{C}[x_i]. \tag{9}$$

Proof. $2 \Rightarrow 3$: The residue classes $\{x_i^k \bmod I \mid k \in \mathbb{Z}_>\}$ must be linearly dependent because they live in a finite-dimensional vector space, so some linear combination is zero modulo I.

$3 \Rightarrow 2$: If we write vdim for the vector space dimension, then

$$\text{vdim}\Big(\mathbb{C}[\bar{x}]/I\Big) \leq \text{vdim}\Big(\mathbb{C}[\bar{x}]/\langle x_i^{m_1}, \ldots, x_n^{m_n}\rangle\Big) = \prod_i m_i \tag{10}$$

$(1 \Rightarrow 3) \Leftrightarrow (\bar{3} \Rightarrow \bar{1})$: If no polynomial of, say, x_1 were in I then there would be a length-1 chain of prime ideals

$$\{0\} \subsetneq \langle x_1\rangle \subsetneq \mathbb{C}[\bar{x}]/I \tag{11}$$

$3 \Rightarrow 1$: Note that the ideal $\langle b_i(x_i)\rangle \subset \mathbb{C}[x_i]$ has dimension 0. Therefore

$$I \supset \langle b_1, \ldots, b_n\rangle \quad \Rightarrow \quad \dim(I) \leq \dim\big(\langle b_1, \ldots, b_n\rangle\big) = 0. \tag{12}$$

□

Using the notation of the theorem, if $\dim(I) = 0$ then there exists a finite number of monomials of the form

$$x_1^{k_1} \cdots x_n^{k_n}, \qquad 0 \leq k_1 < m_1, \ldots, 0 \leq k_n < m_n \tag{13}$$

forming a basis for the residue classes $\mathbb{C}[\bar{x}]/I$. Which exponents we actually use for our residue classes is, in part, up to you. In order to systematically pick a $\mathbb{C}$-basis for the residue classes, one needs to choose an ordering amongs the monomials, and then eliminate all "large" monomials in favor of "smaller" monomials. This is axiomatized as

Definition 6 (Monomial order). *A monomial order is a total order (that is, antisymmetric, transitive, and any two monomials can be compared) on the monomials of a multivariate polynomial ring such that*

1. *the order respects multiplication by monomials:* $a < b \Rightarrow ac < bc$.
2. *any (non-empty) set of monomials has a minimal element.*

Two notable examples are

- Lexicographic order ("lex"): First compare the exponent of x_1; If equal compare the exponent of x_2; If equal compare the exponent of x_3; ...
- Graded reverse lexicographic order ("grevlex"): First compare the total degree (sum of exponents); If equal compare the degree of x_n and reverse the result (that is, $3 > 2 \Rightarrow x_n^3 < x_1 x_n^2$ etc.); If equal compare the degree of x_{n-1} and reverse; ...

Example 1. *Consider the ideal $I = \langle x^2 + xy, x - y^2 \rangle \subset \mathbb{C}[x, y]$. Show that the ideal can also be written as $I = \langle x - y^2, y^4 + y^3 \rangle$. Conclude that you can reduce any polynomial in $\mathbb{C}[x, y]/I$ to a linear combination of the 4 monomials 1, y, y^2, y^3.*

In the above example, presenting the ideal as $I = \langle x - y^2, y^4 + y^3 \rangle$ is an example of a Gröbner basis for the lexicographic monomial order. This means that the leading terms (the highest-degree monomial), written $\mathrm{LT}(p)$, for all polynomials $p \in I$ are generated by the leading terms in the chosen generators of the ideal,

$$\mathrm{LT}(x - y^2) = x, \quad \mathrm{LT}(y^4 + y^3) = y^4. \tag{14}$$

Once we have a Gröbner basis for I, its easy to find a basis of monomials for $\mathbb{C}[\bar{x}]/I$, you only have to look at the leading terms of the generators of I!

Definition 7 (Gröbner basis)**.** *A Gröbner basis (for given monomial order) of an ideal $I \subset \mathbb{C}[\bar{x}]$ is a choice of generators $I = \langle p_1, \ldots, p_k \rangle$ such that*

$$\mathrm{LT}(I) \overset{def}{=} \Big\{ \mathrm{LT}(p) \;\Big|\; p \in I \Big\} = \big\langle \mathrm{LT}(p_1), \ldots, \mathrm{LT}(p_k) \big\rangle \tag{15}$$

The ideal in Example 1 was, initially, not given by a lexicographic Gröbner basis. However, the missing generator with leading term $y^4 \in \mathrm{LT}(I)$ can be generated by systematically subtracting the leading term from two generators. Here, we first subtract off the x^2 term:

$$\big(x^2 + xy\big) - x\big(x - y^2\big) = xy + xy^2 \quad \in I \tag{16}$$

The new generator has a smaller leading term xy, but we can subtract it off again and get

$$\big(xy + xy^2\big) - y\big(x - y^2\big) = xy^2 + y^3 \quad \in I \tag{17}$$

and, finally,

$$\big(xy^2 + y^3\big) - y^2\big(x - y^2\big) = y^3 + y^4 \quad \in I \tag{18}$$

In fact, this method works in general: For any pair of generators, form the so-called S-polynomial that subtracts off the largest leading term. If you get a novel leading term, add it to the generators. Repeat until you have formed all S-polynomials and found no new leading terms. This is known as Buchbergers algorithm and is guaranteed to find a Gröbner basis in a finite number of steps.

Typically, the actual number of steps in Buchbergers algorithm depends very much on the monomial order. A rule of thumb is to use the graded reverse lexicographic order, since it often leads to manageable Gröbner bases. But there are also counterexamples where it produces bases that are exponentially large with the input ideal size, while other monomial orders fare better. Except for trial and error, there is no known way to find the monomial order that leads to the smallest Gröbner basis.

Finally, note that the Gröbner basis does not really depend on the base field of the polynomial ring as long as

- the base field contains all coefficients of the initial generators of the ideal, and

- the base field is either of characteristic 0 or a generic finite field.

Finite fields are slightly tricky, as numerical coefficients can suddenly vanish modulo p. But such coincidences only happen for finitely many primes and not generically.

For applications, it is often important to use this observation and replace a computation over the complex numbers with cyclotomics, rationals, or even finite fields. The latter fields can be represented exactly on a computer, while floating-point numbers cannot. In particular, in forming the S-polynomials it is crucial to know which polynomial coefficients subtract to zero, which is numerically unstable.

2 The Sage Mathematics Software System

2.1 Prologue

If you want to perform non-trivial computations in toric geometry then you invariably end up with the problem that it draws from a wide range of algorithms; you need to solve subproblems dealing with convex geometry, lattices, and Gröbner bases. Because some of these are active areas of research themselves, it is not too surprising that there is no single software written to address all of these with optimal performance. Various toric geometry packages have been written on top of general-purpose systems, for example

- Magma [4] (a.k.a. the "big M" that no physicist has ever heard of) has some support.

- `TorDiv` [5] for Maple.

Other toric geometry packages have been written on top of specialized systems that are very good at things other than toric geometry, for example

- Macaulay2 [6] has a toric variety package.

- Singular [7], too.

- GAP [8] ships with the `toric` [9] package.

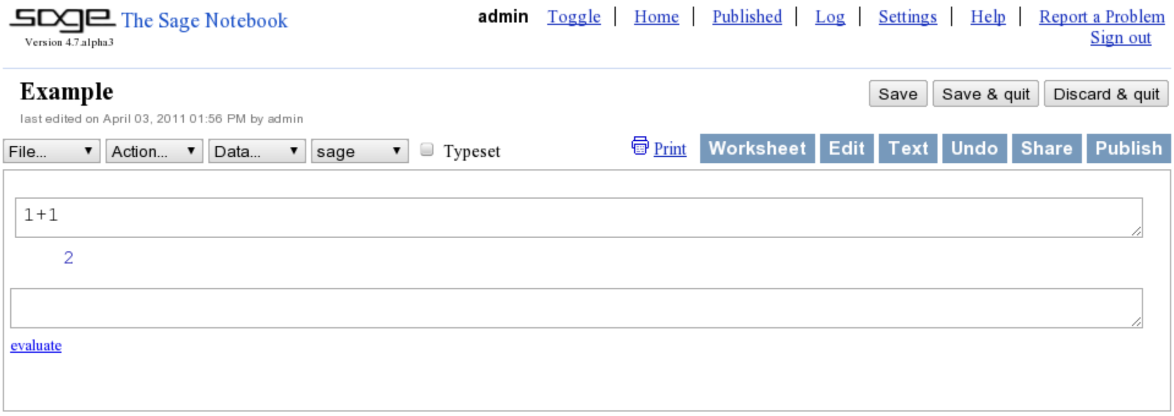


Figure 2: *The browser-based Sage worksheet interface.*

The toric variety package that Andrey Novoseltsev and I wrote differs from all of these in that it does not reinvent the wheel; Instead we reuse a range of open-source libraries that provide fast and well-tested implementations of basic algorithms for dealing with convex geometry, lattices, and Gröbner bases. In fact, this is the philosophy of the Sage [10] mathematics software system, and all necessary libraries were conveniently already packaged in it.

> Sage is free, open-source math software that supports research and teaching in algebra, geometry, number theory, cryptography, numerical computation, and related areas. Both the Sage development model and the technology in Sage itself are distinguished by an extremely strong emphasis on openness, community, cooperation, and collaboration: we are building the car, not reinventing the wheel. The overall goal of Sage is to create a viable, free, open-source alternative to Maple, Mathematica, Magma, and MATLAB.

2.2 The Sage Notebook

The easiest way of interacting with Sage is the notebook interface, which is a HTML / Javascript application that runs in every modern web browser. If you have installed Sage on your computer, you can start the server either via the `sage -notebook` command line option or with the `notebook()` command on the Sage command line. Once you are in the notebook, you can work with the usual question/response paradigm: You type in something to evaluate, press `Shift-Enter` or click on the "evaluate" link, and Sage shows you the result. In the following I will use the command line interface, however, because it is easier to typeset. For example

```
sage: n = 2+3   # creates new variable "n"
sage: n
5
sage: type(n)   # or use "_" to refer to the previous result
<type 'sage.rings.integer.Integer'>
```

The main difference to mathematical software that you may have seen before is that Sage is object-oriented. Roughly, that means that the implementation of algorithms is attached to the data. For you, this means that you usually issue commands in the form `data.command(x,y)` instead of `command(data,x,y)`. These commands that are attached to the data are called *methods* to distinguish them from functions/procedures. For example, to test if an integer is prime you use

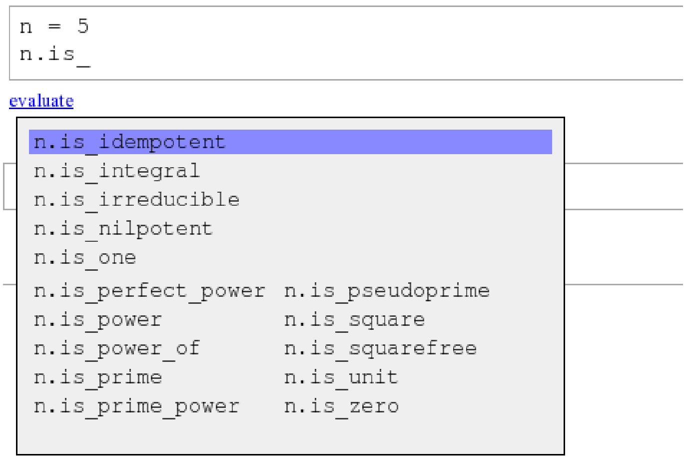


Figure 3: *Example of tab completion.*

```
sage: n.is_prime() # object-oriented
True
```

instead of `is_prime(5)`.[4] One advantage of this approach is that each object, for example the Sage `Integer` with value 5 and assigned to the variable `n`, knows which commands make sense for it. This allows for tab-completion that only returns the methods of the object instead of a giant list of all names. In Figure 3, you can see the tab completion in action.

Once you find method you want to call, you can use the on-line help to learn more about it, including a list of examples showing how it can be used. To access the help, append a question mark to the end of the method name as in "`variable.method?`". Note that tab completion and `?`-help works only for objects that have been assigned to a variable, and not for temporaries. For example, while it is perfectly legal to call `(2+3).is_prime()`, neither tab completion nor on-line help will work on `(2+3)` without assigning it to a variable first.

2.3 Introduction to Sage

Sage itself is mostly written in Python, and it uses Python[5] to interact with the user. Some slight changes are made to the Python syntax to be more suitable for entering maths, for example the caret `^` is parsed as exponentiation instead of bitwise xor and division of integers yields rational numbers instead of C division. See pages. 14, 15 for a quick reference [12] of some elementary Sage commands to get you started.

As an example of how to use Sage, let us revisit the Gröbner basis computation of Subsection 1.4. The first step is to define the polynomial ring and make the generators x, y known to the global namespace,

```
sage: R = PolynomialRing(QQ, 2, 'x, y', order='lex')
sage: R
```

[4]Actually, `is_prime(5)` works as well. It is a function that is added for convenience and calls `5.is_prime()` internally.

[5]Actually, a version of IPython [11].


```
n.is_prime?
```

File: /home/vbraun/Sage/sage/devel/sage/sage/rings/integer.pyx

Type: <type 'builtin_function_or_method'>

Definition: n.is_prime()

Docstring:

Returns `True` if self is prime.

Note

Integer primes are by definition *positive*! This is different than Magma, but the same as in Pari. See also the `is_irreducible()` method.

EXAMPLES:

```
sage: z = 2^31 - 1
sage: z.is_prime()
True
sage: z = 2^31
sage: z.is_prime()
False
sage: z = 7
sage: z.is_prime()
True
sage: z = -7
sage: z.is_prime()
False
sage: z.is_irreducible()
True
```

IMPLEMENTATION: Calls the PARI `isprime` function.

Figure 4: *Example of the on-line help.*

```
Multivariate Polynomial Ring in x, y over Rational Field
sage: R.term_order()
Lexicographic term order
sage: x, y = R.gens()    # or use R.inject_variables()
```

Then we can define the ideal

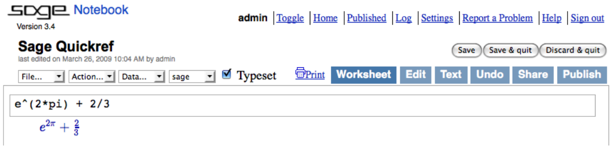
```
sage: I = R.ideal(x^2+x*y, x-y^2)
sage: I.dimension()
0
sage: I.vector_space_dimension()
4
```

and compute the Gröbner basis

```
sage: I.basis_is_groebner()
False
sage: I.groebner_basis()
[x - y^2, y^4 + y^3]
```

The ideal is not prime; Similarly to the decomposition of a prime number into a product of primes, we can decompose the ideal into prime ideals to find the irreducible components

Sage quick reference



Evaluate cell: ⟨shift-enter⟩
Evaluate cell creating new cell: ⟨alt-enter⟩
Split cell: ⟨control-;⟩
Join cells: ⟨control-backspace⟩
Insert math cell: click blue line between cells
Insert text/HTML cell: shift-click blue line between cells
Delete cell: delete content then backspace

Command line

com⟨tab⟩ complete *command*
bar? list command names containing "bar"
command?⟨tab⟩ shows documentation
command??⟨tab⟩ shows source code
`a.`⟨tab⟩ shows methods for object `a` (more: `dir(a)`)
`a._`⟨tab⟩ shows hidden methods for object `a`
`search_doc("`*string or regexp*`")` fulltext search of docs
`search_src("`*string or regexp*`")` search source code
`_` is previous output

Numbers

Integers: $\mathbf{Z}$ = `ZZ` e.g. `-2 -1 0 1 10^100`
Rationals: $\mathbf{Q}$ = `QQ` e.g. `1/2 1/1000 314/100 -2/1`
Reals: $\mathbf{R} \approx$ `RR` e.g. `.5 0.001 3.14 1.23e10000`
Complex: $\mathbf{C} \approx$ `CC` e.g. `CC(1,1) CC(2.5,-3)`
Double precision: `RDF` and `CDF` e.g. `CDF(2.1,3)`
Mod n: $\mathbf{Z}/n\mathbf{Z}$ = `Zmod` e.g. `Mod(2,3) Zmod(3)(2)`
Finite fields: $\mathbf{F}_q$ = `GF` e.g. `GF(3)(2) GF(9,"a").0`
Polynomials: $R[x,y]$ e.g. `S.<x,y>=QQ[] x+2*y^3`
Series: $R[[t]]$ e.g. `S.<t>=QQ[[]] 1/2+2*t+O(t^2)`
p-adic numbers: $\mathbf{Z}_p \approx$`Zp`, $\mathbf{Q}_p \approx$`Qp` e.g. `2+3*5+O(5^2)`
Algebraic closure: $\overline{\mathbf{Q}}$ = `QQbar` e.g. `QQbar(2^(1/5))`
Interval arithmetic: `RIF` e.g. `sage: RIF((1,1.00001))`
Number field: `R.<x>=QQ[];K.<a>=NumberField(x^3+x+1)`

Arithmetic

ab = `a*b` $\frac{a}{b}$ = `a/b` a^b = `a^b` $\sqrt{x}$ = `sqrt(x)`
$\sqrt[n]{x}$ = `x^(1/n)` $|x|$ = `abs(x)` $\log_b(x)$ = `log(x,b)`
Sums: $\sum_{i=k}^{n} f(i)$ = `sum(f(i) for i in (k..n))`
Products: $\prod_{i=k}^{n} f(i)$ = `prod(f(i) for i in (k..n))`

Constants and functions

Constants: π = `pi` e = `e` i = `i` ∞ = `oo`
ϕ = `golden_ratio` γ = `euler_gamma`
Approximate: `pi.n(digits=18)` = 3.14159265358979324
Functions: `sin cos tan sec csc cot sinh cosh tanh sech csch coth log ln exp` ...
Python function: `def f(x): return x^2`

Interactive functions

Put `@interact` before function (vars determine controls)

```
@interact
def f(n=[0..4], s=(1..5), c=Color("red")):
  var("x");show(plot(sin(n+x^s),-pi,pi,color=c))
```

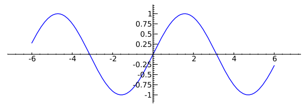
Symbolic expressions

Define new symbolic variables: `var("t u v y z")`
Symbolic function: e.g. $f(x) = x^2$ `f(x)=x^2`
Relations: `f==g f<=g f>=g f<g f>g`
Solve $f = g$: `solve(f(x)==g(x), x)`
`solve([f(x,y)==0, g(x,y)==0], x,y)`
`factor(...) expand(...) (...).simplify_...`
`find_root(f(x), a, b)` find $x \in [a,b]$ s.t. $f(x) \approx 0$

Calculus

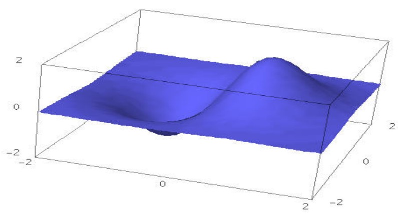
$\lim_{x \to a} f(x)$ = `limit(f(x), x=a)`
$\frac{d}{dx}(f(x))$ = `diff(f(x),x)`
$\frac{\partial}{\partial x}(f(x,y))$ = `diff(f(x,y),x)`
`diff` = `differentiate` = `derivative`
$\int f(x)dx$ = `integral(f(x),x)`
$\int_a^b f(x)dx$ = `integral(f(x),x,a,b)`
$\int_a^b f(x)dx \approx$ `numerical_integral(f(x),a,b)`
Taylor polynomial, deg n about a: `taylor(f(x),x,`a`,`n`)`

2D graphics



`line([(`x_1`,`y_1`),...,(`x_n`,`y_n`)],`*options*`)`
`polygon([(`x_1`,`y_1`),...,(`x_n`,`y_n`)],`*options*`)`
`circle((`x`,`y`),`r`,`*options*`)`
`text("txt",(`x`,`y`),`*options*`)`
options as in `plot.options`, e.g. `thickness=`*pixel*,
`rgbcolor=(`r`,`g`,`b`)`, `hue=`h where $0 \le r,b,g,h \le 1$
`show(`*graphic*, *options*`)`
use `figsize=[w,h]` to adjust size
use `aspect_ratio=`*number* to adjust aspect ratio
`plot(f(`x`),(`x`,`$x_{\min}$`,`$x_{\max}$`),`*options*`)`
`parametric_plot((f(`t`),g(`t`)),(`t`,`$t_{\min}$`,`$t_{\max}$`),`*options*`)`
`polar_plot(f(`t`),(`t`,`$t_{\min}$`,`$t_{\max}$`),`*options*`)`
combine: `circle((1,1),1)+line([(0,0),(2,2)])`
`animate(`*list of graphics, options*`).show(delay=20)`

3D graphics

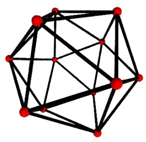


`line3d([(`x_1`,`y_1`,`z_1`),...,(`x_n`,`y_n`,`z_n`)],`*options*`)`
`sphere((`x`,`y`,`z`),`r`,`*options*`)`
`text3d("txt", (`x`,`y`,`z`), `*options*`)`
`tetrahedron((`x`,`y`,`z`),`*size*`,`*options*`)`
`cube((`x`,`y`,`z`),`*size*`,`*options*`)`
`octahedron((`x`,`y`,`z`),`*size*`,`*options*`)`
`dodecahedron((`x`,`y`,`z`),`*size*`,`*options*`)`
`icosahedron((`x`,`y`,`z`),`*size*`,`*options*`)`
`plot3d(f(`x`,`y`),(`x`,`x_b`,`x_e`), (`y`,`y_b`,`y_e`),`*options*`)`
`parametric_plot3d((f,g,h),(`t`,`t_b`,`t_e`),`*options*`)`
`parametric_plot3d((f(`u,v`),g(`u,v`),h(`u,v`)),`
`(`u`,`u_b`,`u_e`),(`v`,`v_b`,`v_e`),`*options*`)`
options: `aspect_ratio=`$[1,1,1]$, `color="red"`
`opacity=0.5, figsize=6, viewer="tachyon"`

Discrete math

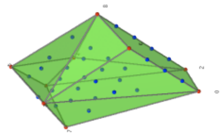
$\lfloor x \rfloor$ = `floor(x)` $\lceil x \rceil$ = `ceil(x)`
Remainder of n divided by k = `n%k` $k|n$ iff `n%k==0`
$n!$ = `factorial(n)` $\binom{x}{m}$ = `binomial(x,m)`
$\phi(n)$ = `euler_phi(n)`
Strings: e.g. `s = "Hello" = "He"+'llo'`
`s[0]="H"  s[-1]="o"  s[1:3]="el"  s[3:]="lo"`
Lists: e.g. `[1,"Hello",x] = []+[1,"Hello"]+[x]`
Tuples: e.g. `(1,"Hello",x)` (immutable)
Sets: e.g. $\{1,2,1,a\}$ = `Set([1,2,1,"a"])` ($= \{1,2,a\}$)
List comprehension ≈ set builder notation, e.g.
$\{f(x) : x \in X, x > 0\}$ = `Set([f(x) for x in X if x>0])`

Graph theory



Graph: `G = Graph({0:[1,2,3], 2:[4]})`
Directed Graph: `DiGraph(`*dictionary*`)`
Graph families: `graphs.`⟨tab⟩
Invariants: `G.chromatic_polynomial()`, `G.is_planar()`
Paths: `G.shortest_path()`
Visualize: `G.plot()`, `G.plot3d()`
Automorphisms: `G.automorphism_group()`,
`G1.is_isomorphic(G2)`, `G1.is_subgraph(G2)`

Combinatorics



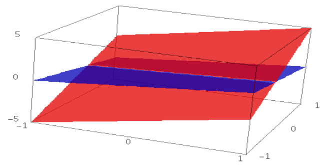
Integer sequences: `sloane_find(`*list*`)`, `sloane.`⟨tab⟩
Partitions: `P=Partitions(`*n*`)` `P.count()`
Combinations: `C=Combinations(`*list*`)` `C.list()`
Cartesian product: `CartesianProduct(P,C)`
Tableau: `Tableau([[1,2,3],[4,5]])`
Words: `W=Words("abc"); W("aabca")`
Posets: `Poset([[1,2],[4],[3],[4],[]])`
Root systems: `RootSystem(["A",3])`
Crystals: `CrystalOfTableaux(["A",3], shape=[3,2])`
Lattice Polytopes: `A=random_matrix(ZZ,3,6,x=7)`
`L=LatticePolytope(A)` `L.npoints()` `L.plot3d()`

Matrix algebra

$\binom{1}{2}$ = `vector([1,2])`
$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ = `matrix(QQ,[[1,2],[3,4]], sparse=False)`
$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$ = `matrix(QQ,2,3,[1,2,3, 4,5,6])`
$\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix}$ = `det(matrix(QQ,[[1,2],[3,4]]))`
Av = `A*v` A^{-1} = `A^-1` A^t = `A.transpose()`
Solve $Ax = v$: `A\v` or `A.solve_right(v)`
Solve $xA = v$: `A.solve_left(v)`
Reduced row echelon form: `A.echelon_form()`
Rank and nullity: `A.rank()` `A.nullity()`
Hessenberg form: `A.hessenberg_form()`
Characteristic polynomial: `A.charpoly()`
Eigenvalues: `A.eigenvalues()`
Eigenvectors: `A.eigenvectors_right()` (also left)
Gram-Schmidt: `A.gram_schmidt()`
Visualize: `A.plot()`

LLL reduction: `matrix(ZZ,...).LLL()`
Hermite form: `matrix(ZZ,...).hermite_form()`

Linear algebra



Vector space K^n = `K^n` e.g. `QQ^3` `RR^2` `CC^4`
Subspace: `span(vectors,` *field* `)`
E.g., `span([[1,2,3], [2,3,5]], QQ)`
Kernel: `A.right_kernel()` (also left)
Sum and intersection: `V + W` and `V.intersection(W)`
Basis: `V.basis()`
Basis matrix: `V.basis_matrix()`
Restrict matrix to subspace: `A.restrict(V)`
Vector in terms of basis: `V.coordinates(`*vector*`)`

Numerical mathematics

Packages: `import numpy, scipy, cvxopt`
Minimization: `var("x y z")`
  `minimize(x^2+x*y^3+(1-z)^2-1, [1,1,1])`

Number theory

Primes: `prime_range(n,m)`, `is_prime`, `next_prime`
Factor: `factor(n)`, `qsieve(n)`, `ecm.factor(n)`
Kronecker symbol: $\left(\frac{a}{b}\right)$ = `kronecker_symbol(`a,b`)`
Continued fractions: `continued_fraction(x)`
Bernoulli numbers: `bernoulli(n)`, `bernoulli_mod_p(p)`
Elliptic curves: `EllipticCurve(`$[a_1,a_2,a_3,a_4,a_6]$`)`
Dirichlet characters: `DirichletGroup(`*N*`)`
Modular forms: `ModularForms(`*level*, *weight*`)`
Modular symbols: `ModularSymbols(`*level*, *weight*, *sign*`)`
Brandt modules: `BrandtModule(`*level*, *weight*`)`
Modular abelian varieties: `J0(`*N*`)`, `J1(`*N*`)`

Group theory

`G = PermutationGroup([[(1,2,3),(4,5)],[(3,4)]])`
`SymmetricGroup(`*n*`)`, `AlternatingGroup(`*n*`)`
Abelian groups: `AbelianGroup([3,15])`
Matrix groups: `GL, SL, Sp, SU, GU, SO, GO`
Functions: `G.sylow_subgroup(p)`, `G.character_table()`,
`G.normal_subgroups()`, `G.cayley_graph()`

Noncommutative rings

Quaternions: `Q.<i,j,k> = QuaternionAlgebra(a,b)`
Free algebra: `R.<a,b,c> = FreeAlgebra(QQ, 3)`

Python modules

`import` *module_name*
module_name`.`⟨tab⟩ and `help(module_name)`

Profiling and debugging

`time` *command*: show timing information
`timeit("command")`: accurately time command
`t = cputime(); cputime(t)`: elapsed CPU time
`t = walltime(); walltime(t)`: elapsed wall time
`%pdb`: turn on interactive debugger (command line only)
`%prun command`: profile command (command line only)

```
sage: I.primary_decomposition()
[Ideal (y^3, x - y^2) of Multivariate Polynomial Ring in x, y over
    Rational Field, Ideal (y + 1, x - 1) of Multivariate
   Polynomial Ring in x, y over Rational Field]
sage: I.radical().primary_decomposition()
[Ideal (y, x) of Multivariate Polynomial Ring in x, y over
   Rational Field, Ideal (y + 1, x - 1) of Multivariate Polynomial
    Ring in x, y over Rational Field]
```

So, thought of as a 0-dimensional variety, we obtain $V(I) = \{(0,0),\ (1,-1)\}$.

2.4 Algebraic Geometry in Sage

The ideal-theoretic commands explored in the previous subsection form the basis for geometric computations. But you don't necessarily have to do the translation into geometry by hand; Sage also has object representing algebraic varieties directly. As an example, consider the

Example 2 (twisted cubic)**.** *The twisted cubic is the subvariety of $\mathbb{P}^3$ with homogeneous variables z_0, z_1, z_2, z_3 cut out by the 2×2 minors of the matrix $\left(\begin{smallmatrix} z_0 & z_1 & z_2 \\ z_1 & z_2 & z_3 \end{smallmatrix}\right)$.*

Since there are three columns, the twisted cubic is defined by three homogeneous polynomial equations. None of the three equations is implied by the other two. Nevertheless, their common solution set is a 1-dimensional variety.

```
sage: P3.<z0,z1,z2,z3> = ProjectiveSpace(3, QQ)
sage: minors = matrix([[z0, z1, z2], [z1, z2, z3]]).minors(2)
sage: twisted_cubic = P3.subscheme(minors)
sage: twisted_cubic
Closed subscheme of Projective Space of dimension 3 over Rational
   Field defined by:
  -z1^2 + z0*z2,
  -z1*z2 + z0*z3,
  -z2^2 + z1*z3
sage: twisted_cubic.dimension()
1
```

Let us consider the affine patch where $z_0 = 1$. In this patch, the affine algebraic variety is

```
sage: C = twisted_cubic.affine_patch(0)
sage: C
Closed subscheme of Affine Space of dimension 3 over Rational
   Field defined by:
  -x0^2 + x1,
  -x0*x1 + x2,
  -x1^2 + x0*x2
```

In the affine patch, the curve C is actually the intersection of two hypersurfaces:

```
sage: A3 = C.ambient_space()
sage: A3.inject_variables()
None
sage: H1 = A3.subscheme(x0*x1-x2)
sage: H2 = A3.subscheme(x0^2-x1)
sage: C == H1.intersection(H2)
```

```
True
```

You can use tab-completion to explore further geometric properties.

Exercise 4. *Using Sage, construct the x-axis and the yz-plane as affine algebraic varieties in $\mathbb{C}^3$. Take their union. What is its dimension? Decompose it into irreducible components.*

2.5 The Python Language

There are lots of good resources for learning Python. For example, the official Python tutorial [13]. I'll just cover a few key concepts to get you started.

2.5.1 List Comprehensions

A list is the basic Python container. It can contain anything:

```
sage: [ 1, PolynomialRing(QQ,2,'x,y'), 'a␣string' ]
[1, Multivariate Polynomial Ring in x, y over Rational Field, 'a
   string']
sage: primes_first_20 = primes_first_n(20)
sage: primes_first_20
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59,
   61, 67, 71]
```

List comprehensions make new lists from old. For example, the squares of the entries

```
sage: [ i^2 for i in primes_first_20 ]
[4, 9, 25, 49, 121, 169, 289, 361, 529, 841, 961, 1369, 1681,
   1849, 2209, 2809, 3481, 3721, 4489, 5041]
```

or pick every second entry

```
sage: [ p for i,p in enumerate(primes_first_20) if i%2 == 0 ]
[2, 5, 11, 17, 23, 31, 41, 47, 59, 67]
sage: primes_first_20[0::2]   # can also be done with the slicing
   operator
[2, 5, 11, 17, 23, 31, 41, 47, 59, 67]
```

2.5.2 Control Flow

For loops can only run over a elements of a list or more general iterators. But there is no special syntax to loop over a range of integers, you have to use the `range()` function to generate lists of subsequent integers to loop over:

```
sage: log = []
...   for i in range(0,5):
...       temporary_variable = 1
...       log.append('i␣=␣'+str(i))
...   log.append('End␣of␣loop')
sage: log
['i = 0', 'i = 1', 'i = 2', 'i = 3', 'i = 4', 'End of loop']
```

Note also that blocks like the loop body are marked by indentation, which is a Python specialty and generally not found in other languages. Therefore, Python code must be consistently indented and the interpreter will catch some indentation mistakes for you.

2.5.3 Functions

Defining functions is rather straightforward. Note that the function body must again be indented to make clear where it starts and where it ends:

```
sage: def hello_world():
...       return 'Hello,␣World!'
```

Once we have declared the function, we can call it as usual

```
sage: hello_world()
Hello, World!
```

2.5.4 Classes

Finally, here is a quick overview over how to declare new classes. I'm assuming here that you have seen another object-oriented language before. Otherwise you can skip over it, we will not need it in the following.

A class must always inherit from somewhere, possibly from the Python base `object`. Here is a class with a single method and a data member:

```
sage: class foo(object):
...       def bar(self):
...           return 'method_bar()␣of␣class␣foo.'
...       y = 1    # data member
```

Note again that the indentation is crucial! Instantiation is then rather straight-forward:

```
sage: x = foo()
sage: x
<__main__.foo object at 0x5543090>
```

Finally, we can call the method and access the data member:

```
sage: x.bar()
method bar() of class foo.
sage: x.y
1
```

2.6 Cython and Scientific Computation

Python is an interpreted language, so you can write Python code and immeadiately execute it without a compile/link/execute cycle. Also, you can inspect objects and call methods directly from the command line. The disadvantage is speed: An integer, say, is not represented by a machine integer but wrapped in a `PyObject` C struct.

To sum two integers, say, Python first has to figure that the `PyObjects` represent integers, then add the integers, then allocate memory for the resuling new `PyObject`, store the integer, and update its internal reference counting.

The most basic way of speeding up evaluation of symbolic expressions is to create and store the intermediate expression trees. Then, during the evaluation, the interpreter doesn't have to wrap intermediate results over and over into objects of the interpreter. For example, this done in Sage with `fast_callable()` and in Mathematica with `Compile[]`. But that will only help with evaluating some kinds of expressions, and not improve arbitrary code. Moveover, the result is still a far cry from the performance you would get from straight C/C++ code.

The reason why C/C++ code is so much faster is that the compiler and optimizer can apply vast knowledge about the CPU architecture because they directly control the resulting machine code. For example, often-used variables can be stored in registers directly in the CPU instead of the main memory. And accessing the main memory takes ≥ 100 clock cycles on modern architectures. Hence, ideally, all speed-critical code should be passed through an optimizing compiler. But writing C/C++ code is difficult and making it pass data to/from the Python interpreter is even worse. The solution to this dilemma is Cython, which essentially transforms Python code into C/C++. It makes use of a few Python language enhancements to, for example, specify the type of variables. Cython also automatically generates the necessary code to pass variables from Python to C/C++ and back. By compiling the C/C++ source into a shared library and dynamically loading it into the current Python session, you can use the result as if it were a Python function. For example, take this sample procedure that adds the integers from 0 to 99:

```
sage: def python_sum_0_99():
...       s = 0
...       for i in range(0,100):
...           s += i
...       return s
sage: python_sum_0_99()
4950
```

The analogous Cython version is almost identical[6]

```
%cython
def cython_sum_0_99():
    cdef int i, s
    s = 0
    for i in range(0,100):
        s += i
    return s
```

Cython translates this into the following C code

```
int __pyx_v_i;
int __pyx_v_s;
int __pyx_t_1;
__pyx_v_s = 0;
for (__pyx_t_1 = 0; __pyx_t_1 < 100; __pyx_t_1+=1) {
```

[6]Note that the `%cython` magic only works in the Sage worksheet and not on the command line. There, you have to use the `cython()` function.

```
    __pyx_v_i = __pyx_t_1;
    __pyx_v_s = (__pyx_v_s + __pyx_v_i);
  }
```

together with C comments to help relate the generated source back to the Cython code. In the interest of breverety I removed these comments in the above code snippet. As advertised, the inner loop now uses plain machine integers and no Python objects any more. We can get accurate timing with the `timeit()` function:

```
sage: timeit('python_sum_0_99()')
625 loops, best of 3: 206 µs per loop
sage: timeit('cython_sum_0_99()')
625 loops, best of 3: 176 ns per loop
```

So the Cython version runs about 1000 times faster!

Exercise 5. *Remove the `cdef int i,s` line from the Cython version. Does it still work? Look at how the loop body is now implemented in the generated C source code.*

3 Affine Toric Varieties

3.1 Schemes

Given the ideal $I = \langle x^2 \rangle \subset \mathbb{C}[x]$, the associated variety is $V(I) = \{0\}$. The variety is the same as the variety of the radical $\sqrt{I} = I(V(I)) = \langle x \rangle$. It is a pity that we are loosing information when going to the variety, it would be much nicer if we could associate to I the point 0 "with multiplicity two". An *affine scheme* is precisely that, a generalization of an affine algebraic variety that keeps track of the "multiplicities". One might be tempted to define a scheme directly as being equivalent to the ideal I. But that is not quite satisfactory. For example, the ideals $\langle x \rangle$ and $\langle y \rangle \subset \mathbb{C}[x, y]$ are definitely different ideals. Yet we would like to treat them as isomorphic schemes, since they are both a flat hyperplane $\mathbb{C} \subset \mathbb{C}^2$ with unit multiplicity. Instead, we define a scheme as the geometric space defined via its functions.

First, consider an ordinary variety $X = V(I)$ associated to an ideal $I \subset \mathbb{C}[\bar{x}]$. What are the functions on it? Any polynomial in $\mathbb{C}[\bar{x}]$ defines a function on X. But any two polynomials whose difference is in $I(X)$ yield the same function, since the polynomials in $I(X)$ vanish on X. Therefore, the ring of polynomial functions on X is the quotient ring $\mathbb{C}[\bar{x}]/I(X)$. The elements are equivalence classes of polynomials modulo the equivalence relation

$$p \sim q \quad \Leftrightarrow \quad p - q \in I(X). \tag{19}$$

We can now generalize this definition to any ideal $I \in \mathbb{C}[\bar{x}]$, not necessarily of the form $I(X)$ for some variety X. The ideal defines a quotient ring $C[\bar{x}]/I$, which we can think of as the ring of functions defining the scheme. We write

Definition 8 (Affine scheme)**.** *Given a ring R (for example, $R = \mathbb{C}[\bar{x}]/I$), we denote the corresponding affine scheme by* $\mathrm{Spec}(R)$*, the spectrum of the ring R. Two schemes are isomorphic if their defining rings are.*

Exercise 6. *Describe the scheme* $\operatorname{Spec}\big(\langle x^2y - xy,\ xy^2 - y^2\rangle\big)$*. Hint: Compare the irreducible components of the variety with the primary decomposition of the defining ideal.*

No, really. What is a scheme?

I have not formally defined what an affine scheme is as we do not need the machinery in general. Really, you should think of the scheme as being the geometric object defined by the ring of functions $\mathbb{C}[\bar{x}]/I$. All geometric properties, like the dimension, are defined in terms of algebraic properties of the ideal I. But inquiring minds want to know more. So let me give you some of the salient points. For more details that you will care to know, see [14].

In fact, defining a scheme is very much analogous to how you would define a smooth manifold. There are essentially three sucessive layers of structure. First of all, a manifold is a set of points. The second layer is the topology, that is, you have to define which subsets of points are called "open". Together, this defines a topological space. To furthermore define a smooth manifold X, you need to pick a subset $\mathbb{C}^\infty(X) \subset C^0(X)$ of "smooth" functions amongs the continuous functions. Usually, you do this by choosing smooth transition functions. At each level, you can make extra choices. There are usually multiple topologies on a given set of points, and multiple smooth structures on a give toplogical space.

Schemes, by comparison, also start at the level of sets. For a given ring R, the affine scheme $\operatorname{Spec}(R)$ is the set of prime ideals in R. On top of that, there is the Zariski topology. So the scheme $\operatorname{Spec}(R)$ is a topological space, but that is not all. Finally, a scheme knows about its ring of functions. As with smooth manifolds, there are usually multiple choices at each level. All three together, the set of prime ideals with a topology and a choice of its ring of functions, constitute a scheme.

Example 3. *Consider the (non-reduced) scheme* $\operatorname{Spec}\big(\mathbb{C}[x]/\langle x^2\rangle\big)$*. The only*[7] *prime ideal in* $\mathbb{C}[x]/\langle x^2\rangle$ *is* $\langle x\rangle$*. So as a set, the scheme consists of a single point. The topology on a single point is uniquely defined. So far, everything is the same as the reduced scheme* $\operatorname{Spec}(\mathbb{C}[x]/\langle x\rangle)$*, which also contains a single single point, namely the prime ideal* $\langle 0\rangle$*. The difference between these two schemes is the ring of functions.*

3.2 Cones and Lattices

Here is a generalization of a polynomial ring that we can use to define new schemes. Lets start with the polynomial ring $\mathbb{C}[\bar{x}] = \mathbb{C}[x_0, \dots, x_d]$ in d variables. We can think of it as formal $\mathbb{C}$-linear combinations of monomials; Addition is the formal addition and multiplication is defined by distributivity and the semigroup law

$$\left(x_1^{m_1} x_2^{m_2} \cdots x_d^{m_d}\right) \cdot \left(x_1^{n_1} x_2^{n_2} \cdots x_d^{n_d}\right) = x_1^{m_1+n_1} x_2^{m_2+n_2} \cdots x_d^{m_d+n_d} \tag{20}$$

on monomials. The (Abelian) semigroup of monomials is just $(\mathbb{Z}_\geq)^d$, the semigroup of d-tuples of nonnegative integers with componentwise addition.

[7] In particular, the zero ideal is not prime.

We can define interesting new rings by exchanging the semigroup $(\mathbb{Z}_{\geq})^d$ for a different semigroup. The easiest generalization is to go from the integral points in the positive d-orthant to the integral points in any cone. For simplicity, we require the cone to be rational polyhedral:

Definition 9 (rational polyhedral cone)**.** *A rational polyhedral cone σ is a subset*

$$\sigma = \operatorname{span}_{\mathbb{Q}_{\geq}} \big\{ r_1, \dots, r_n \big\} \subset \mathbb{Q}^d. \tag{21}$$

We usually scale each generating ray to be integral, $r_i \in \mathbb{Z}^d$.

Example 4. *The origin $\{0\}$ as well as the whole space $\mathbb{Q}^d$ are rational polyhedral cones.*

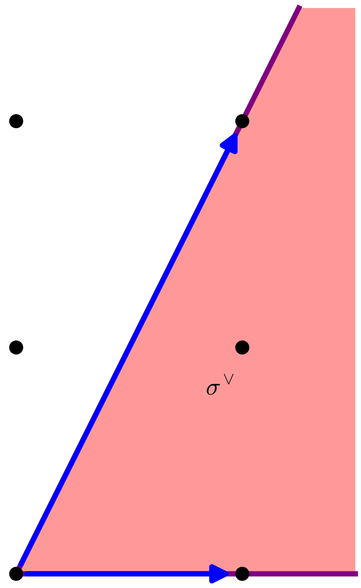


Figure 5: *A 2-d cone.*

Note that there are two ways to describe cones, using rays or using inequalities. The cone in Figure 5 is

$$\sigma^\vee = \operatorname{span}_{\mathbb{Q}_{\geq}} \big\{ \left(\begin{smallmatrix}1\\0\end{smallmatrix}\right), \left(\begin{smallmatrix}1\\2\end{smallmatrix}\right) \big\} = \Big\{ \left(\begin{smallmatrix}x\\y\end{smallmatrix}\right) \;\Big|\; 0 \cdot x + 1 \cdot y \geq 0,\; 2 \cdot x - 1 \cdot y \geq 0 \Big\} \tag{22}$$

In fact, these are dual descriptions: the coefficients of the inequalities are rays of the *dual* cone. The dual cone can again be described by inequalities, whose coefficients are the original rays. So dualizing a cone twice reproduces the cone one started with. For example, the dual cone to the cone in Figure 5 is

$$\sigma^{\vee\vee} = \sigma = \operatorname{span}_{\mathbb{Q}_{\geq}} \big\{ \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right), \left(\begin{smallmatrix}2\\-1\end{smallmatrix}\right) \big\} = \Big\{ \left(\begin{smallmatrix}x\\y\end{smallmatrix}\right) \;\Big|\; 1 \cdot x + 0 \cdot y \geq 0,\; 1 \cdot x + 2 \cdot y \geq 0 \Big\} \tag{23}$$

For reasons that will remain mysterious until we reach Part II, we will always use the dual cone to define toric varieties. Given a cone $\sigma^\vee \subset \mathbb{Q}^d$, its integral points $\sigma^\vee \cap \mathbb{Z}^d$ form a semigroup and allows us to define a ring $\mathbb{C}[\sigma^\vee \cap \mathbb{Z}^d]$ analogously to a polynomial ring.

Definition 10 (Affine toric variety)**.** *Let $M \simeq \mathbb{Z}^d$ be a lattice in $M_{\mathbb{Q}} \simeq \mathbb{Q}^d$. The affine toric variety defined by the dual cone $\sigma^\vee \subset M_{\mathbb{Q}}$ is the affine scheme*

$$\mathbb{P}_\sigma = \operatorname{Spec}\big(\mathbb{C}[\sigma^\vee \cap M]\big). \tag{24}$$

The analogue of the variables generating the polynomial ring are the irreducible lattice points in $\sigma^\vee \cap M$, that is, the non-zero lattice points that cannot be written as a sum of two other non-zero points in $\sigma^\vee \cap M$.

Definition 11 (Hilbert basis)**.** *The set of all non-zero irreducible elements in $\sigma^\vee \cap M$ is called a Hilbert basis.*

A Hilbert basis always exists and forms a minimal generating set for the semigroup $\sigma^\vee \cap M$. It is unique if the cone is strictly convex, that is, it does not contain a straight line through the origin. For the cone in Figure 5, it is

```
sage: sigma_dual = Cone([(0,1),(2,-1)]).dual()
sage: sigma_dual
2-d cone in 2-d lattice M
sage: sigma_dual.rays()
(M(1, 0), M(1, 2))
sage: sigma_dual.Hilbert_basis()
(M(1, 0), M(1, 2), M(1, 1))
```

It is convenient to denote the generators by variables analogous to polynomial rings. We set

$$x \stackrel{\text{def}}{=} \left(\begin{smallmatrix}1\\0\end{smallmatrix}\right)_M, \quad y \stackrel{\text{def}}{=} \left(\begin{smallmatrix}1\\2\end{smallmatrix}\right)_M, \quad z \stackrel{\text{def}}{=} \left(\begin{smallmatrix}1\\1\end{smallmatrix}\right)_M. \tag{25}$$

These satisfy one relation

$$\left(\begin{smallmatrix}1\\0\end{smallmatrix}\right)_M + \left(\begin{smallmatrix}1\\2\end{smallmatrix}\right)_M = 2\left(\begin{smallmatrix}1\\1\end{smallmatrix}\right)_M \qquad \Leftrightarrow \qquad xy = z^2 \tag{26}$$

Therefore, we can identify the semigroup ring of $\sigma^\vee \cap M$ with the quotient ring

$$\mathbb{C}[\sigma^\vee \cap M] = \mathbb{C}[x, y, z]/\langle xy - z^2\rangle. \tag{27}$$

By rewriting the semigroup ring as a quotient of a polynomial ring, we see that the affine toric variety $\mathbb{P}_\sigma$ is also the affine algebraic variety $V(\langle xy - z^2\rangle) \subset \mathbb{C}^3$.

Exercise 7. *Let $\mathbb{Z}_2$ act on $\mathbb{C}^2 = \operatorname{Spec}\mathbb{C}[X, Y]$ by $(X, Y) \mapsto (-X, -Y)$. Show that the ring of $\mathbb{Z}_2$-invariant polynomials in $\mathbb{C}[X, Y]$ is isomorphic to $\mathbb{C}[\sigma^\vee \cap M]$ as in eq.* (27)*. Conclude that $\mathbb{P}_\sigma \simeq \mathbb{C}^2/\mathbb{Z}_2$.*

3.3 Torus Action and Orbifolds

Let $\mathbb{C}^\times = \mathbb{C} - \{0\}$ be the complexification of $U(1)$. We observe that we can let the *algebraic torus* $(\mathbb{C}^\times)^d$ act on the monomial $x^{(m_1,\dots,m_d)}$ associated to the point $(m_1, \dots, m_d) \in \sigma^\vee \cap M$ as the phase

$$(\chi_1, \dots, \chi_m) \cdot x^{(m_1,\dots,m_d)} = \left(\prod_{i=1}^{d} \chi_i^{m_i}\right) x^{(m_1,\dots,m_d)}. \tag{28}$$

This extends to a group action on the semigroup ring $\mathbb{C}[\sigma^\vee \cap M]$ and, hence, to a group action on the affine toric variety $\operatorname{Spec}(\mathbb{C}[\sigma^\vee \cap M])$. There is one maximal-dimensional orbit where all coordinates are non-zero, which we call the maximal torus or big torus. Here, torus always means the algebraic torus $(\mathbb{C}^\times)^d$. We can think of the algebraic torus as the maps $M \to \mathbb{C}^\times$ respecting the group law, that is, as group homomorphisms

$$(\mathbb{C}^\times)^d = \operatorname{Hom}(M, \mathbb{C}^\times) \tag{29}$$

Something special happens if we refine the lattice M, or, equivalently, make the dual lattice $N \stackrel{\text{def}}{=} M^\vee$ more coarse. That is, consider the case where we have a finite-index sublattice $N' \subset N$ with dual lattices $M = N^\vee$ and $M' = (N')^\vee$. Clearly, the semigroup ring

$$\mathbb{C}[\sigma^\vee \cap M] \ \subset \ \mathbb{C}[\sigma^\vee \cap M'] \tag{30}$$

is a subring. How can we characterize it? First, let us express the sublattice relations as short exact sequences

$$\begin{aligned} 0 \longrightarrow N' \longrightarrow N \longrightarrow N/N' \longrightarrow 0 \\ 0 \longrightarrow M \longrightarrow M' \longrightarrow M'/M \longrightarrow 0 \end{aligned} \tag{31}$$

with $G \stackrel{\text{def}}{=} N/N'$ a finite Abelian group. It implies [15] that

$$0 \longrightarrow \underbrace{\mathrm{Hom}(M'/M, \mathbb{C}^\times)}_{\simeq G} \longrightarrow \mathrm{Hom}(M', \mathbb{C}^\times) \longrightarrow \mathrm{Hom}(M, \mathbb{C}^\times) \longrightarrow 0. \tag{32}$$

So the algebraic torus action on $\mathbb{C}[\sigma^\vee \cap M']$ is almost the same as the action on $\mathbb{C}[\sigma^\vee \cap M])$ except that it "spins faster", and there is a finite subgroup G that acts trivially on a monomial $m' \in M'$ if and only if it is actually in the sublattice $M \subset M'$. In other words,

$$\mathbb{C}[\sigma^\vee \cap M] = \mathbb{C}[\sigma^\vee \cap M']^G \tag{33}$$

is the subring of G-invariants. Hence, the inclusion $N' \to N$ induces the quotient

$$\mathrm{Spec}\left(\mathbb{C}[\sigma^\vee \cap M]\right) \simeq \mathrm{Spec}\left(\mathbb{C}[\sigma^\vee \cap M']\right)/G \tag{34}$$

Consider the case where the cone is *simplicial*, that is, rational polyhedral and $\sigma^\vee = \mathrm{span}\{r_1, \dots, r_d\} \subset \mathbb{Q}^d$ is the cone over a $(d-1)$-simplex. We can take the coordinates of the r_i to be integral. The lattice $M = \mathbb{Z}r_1 \oplus \cdots \oplus \mathbb{Z}r_d$ is then a sublattice of the standard lattice $M' = \mathbb{Z}^d$. Then $\mathrm{Spec}(\mathbb{C}[\sigma^\vee \cap M']) = \mathbb{C}^d$ is just the ordinary affine space, and $\mathrm{Spec}(\mathbb{C}[\sigma^\vee \cap M]) = \mathbb{C}^d/G$ is an orbifold.[8]

To summarize,

Proposition 1. *We can distinguish three successively more singular cases:*

- *If the strictly convex cone is smooth, that is,* $\sigma^\vee = \mathrm{span}\{e_1, \dots, e_d\}$ *is* $GL(d,\mathbb{Z})$*-equivalent to the standard d-orthant, then* $\mathrm{Spec}(\mathbb{C}[\sigma^\vee \cap M]) = \mathbb{C}^d$ *is the d-dimensional affine space. This is the only smooth affine toric variety.*
- *If the strictly convex cone* $\sigma^\vee = \mathrm{span}\{r_1, \dots, r_d\}$ *is simplicial, then* $\mathrm{Spec}(\mathbb{C}[\sigma^\vee \cap M]) = \mathbb{C}^d/G$ *is an orbifold.*
- *If the strictly convex cone* $\sigma^\vee = \mathrm{span}\{r_1, \dots, r_k\}$, $k > d$ *is not simplicial then the toric variety has a singularity that is not an orbifold.*

Exercise 8. *What about non-strictly convex cones? Consider the full space* $\sigma^\vee = \mathbb{Q} \subset \mathbb{Q}$*. What is the semigroup ring? Show that*

$$\mathrm{Spec}\left(\mathbb{C}[\sigma^\vee \cap M]\right) = \mathbb{C}^\times. \tag{35}$$

Generalizing 8, if we split a non-strictly convex cone $\sigma^\vee = \rho^\vee \times \mathbb{Q}^k \subset \mathbb{Q}^d$ into a $(d-k)$-dimensional cone times an affine subspace, then

$$\mathrm{Spec}\left(\mathbb{C}[\sigma^\vee \cap M]\right) = \mathrm{Spec}\left(\mathbb{C}[\rho^\vee \cap M]\right) \times \left(\mathbb{C}^\times\right)^k \tag{36}$$

[8] Here and in the following, orbifold will always mean a quotient by a *finite* group.

3.4 The Conifold

A singularity that occurs very often is the conifold, which is the simplest non-quotient singularity. In terms of toric geometry, it is defined by the non-simplicial cone over a minimal lattice square at distance 1:

```
sage: conifold = toric_varieties.Conifold()
sage: conifold.is_smooth()
False
sage: conifold.is_orbifold()
False
sage: square_cone = conifold.fan().generating_cone(0)
sage: square_cone.rays()
(N(0, 0, 1), N(0, 1, 1), N(1, 0, 1), N(1, 1, 1))
sage: patch = conifold.affine_algebraic_patch(square_cone)
sage: patch
Closed subscheme of Affine Space of dimension 4 over Rational
    Field defined by:
  z0*z2 - z1*z3
```

The conifold is not smooth because the hypersurface equation is not transverse at $\bar{z} = (0, 0, 0, 0)$. More precisely, the singularities are the variety of the Jacobian ideal

$$\operatorname{Jac}(f) = \left\langle f,\ \frac{\partial f}{\partial z_0},\ \ldots,\ \frac{\partial f}{\partial z_3} \right\rangle \tag{37}$$

For the conifold, it is

```
sage: Jac = patch.Jacobian()
sage: Jac
Ideal (z0*z2 - z1*z3, z2, -z3, z0, -z1) of Multivariate Polynomial
     Ring in z0, z1, z2, z3 over Rational Field
sage: A4 = patch.ambient_space()
sage: origin = A4.subscheme(A4.gens())  # the origin (0,0,0,0)
sage: A4.subscheme(Jac) == origin
True
```

The most basic invariant of an isolated hypersurface singularity is its *Milnor number*, which is the vector space dimension of $\mathbb{C}[\bar{x}]/\operatorname{Jac}(f(\bar{x}))$. For the conifold, it is one:

```
sage: Jac.vector_space_dimension()
1
```

In fact, the converse is also true: A 3-dimensional isolated singularity of Milnor number one is a conifold. Note, however, that higher Milnor numbers no longer uniquely determine the singularity.

Exercise 9. *Use Sage to compute the Milnor number of the singularity* $\mathbb{C}^2/\mathbb{Z}_n$ *for* $n \in \{2, 3, \ldots, 10\}$.

Since drawing 3-dimensional cones in a recognizable way requires some graphical designer skills, we use the following 2-dimensional notation.

Definition 12 (Toric diagram)**.** *A toric diagram is a 2-dimensional lattice polytope. It determines a 3-dimensional cone by embedding it at distance 1 and taking the cone over it. That is, by assigning the point at* $(x, y) \mapsto$ *ray* $(1, x, y)$.

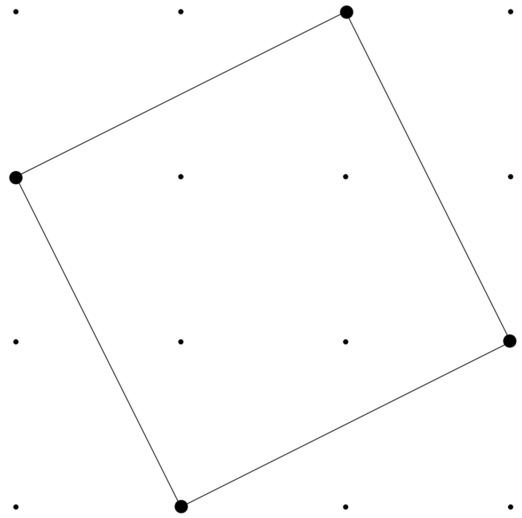


Figure 6: *The toric diagram for the $\mathbb{Z}_5$ hyperconifold.*

Exercise 10. *In Figure 6 is the toric diagram for the conifold as well as the so-called $\mathbb{Z}_5$-hyperconifold [16, 17]. Show that it is the $\mathbb{Z}_5$-quotient of the conifold.*

Part II
Global Aspects

4 Coordinate Patches and Compact Varieties

To build interesting manifolds, one needs to patch together the (by themselves) rather boring local charts. For toric varieties, there is actually much more structure in the local charts as they can be very complicated singularities. Still, there is no topology: each affine toric variety corresponding to a strictly convex dual cone is a conical singularity and can be contracted to a point. To get any non-trivial topology, we need to patch together local affine toric varieties.

As an example, consider the projective plane $\mathbb{P}^2$, which happens to be also a toric variety. In terms of homogeneous coordinates $[z_0 : z_1 : z_2] = [\lambda z_0 : \lambda z_1 : \lambda z_2]$, it is covered by three affine patches

$$\begin{aligned} U_0 &= \big\{[1 : z_1 : z_2] \;\big|\; z_1, z_2 \in \mathbb{C}\big\}, \\ U_1 &= \big\{[z_0 : 1 : z_2] \;\big|\; z_0, z_2 \in \mathbb{C}\big\}, \\ U_2 &= \big\{[z_0 : z_1 : 1] \;\big|\; z_0, z_1 \in \mathbb{C}\big\}. \end{aligned} \tag{38}$$

To specify how the charts are glued together, we can either define gluing maps $\varphi_{ij} : U_i \cap U_j \to U_i \cap U_j$ or write each patch as a scheme and "patch" the defining rings. For example, starting with coordinates $(x, y) \in U_0$,

$$\begin{aligned} \varphi_{01}(x, y) &= [1 : x : y] = [\tfrac{1}{x} : 1 : \tfrac{y}{x}] = \big(\tfrac{1}{x}, \tfrac{y}{x}\big) \in U_1 \\ \varphi_{02}(x, y) &= [1 : x : y] = [\tfrac{1}{y} : \tfrac{x}{y} : 1] = \big(\tfrac{1}{y}, \tfrac{x}{y}\big) \in U_2. \end{aligned} \tag{39}$$

Hence, we are led to identify the patches as affine schemes[9]

$$U_0 = \operatorname{Spec}\big(\mathbb{C}[x, y]\big), \quad U_1 = \operatorname{Spec}\big(\mathbb{C}\big[\tfrac{1}{x}, \tfrac{y}{x}\big]\big), \quad U_2 = \operatorname{Spec}\big(\mathbb{C}\big[\tfrac{1}{y}, \tfrac{x}{y}\big]\big). \tag{40}$$

We recognize these three affine patches as the affine toric varieties corresponding to the three dual cones in Figure 7. Perhaps surprisingly, if we dualize the dual cones $\sigma_0^\vee$, $\sigma_1^\vee$, $\sigma_2^\vee \in M_\mathbb{Q}$ to get the cones σ_0, σ_1, $\sigma_2 \in N_\mathbb{Q}$ then they fit together nicely, see Figure 8!

What is so special about the cones fitting together as in Figure 8? Note that any two 2-dimensional cones σ_i and σ_j intersect in a 1-dimensional cone σ_{ij}. The corresponding affine toric variety

$$\operatorname{Spec}\big(\mathbb{C}[\sigma_{01}^\vee \cap M]\big) = \operatorname{Spec}\big(\mathbb{C}[x, x^{-1}, y]\big) = \mathbb{C}^\times \times \mathbb{C} = U_0 \cap U_1 \tag{41}$$

is precisely the overlap of the two charts $\operatorname{Spec}(\mathbb{C}[\sigma_0^\vee \cap M])$ and $\operatorname{Spec}(\mathbb{C}[\sigma_1^\vee \cap M])$. Finally, there is a triple overlap $\sigma_{012} = \{0\}$, which is also a rational polyhedral cone. The associated toric variety

$$\operatorname{Spec}\big(\mathbb{C}[\sigma_{012}^\vee \cap M]\big) = \big(\mathbb{C}^\times\big)^d \tag{42}$$

is precisely the maximal torus. In general, if and only if the cones fit together then the overlap is a lower-dimensional toric variety times a torus factor $(\mathbb{C}^\times)^k$. This is crucial for matching the torus action on each patch together to a torus action onto glued variety.

[9] The notation $\mathbb{C}[\frac{1}{x}]$ means simply polynomials in $\frac{1}{x}$.

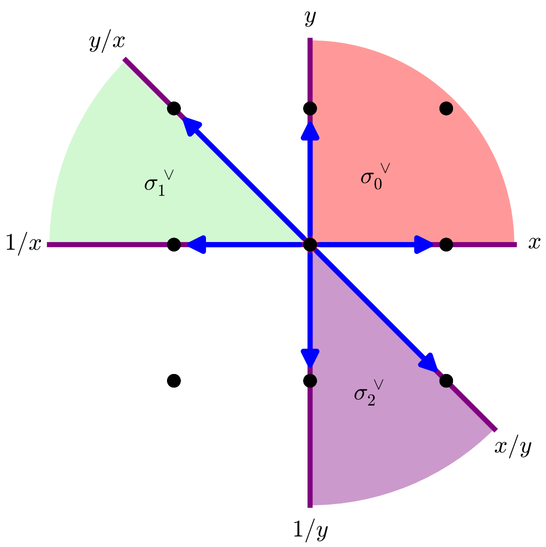


Figure 7: *Dual cones (in M) defining the three affine patches of $\mathbb{P}^2$.*

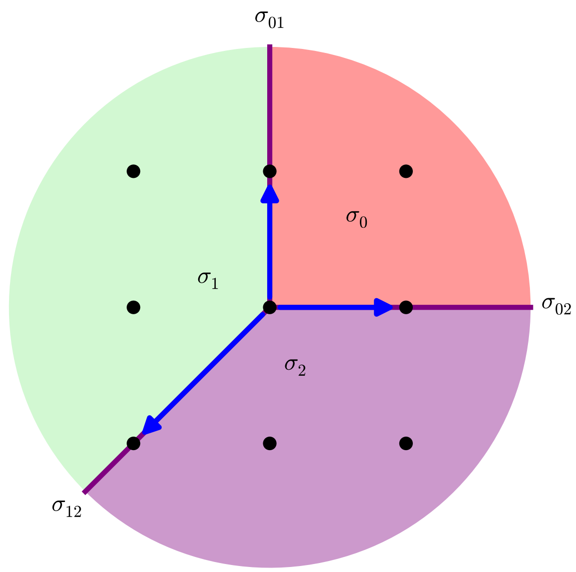


Figure 8: *The cones (in N) defining the three affine patches of $\mathbb{P}^2$.*

5 Toric Varieties

5.1 Fans

A fan is the generalization of Figure 8 for arbitrary cones:

Definition 13 (Fan). *A fan Σ is a finite set of strict convex polyhedral cones $\sigma \subset N_{\mathbb{Q}}$ such that*

1. *for each cone $\sigma \in \Sigma$, each face is also a cone of Σ.*
2. *any two cones σ, $\rho \in \Sigma$ intersect in a common face.*

So each fan contains cones of various dimension. The cones that generate the fan by taking faces and intersections are It is common usage to denote the lattice intersecting the fa**N** by N and its dual lattice, corresponding to **M**onomials) by M.

Sage implements various ways to define a fan. The most straightforward one is to specify the generating cones:

```
sage: P2_fan = Fan([Cone([(1,0),(0,1)]), Cone([(0,1),(-1,-1)]),
    Cone([(-1,-1),(1,0)])])
sage: P2_fan
Rational polyhedral fan in 2-d lattice N
sage: P2_fan.ngenerating_cones()
3
sage: c0, c1, c2 = P2_fan.generating_cones()
sage: c0
2-d cone of Rational polyhedral fan in 2-d lattice N
sage: c0.rays()
(N(0, 1), N(1, 0))
```

The fan can clearly be stratified by the dimension of the cones. The standard notation is that $\Sigma(k)$ denotes the subset of k-dimensional cones of the fan:

```
sage: P2_fan(1)
(1-d cone of Rational polyhedral fan in 2-d lattice N, 1-d cone of
    Rational polyhedral fan in 2-d lattice N, 1-d cone of Rational
    polyhedral fan in 2-d lattice N)
sage: P2_fan(0)
(0-d cone of Rational polyhedral fan in 2-d lattice N,)
```

Sage will also check that the given cones do indeed form a fan, and raise an error otherwise.

Exercise 11. *Use Sage to test which set of cones generates a fan:*

1. *in $\mathbb{Q}^2$, the first quadrant and the ray $(-1,-1)$.*
2. *in $\mathbb{Q}^2$, the first quadrant and the ray $(1,1)$.*
3. *in $\mathbb{Q}^2$, the first quadrant and the lower half plane.*
4. *in $\mathbb{Q}^3$, $\operatorname{span}\{(3,-1,0),(-1,-2,2),(3,0,-1)\}$ and $\operatorname{span}\{(-1,-1,1),\ (-1,-2,2),\ (2,-2,1)\}$. What is their intersection? What are their facets, that is, codimension-1 faces?*

If you have many generating cones then every ray of the fan tends to appear multiple times as a cone generator. This quickly becomes cumbersome, and it would be easier to specify the cones by the ray indices instead of having to repeat the coordinates of the rays over and over again. This is the other supported syntax for constructing an fan:

```
sage: rays = [(1,0), (0,1), (-1,-1)]
sage: cones = [ [0,1], [1,2], [2,0] ]
sage: alternate_P2_fan = Fan(cones, rays)
sage: alternate_P2_fan.is_equivalent(P2_fan)
True
```

5.2 Gluing

Each fan Σ determines a toric variety analogous to the example in Section 4. Every cone σ (irregardless of its dimension) is an d-dimensional affine toric variety $\mathbb{P}_\sigma$, and the relative position of the cones determines how they are glued together.

Definition 14 (Toric variety)**.** *A fan $\Sigma \in N_\mathbb{Q}$ defines a toric variety by gluing*

$$\mathbb{P}_\Sigma = \bigcup_{\sigma\in\Sigma} \operatorname{Spec}\big(\mathbb{C}[\sigma^\vee \cap M]\big). \tag{43}$$

In particular, the trivial cone $\langle\rangle$ corresponds to the algebraic torus $\mathbb{P}_{\langle\rangle} = (\mathbb{C}^\times)^d$ and is a dense Zariski-open subset. The algebraic torus acts on itself in the straightforward way, and this action extends to an action on the whole toric variety. One can show that the reverse is also true, and the definition is equivalent to

Definition 15 (Alternative definition of toric variety)**.** *A toric variety is a d-dimensional variety X that contains a dense Zariski-open algebraic torus $T_N = (\mathbb{C}^\times)^d \subset X$ such that the action of $T_\mathbb{C}$ on itself extends to an action on X.*

So we can also think of toric varieties as the (partial) compactifications of an algebraic torus, and this point of view is the origin of their name.

Exercise 12. *Consider the weighted projective space $\mathbb{P}^2[1,2,3]$. It is one of the example toric varieties in Sage, and you can construct it via* ***`toric_varieties.P2_123()`****. Describe the patches and their singularities.*

5.3 Torus Orbits

The trivial cone corresponds to the maximal torus $T_N = (\mathbb{C}^\times)^d$. The one-dimensional cones $\langle r_i\rangle$ define the toric variety

$$\mathbb{P}_{\langle r_i\rangle} = \operatorname{Spec}\big(\mathbb{C}[\langle r_i\rangle^\vee \cap M]\big) = \mathbb{C}\times\big(\mathbb{C}^\times\big)^{d-1} \tag{44}$$

that contains the maximal torus as a dense open subset and the $(d-1)$-dimensional orbit $\{0\}\times(\mathbb{C}^\times)^{d-1}$. We can continue this process and isolate a torus orbit for each cone of the fan. Note that the $(d-1)$-torus factor corresponds to evaluating the $(d-1)$ monomials perpendicular to the ray $\langle r_i\rangle$. This construction generalizes to

Definition 16 (Torus orbit)**.** *For each k-dimensional cone $\sigma \in \Sigma$, let*

$$O(\sigma) = \mathrm{Hom}\left(\sigma^\perp \cap M,\ \mathbb{C}^\times\right) \simeq \left(\mathbb{C}^\times\right)^{d-k} \subset \mathbb{P}_\Sigma \tag{45}$$

be the torus orbit associated to the cone σ.

In fact, these are the only torus orbits in the toric variety. The

Theorem 4 (Cone-orbit correspondence [15])**. ??** *There is a one-to-one correspondence between the cones $\sigma \in \Sigma$ and the torus orbits $O(\sigma)$.*

Explicitly writing down the torus orbits in general will be easier once we have introduced homogeneous coordinates in Section 7, and we will postpone further discussion until then. However, a few basic properties are clear:

- $O(\langle\rangle) = T_N \simeq (\mathbb{C}^\times)^d$,
- $O(\langle r_i \rangle) \simeq \{0\} \times (\mathbb{C}^\times)^{d-1}$, and
- $\dim O(\sigma) = d - \dim(\sigma)$ for all cones $\sigma \in \Sigma$.

Exercise 13. *Show that on $\mathbb{P}^2$ defined by the fan Figure 8, the three affine patches can be written as*[10]

$$U_\sigma = \bigcup_{\tau \leq \sigma} O(\tau). \tag{46}$$

5.4 Orbit Closures

Each $T_N = (\mathbb{C}^\times)^d$ orbit is necessarily of the form $(\mathbb{C}^\times)^k$ for some $k \leq d$. To get interesting subvarieties, we should compactify the torus orbits $O(\sigma)$ by adding limit points.

Definition 17 (Orbit closure)**.** *For any cone $\sigma \in \Sigma$, let $V(\sigma) = \overline{O(\sigma)}$ be the closure of the torus orbit associated to σ.*

This amounts to adding lower-dimensional torus orbits. The resulting orbit closure is again a toric variety, and its fan is determined by the cones $\tau \in \Sigma$ of the ambient fan that contain σ.

FIXME: define Star

Theorem 5. *For any cone $\sigma \in \Sigma$, the orbit closure is the toric variety*

$$V(\sigma) = \mathbb{P}_{\mathrm{Star}(\sigma)}. \tag{47}$$

Taking the orbit closure associated to a cone reverses inclusions,

$$\sigma > \tau \quad \Leftrightarrow \quad V(\sigma) \subset V(\tau) \qquad \forall \sigma, \tau \in \Sigma \tag{48}$$

In Figure 9, we draw the partially ordered set of cones and orbit closures on $\mathbb{P}^2$ to illustrate the inclusion-reversing correspondence.

[10] This is true in general.

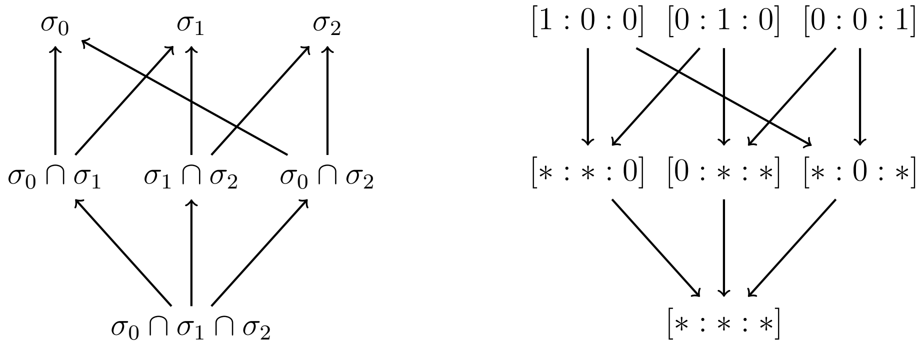


Figure 9: *Poset of cones and poset of orbit closures for* $\mathbb{P}^2$.

5.5 Lattice Polytopes

FIXME

define support, compact, face fan

Exercise 14. *Construct the fan of* $\mathbb{P}^2$ *from Figure 8 as the face fan of a polytope.*

5.6 Resolution of Singularities

subdivision = blow-up

Exercise 15. *Recall the* $\mathbb{Z}_5$ *hyperconifold from 10. In the worksheet FIXME:URL, you can find all 80 triangulations of the toric diagram. Every triangulation gives a partial resolution of the hyperconifold. How many triangulations yield complete resolutions ,that is, get rid of all singularities?*

6 Topology

6.1 Cartier and Weil Divisors

Especially in the physics literature, the two notions of divisor are often confused. They are

Definition 18 (Weil divisor)**.** *A Weil divisor is a formal* $\mathbb{Z}$*-linear combination of codimension-one algebraic subvarieties.*

and the other one is

Definition 19 (Cartier divisor)**.** *A Cartier divisor is a Weil divisor where every subvariety is locally cut out by a single meromorphic function.*

A special kind of a Cartier divisor is a *principal* divisor (f) which is a divisor that is cut out by a *global* meromorphic function f. Here, by "cut out" we mean that we count the algebraic subvarieties with their zero order or minus their pole order.

Example 5. *The meromorphic function $\frac{x}{y^2}$ on $\mathbb{C}^2 \ni (x,y)$ defines the principal divisor*

$$\left(\frac{x}{y}\right) = V(x) - 2V(y) \quad \in \mathrm{Div}\left(\mathbb{C}^2\right). \tag{49}$$

Now, clearly every Cartier divisor is Weil, so we find that principal $\subsetneq$ Cartier $\subset$ Weil. Are there Weil divisors that are not Cartier? Yes, as the following example demonstrates:

Example 6. *Consider the conifold $V(I)$ with $I = \langle ab + uv \rangle \subset \mathbb{C}[a,b,u,v]$. Then $D = \{a = 0, u = 0\}$ is a Weil divisor that is not Cartier.*

Exercise 16. *Check that D is indeed a codimension-one subvariety. What are local coordinates on D? Compare D with the Cartier divisors $a = 0$ and $u = 0$.*

However, if the variety is smooth then the notions of Weil and Cartier divisor coincide. This is why it is often not mentioned. But even if one is only interested in smooth varieties, they are often constructed as smooth subvarieties of singular ambient spaces. To summarize, there are three important types of divisors on a variety X:

- principal divisors $\mathrm{Div}_0(X)$,
- Cartier divisors $\mathrm{CDiv}(X)$, and
- Weil divisors $\mathrm{Div}(X)$.

They are almost always infinitely generated groups, that is, have no finite-dimensional $\mathbb{Z}$-basis. In general these groups can be difficult to study since they contain so much information. On a toric variety $\mathbb{P}_\Sigma$, however, there is a brutal yet effective way to make them finite-dimensional: Restrict to divisors that are torus orbits, that is,

- torus-invariant principal divisors $\mathrm{Div}_{0,T_N}(\mathbb{P}_\Sigma) = M$,
- torus-invariant Cartier divisors $\mathrm{CDiv}_{T_N}(\mathbb{P}_\Sigma)$, and
- torus-invariant Weil divisors $\mathrm{Div}_{T_N}(\mathbb{P}_\Sigma)$.

In particular, the torus-invariant Weil divisors are just the $\mathbb{Z}$-linear combination of the codimension-1 orbits:

$$\mathrm{Div}_{T_N}(\mathbb{P}_\Sigma) = \bigoplus_{\rho \in \Sigma(1)} \mathbb{Z} \cdot V(\rho). \tag{50}$$

6.2 Rational Equivalence

The Weil divisor group $\mathrm{Div}(X)$ and the Cartier divisor group $\mathrm{CDiv}(X)$ are almost always gigantic groups. For example, projective varieties admit families $f_t : X \to \mathbb{P}^1$, $t \in \mathbb{C}$ of meromorphic functions. Then every one of the uncountably many principal divisors (f_t) is an independent basis element in the two divisor groups. In order to cut this down to a manageable group, we mod out the principal divisors. This amounts to identifying divisors that can be continuously deformed within a one-parameter holomorphic family. The resulting quotient groups carry odd names:

- Divisor class group $\mathrm{Cl}(X) = \mathrm{Div}(X) \big/ \mathrm{Div}_0(X)$
- Picard group $\mathrm{Pic}(X) = \mathrm{CDiv}(X) \big/ \mathrm{Div}_0(X)$

We will have more to say about the Picard group in Subsection 8.1, but in the remainder of this section we will focus on the divisor class group.

Theorem 6. *If the fan $\Sigma \subset N_{\mathbb{Q}}$ is not contained in a hyperplane in $N_{\mathbb{Q}}$, then the sequence*

$$0 \longrightarrow M \xrightarrow{(\chi)} \mathrm{Div}_{T_N}(\mathbb{P}_\Sigma) \longrightarrow \mathrm{Cl}(\mathbb{P}_\Sigma) \longrightarrow 0 \tag{51}$$

is exact. The first arrow maps $m \in M$ to the principal divisor (χ^m).

In particular, the divisor class group of a toric variety is a finitely generated Abelian group. This also proves that an elliptic curve $\mathbb{C}/\Lambda$ cannot be a toric variety because its divisor class group is infinitely generated. A Weil divisor on a curve is a formal linear combination of points $D = \sum_{i \in I} a_i - \sum_{j \in J} b_j$. One invariant of the divisor class is the total number of points $|I| - |J| \in \mathbb{Z}$. But there is another continuous-valued invariant on an elliptic curve. Recall

Theorem 7 (Abel's theorem)**.** *Let $\mathbb{Z}^2 \simeq \Lambda \subset \mathbb{C}$ be a lattice and $E = \mathbb{C}/\Lambda$ the corresponding elliptic curve. There exists a meromorphic function on E with prescribed zeroes at $a_1, \dots, a_n \in \mathbb{C}$ and poles at $b_1, \dots, b_n \in \mathbb{C}$ if and only if*

$$a_1 + \dots + a_n = b_1 + \dots + b_n \mod \Lambda. \tag{52}$$

Hence, the sum in $\mathbb{C}$ of the points $a_1 + \dots + a_n - b_1 - \dots - b_n \in \mathbb{C}/\Lambda$ is also an invariant of the divisor class group of the elliptic curve $\mathbb{C}/\Lambda$. In fact, one can show that

$$\mathrm{Cl}\left(\mathbb{C}^2/\Lambda\right) \simeq \mathbb{Z} \times \left(\mathbb{C}^2/\Lambda\right) \tag{53}$$

6.3 Chow Group

Dividing out the rational equivalence of divisors gets rid of the "obvious" families of divisors that come from deforming the defining equations of the divisor by a principal divisor. One should ask oneself if one can repeat this with algebraic cycles (formal linear combinations of subvarieties) of arbitrary dimension. In fact, this is possible. Two k-dimensional algebraic cycles C_1, C_2 are rationally equivalent if they both live on a $(k+1)$-dimensional algebraic variety D and are rationally equivalent as divisors on $C_1, C_2 \in \mathrm{Div}(D)$.

Just like the divisor class group, the Chow group is an Abelian group. In general, it is not finitely generated. For toric varieties, there is a toric version of the Chow group where the generating cycles are the torus orbit closures and the rational equivalence is equivalence in torus-invariant families. This toric version is, by definition, finitely generated. And, similarly to the divisor class group, the toric Chow group equals the whole Chow group. Sage can compute the Chow group

```
sage: dP7 = toric_varieties.dP7()
```

```
sage: A = dP7.Chow_group()
sage: A
Chow group of 2-d CPR-Fano toric variety covered by 5 affine
   patches
sage: A.degree()
(Z, Z^3, Z)
```

and the Chow cycles associated to orbit closures $V(\sigma)$, $\sigma \in \Sigma$

```
sage: cone = dP7.fan(1)[3]
sage: A(cone)
( 0 | 0, 1, 0 | 0 )
```

The Chow group is very useful for intersection theory because it contains the maximal information about the algebraic cycles; Rational equivalence is the finest equivalence that one can reasonably impose. The downside, however, is that you can only intersect cycles that are transverse or can be made transverse. Sage implements the intersection of an arbitrary-dimension Chow cycle with a Cartier divisor, for which there exists a toric algorithm:

```
sage: a = A(cone)
sage: D = dP7.divisor(2)
sage: a.intersection_with_divisor(D)
( 1 | 0, 0, 0 | 0 )
```

Finally, when computing intersection numbers we often end up with a 0-cycle and want to count the number of points. This is done with the `count_points()` method:

```
sage: a.intersection_with_divisor(D).count_points()
1
sage: D1 =  dP7.divisor(2)
sage: D2 =  dP7.divisor(3)
sage: A(D1).intersection_with_divisor(D2).count_points()
1
```

6.4 The Cohomology Ring

Definition 20 (Stanley-Reisner ideal)**.** *Let $\Sigma \in N_{\mathbb{Q}}$ be a fan with rays $\Sigma(1) = \{\rho_1, \dots, \rho_r\}$. The Stanley-Reisner ideal (in the formal variables $x_1, \dots, x_r$) is the ideal*

$$\mathrm{SR}(\Sigma) = \left\langle x_{i_1} \cdots x_{i_k} \;\middle|\; \mathrm{span}\{\rho_{i_1}, \dots, \rho_{i_k}\} \notin \Sigma \right\rangle \quad \subset \mathbb{Z}[x_1, \dots, x_r]. \tag{54}$$

We write $\mathrm{SR}_{\mathbb{Q}}(\Sigma) = \mathrm{SR}(\Sigma) \otimes_{\mathbb{Z}} \mathbb{Q}$ for the analogous ideal with base ring $\mathbb{Q}$.

Theorem 8 (Cohomology of compact toric varieties [18, 19])**.** *Let $\Sigma \in N_{\mathbb{Q}}$ be a complete fan with $r = |\Sigma(1)|$ rays.*

- *if Σ is smooth,[11] then the integral cohomology ring of the toric variety is*

$$H^\bullet(\mathbb{P}_\Sigma, \mathbb{Z}) = \mathbb{Z}[x_1, \dots, x_r] \big/ \big(SR(\Sigma) + \mathrm{Lin}(\Sigma)\big) \tag{55}$$

  with all generators x_i having degree 2.

[11] A fan is smooth (simplicial) if every cone is.

- *if Σ is simplicial, then the rational cohomology ring of the toric variety is*

$$H^\bullet\big(\mathbb{P}_\Sigma, \mathbb{Q}\big) = \mathbb{Q}[x_1, \ldots, x_r] \big/ \big(SR_\mathbb{Q}(\Sigma) + \mathrm{Lin}_\mathbb{Q}(\Sigma)\big) \tag{56}$$

with all generators x_i having degree 2.

Suspiciously absent in the theorem is the case where Σ is not simplicial. In that case the cohomology ring is more complicated and there is no description in terms of a quotient of a polynomial ring. In particular, it need not be purely even-dimensional.

Exercise 17. *Compute* $\mathbb{Z}[x_1, \ldots, x_4]/(\mathrm{SR} + \mathrm{Lin})$ *for the conifold. Compare with the singular cohomology.*

Advantages: Fast
In the smooth case $H^{2k}(\mathbb{P}_\Sigma, \mathbb{Z}) = A_k(\mathbb{P}_\Sigma, \mathbb{Z})$, see [19].
Disadvantages: doesn't work for singular varieties

Part III
Divisors and Line Bundles

7 Homogeneous Coordinates

So far, we always constructed toric varieties by patching local charts. This is rather tedious, and one should try to find an analogue of homogeneous coordinates as used in projective spaces. In fact, such an analog exists and reduces to the usual homogeneous coordinates if your toric variety happens to be projective space. To generalize the homogeneous coordinates, consider a fan Σ with $|\Sigma(1)| = r$ rays (one-dimensional cones). We know that each ray corresponds to a T_N-invariant divisor, which should be cut out by setting one of the homogeneous coordinates to zero. So there should be r homogeneous coordinates and $r - d$ rescalings to produce a d-dimensional toric variety. You might remember that the divisor class group[12]

$$\mathrm{Cl}(\mathbb{P}_\Sigma) = A_{d-1}(\mathbb{P}_\Sigma) \simeq \mathbb{Z}^{r-d} \oplus (\text{finite group}), \tag{57}$$

which is suggestive that it might play a role. Finally, there must be some disallowed values for the homogeneous coordinates, otherwise they would parametrize a contractible space. Explicitly, the homogeneous coordinate construction of a toric variety is [20]

$$\mathbb{P}_\Sigma = \frac{\mathbb{C}^{\Sigma(1)} - Z}{\mathrm{Hom}\big(A_{d-1}(X),\ \mathbb{C}^\times\big)} \simeq \frac{\mathbb{C}^r - Z}{(\mathbb{C}^*)^{n-r} \times A_{d-1}(X)_{\mathrm{tors}}}, \tag{58}$$

where we still have to define the details of the quotient group action, the exceptional set Z, and what we mean by quotient.

Let me start with the quotient group action, and recall the short exact sequence eq. (51),

$$0 \longrightarrow M \xrightarrow{(\chi)} \mathrm{Div}_{T_N}(\mathbb{P}_\Sigma) \longrightarrow A_{d-1}(\mathbb{P}_\Sigma) \longrightarrow 0 \tag{59}$$

which dualizes to

$$0 \longrightarrow \mathrm{Hom}\big(A_{d-1}(\mathbb{P}_\Sigma),\ \mathbb{C}^\times\big) \longrightarrow \mathrm{Hom}\big(\mathrm{Div}_{T_N}(\mathbb{P}_\Sigma),\ \mathbb{C}^\times\big) \longrightarrow \mathrm{Hom}\big(M,\ \mathbb{C}^\times\big) \longrightarrow 0. \tag{60}$$

An element of $\mathrm{Hom}(\mathrm{Div}_{T_N}(\mathbb{P}_\Sigma),\ \mathbb{C}^\times)$ is a choice of multiplicative constant λ_i for each ray $\langle r_i \rangle \in \Sigma(1)$. It induces the trivial map $M \to \mathbb{C}^\times$ if

$$1 = \prod \lambda_i^{\langle r_i, m\rangle} \quad \forall m \in M. \tag{61}$$

In this case, it defines a map $A_{d-1}(\mathbb{P}_\Sigma) \to \mathbb{C}^\times$. Hence,

$$\mathrm{Hom}\big(A_{d-1}(\mathbb{P}_\Sigma),\ \mathbb{C}^\times\big) = \Big\{(\lambda_1, \dots, \lambda_r) \in (\mathbb{C}^\times)^{\Sigma(1)}\ \Big|\ 1 = \prod \lambda_i^{\langle r_i, e_j\rangle} \quad \forall e_j\Big\}, \tag{62}$$

where the e_j are a basis for the M-lattice.

[12] Here and in the following I will assume that the fan $\Sigma \subset N_\mathbb{Q}$ is not contained in a hyperplane. Geometrically, this means that the toric variety does not decompose as a product of a k-dimensional toric variety times a torus factor $(\mathbb{C}^\times)^{d-k}$. The reason is that, otherwise, the structure of the divisor class group is slightly more complicated, see Theorem 6.

Exercise 18. *Consider the face fan* Σ *of the lattice tetrahedron* $\nabla = \operatorname{conv}\{(-3,-2,4),$ $(0,1,0),$ $(1,0,0),$ $(2,1,-4)\}$. *Use Sage to compute* $A_2(\mathbb{P}_\Sigma) = \mathbb{Z}\times\mathbb{Z}_4$. *Find the two maps* $[D_1] \to \lambda$, $[D_2] \to \mu$ *corresponding to the two homogeneous rescalings*

$$\begin{aligned} \big[x_0 : x_1 : x_2 : x_3\big] &= \big[\lambda x_0 : \lambda x_1 : \lambda x_2 : \lambda x_3\big] \quad \forall \lambda \in \mathbb{C}^\times, \\ \big[x_0 : x_1 : x_2 : x_3\big] &= \big[x_0 : \mu x_1 : \mu^2 x_2 : \mu^3 x_3\big] \quad \forall \mu \in \{1, i, i^2, i^3\} \simeq \mathbb{Z}_4. \end{aligned} \tag{63}$$

Second, we need to find the exceptional set Z. This needs to be chosen such that the divisors $\{x_i = 0\}$, $i \in I$ only intersect if the corresponding orbit closures $V(\rho_i)$ do intersect. By the cone-orbit correspondence **??**, this is the case if and only if there exists a cone $\sigma \in \Sigma$ containing all rays $\rho_i \in \sigma$ for all $i \in I$. We saw a very similar condition already in the Stanley-Reisner ideal, 20. We can hence formulate the exceptional set as

$$Z = \bigcup_{(\prod_{i\in I} x_i)\in SR(\Sigma)} V\left(\langle x_i | i \in I\rangle\right) = \bigcup_{\substack{\{i_1,\cdots i_k\} \\ \operatorname{span}\{\rho_{i_1},\ldots,\rho_{i_k}\}\not\in\Sigma}} \big\{x_{i_1} = \cdots = x_{i_k} = 0\big\}. \tag{64}$$

Finally, the notion of quotient can be rather complicated in general.

The homogeneous coordinates make it easy to write down the orbit closures $V(\sigma)$. For example,

- the maximal torus the subset where all homogeneous coordinates are non-vanishing, and
- the divisors $V(\langle r_i\rangle)$, $\langle r_i\rangle \in \Sigma(1)$, are of the form $\{x_i = 0\}$.

In general, we can write the orbit-cone correspondence as the correspondence between the cone $\sigma = \langle r_{i_1}, \ldots, r_{i_n}\rangle$ and the orbit

$$V(\sigma) = \overline{O(\sigma)} = \big\{x_{i_1} = \cdots = x_{i_n} = 0\big\} \tag{65}$$

8 Sheaves

8.1 Line Bundles and Cartier Divisors

By definition, a Cartier divisor is equivalent to a local holomorphic function f_i on each affine patch U_i. The functions do not have to fit together to a global function, but can differ by a $\mathbb{C}^\times$-valued function

$$\varphi_{ij} = \frac{f_j}{f_i} : \quad U_i \cap U_j \to \mathbb{C}^\times. \tag{66}$$

Note that f_i can and will have zeroes and poles at the divisor, but they are at the same place as the zeroes and poles in f_j since they cut out the same divisor. So the quotient is, indeed, a well-defined function. But the collection of $\mathbb{C}^\times$-valued transition functions is nothing but the defining data of a holomorphic line bundle.

FIXME: isomorphism classes

Theorem 9. *If the fan $\Sigma \subset N_{\mathbb{Q}}$ is not contained in a hyperplane in $N_{\mathbb{Q}}$, then the sequence*

$$0 \longrightarrow M \xrightarrow{(\chi)} \mathrm{CDiv}_{T_N}(\mathbb{P}_\Sigma) \longrightarrow \mathrm{Pic}(\mathbb{P}_\Sigma) \longrightarrow 0 \tag{67}$$

is exact.

8.2 Support Functions

Both the divisor class group $\mathrm{Cl}(\mathbb{P}_\Sigma)$ and the Picard group $\mathrm{Pic}(\mathbb{P}_\Sigma)$ of a toric variety are finitely generated Abelian groups, that is, of the form $\mathbb{Z}^r \oplus \mathbb{Z}_{t_1} \oplus \cdots \oplus \mathbb{Z}_{t_k}$. But what is the precise relation between these groups? By definition, a Cartier divisor is a Weil divisor so there is an embedding

$$\mathrm{CDiv}(\mathbb{P}_\Sigma)\big/\,\mathrm{Div}_0(\mathbb{P}_\Sigma) = \mathrm{Pic}(\mathbb{P}_\Sigma) \;\subseteq\; \mathrm{Cl}(\mathbb{P}_\Sigma) = \mathrm{Div}(\mathbb{P}_\Sigma)\big/\,\mathrm{Div}_0(\mathbb{P}_\Sigma). \tag{68}$$

What is the defining data of a toric Cartier divisor? Let's take a closer look at the short exact sequence eq. (67). In general, a principal divisor is given by the zeroes and poles of a function. But a torus-invariant principal divisor is given by the zeroes and poles of a Laurent monomial, otherwise it would not be fixed by the torus action. We see that specifying a torus-invariant principal divisor amounts to picking $m \in M$. A torus-invariant Cartier divisor is given by a principal divisor $m_\sigma \in M$ on each affine patch $\mathrm{Spec}(\mathbb{C}[\sigma^\vee \cap M)$, fitting together on overlaps. What does this matching condition translate into for the m_σ? Note that $m_\sigma \in M$ can be thought of as a integral linear function on $\sigma \in N_Q$. It turns out that for the local principal divisors to fit together, the linear functions m_σ must fit together into a continuous piece-wise linear function on the fan. Therefore, we define

Definition 21. *Let $\Sigma \subset N_{\mathbb{Q}}$ be a fan. An integral support function is a function $f : |\Sigma| \to \mathbb{Q}$ such that*

1. *f is linear on each cone $\sigma \in \Sigma$.*
2. *$f(n) \in \mathbb{Z}$ for all lattice points $n \in |\Sigma| \cap N$.*

The set of all integral support functions is denoted $\mathrm{SF}(\Sigma, N)$*.*

The integral support functions form a finitely generated Abelian group under point-wise addition, which is canonically isomorphic to

$$\mathrm{SF}(\Sigma, N) = \mathrm{CDiv}_{T_N}(\mathbb{P}_\Sigma) \tag{69}$$

the torus-invariant Cartier divisor group. To find the underlying Weil divisor, just evaluate the function on the ray generator:

$$\mathrm{CDiv}_{T_N}(\mathbb{P}_\Sigma) \ni \quad D = \sum_{\rho \in \Sigma(1)} \langle m_\rho | \rho \rangle \cdot V(\rho) \quad \in \mathrm{Div}_{T_N}(\mathbb{P}_\Sigma) \tag{70}$$

Finally, we can mod out the (globally) principal divisors to obtain

Theorem 10 (Picard group). *The Picard group of a toric variety $\mathbb{P}_\Sigma$, $\Sigma \in N_\mathbb{Q}$, that is, the isomorphism classes of line bundles, is*

$$\mathrm{Pic}(\mathbb{P}_\Sigma) = \mathrm{SF}(\Sigma, N)/M, \tag{71}$$

the integral linear support functions modulo everywhere linear integral functions on N_Q. Moreover, the Picard group is torsion-free.

If every cone is smooth, then its easy to see that the Weil divisor data defines a integral support function. But once non-smooth cones appear, specifying the values on the rays may not define an integral support function. Analogous to the classification 1 of cones into smooth, simplicial, and everything else, we can distinguish three successively more singular cases.

Proposition 2.

- *If the fan Σ is smooth, then* $\mathrm{Pic}(\mathbb{P}_\Sigma) = \mathrm{Cl}(\mathbb{P}_\Sigma)$.
- *If the fan Σ is simplicial, then* $\mathrm{Pic}(\mathbb{P}_\Sigma) \subset \mathrm{Cl}(\mathbb{P}_\Sigma)$ *has finite index.*
- *If the fan Σ is simplicial, then* $\mathrm{Pic}(\mathbb{P}_\Sigma) \subset \mathrm{Cl}(\mathbb{P}_\Sigma)$ *is a sublattice of strictly lower rank.*

8.3 Global Sections

We found that Cartier divisor classes $\mathrm{Pic}(\mathbb{P}_\Sigma)$ correspond to line bundles. Generalizing to Weil divisor classes $\mathrm{Cl}(\mathbb{P}_\Sigma)$ yields the more general *reflexive sheaves*, which we can think of as line bundles with singularities. They share many properties of line bundles. In particular, the relation to homogeneous coordinates remains the same, that is, global sections of reflexive sheaves can be written as homogeneous polynomials. Every $\mathcal{O}(D)$ has lots of meromorphic sections, but by *global sections* we mean holomorphic sections, that is, without poles. For example, $\mathcal{O}_{\mathbb{P}^d}(n)$ has global sections (the degree-n homogeneous polynomials) for $n \geq 0$ but not for $n < 0$. If the divisor class group is of rank > 1 then the divisors with nonnegative coefficients define a cone:

Definition 22. *A Weil divisor $D = \sum a_i D_i$ is effective, written $D \geq 0$, if all coefficients $a_i \in \mathbb{Z}_\geq$ are nonnegative.*

This definition very much depends on the actual divisor and not just its class, for every effective divisor there is some rationally equivalent divisor that is not effective.

Because of the torus action, the global sections form a representations of the algebraic group $T_N = \mathrm{Hom}(M, \mathbb{C}^\times) \simeq (\mathbb{C}^\times)^d$. We can simplify the problem of finding global sections by restricting to a fixed irreducible representation, which are labelled by the lattice points $m \in M$. The associated irrep is the torus character χ^m, which is a function on the maximal torus $V(\langle\rangle) \simeq (\mathbb{C}^\times)^d$. However, it does not necessarily extend to a holomorphic function on the whole toric variety, as there may be poles along the $(d-1)$-dimensional torus orbits $V(\rho)$, $\rho \in \Sigma(1)$. Assuming that the toric variety is compact, if we take χ^m to be a global function then there must be poles somewhere. However, we can think of χ^m also as a local trivialization on the maximal torus of some line bundle $\mathcal{O}(D)$. Then it will depend on the transition function whether or not χ^m extends to a global section. This will be the case if D cancels the poles of the effective divisor (χ^m), that is, it extends to a global section of D if and only if $(\chi^m)+D \geq 0$. This condition is a finite set of linear constraints on the lattice points $m \in M$ that contribute a single global section each. Hence, the allowed region is a polyhedron

Definition 23 (The polyhedron of a divisor)**.** *Given a torus-invariant Weil divisor $D = \sum_\rho a_\rho V(\rho)$, let*

$$P_D \stackrel{def}{=} \big\{ m \in M_{\mathbb{Q}} \;\big|\; \langle m, u_\rho \rangle \geq -a_\rho \text{ for all } \rho \in \Sigma(1) \big\}. \tag{72}$$

The integral points in the polyhedron P_D are then the global sections of D:

Proposition 3. *Let D be a torus-invariant Weil divisor. Then the global sections of $\mathcal{O}(D)$ are*

$$\Gamma\big(\mathbb{P}_\Sigma, \mathcal{O}(D)\big) = \bigoplus_{(\chi^m)+D\geq 0} \mathbb{C}\cdot\chi^m = \bigoplus_{m\in P_D \cap M} \mathbb{C}\cdot\chi^m \tag{73}$$

Sage can compute the polyhedron of a divisor as well as convert the points $m \in M$ into homogeneous polynomials:

```
sage: X = toric_varieties.dP8()
sage: D = X.divisor([0,1,1,1])
sage: P_D = D.polyhedron()
sage: P_D.Vrepresentation()
[A vertex at (1, -1), A vertex at (2, -1), A vertex at (-1, 1), A
   vertex at (-1, 2)]
sage: P_D.integral_points()
[(1, -1), (2, -1), (-1, 1), (-1, 2), (0, 0), (1, 0), (0, 1)]
```

For convenience, you don't have to manually construct the polyhedron of the divisor. You can just use methods of the divisor to compute the global sections.

```
sage: D.sections()
(M(1, -1), M(2, -1), M(-1, 1), M(-1, 2), M(0, 0), M(1, 0), M(0, 1)
   )
sage: D.sections_monomials()
(y*z^2, t*z^3, x^2*y, t*x^3, x*y*z, t*x*z^2, t*x^2*z)
```

Note that the polyhedron P_D need not be full-dimensional. Clearly it can be empty or of any intermediate dimension. For example, consider

```
sage: X = toric_varieties.dP8()
sage: D1 = X.divisor([0,1,0,1])
sage: D1.polyhedron()
A 1-dimensional polyhedron in QQ^2 defined as the convex hull of 2
    vertices.
sage: D1.sections()
(M(1, -1), M(-1, 1), M(0, 0))
```

Exercise 19. *Find a divisor D on the weighted projective space $\mathbb{P}^2[1,2,3]$ such that P_D is not a lattice polytope.*

example: elliptic fibrations

8.4 Cohomology

9 Positivity

9.1 Ampleness

Kähler metric

$$\begin{aligned} g_{i\bar{\jmath}}(z,\bar{z}) &= \partial_i \bar{\partial}_{\bar{\jmath}} K(z,\bar{z}) \\ \omega = g_{i\bar{\jmath}}(z,\bar{z})\, \mathrm{d}z^i\, \mathrm{d}\bar{z}^{\bar{\jmath}} &= \partial\bar{\partial} K(z,\bar{z}). \end{aligned} \tag{74}$$

Kähler potential

$$K(z,\bar{z}) = \ln \sum_{\alpha,\bar{\beta}} h^{\alpha\bar{\beta}} s_\alpha \bar{s}_{\bar{\beta}}, \quad \mathrm{span}\{s_1,\dots\} = \Gamma(X, \mathcal{O}(nD)),\ n \gg 1 \tag{75}$$

Metric must be positive definite, which turns out to be a constraint for the divisor D.

Definition 24. *A Cartier divisor $D \in \mathrm{Pic}(\mathbb{P}_\Sigma)$ with corresponding function $\varphi_D \in SF(\Sigma, N)$ is ample if and only if φ_D is strictly convex.*

A strictly convex support function does not always exist! One tautological case where it does exists is when the cones of the fan are the linear regions of a strictly convex support function.

In particular, the resolution of a singularity $\hat{X} \to X$ corresponding to a subdivision of the fan may fail to be Kähler even if the singular variety X is. This is so because subdividing a cone can change the strictly convex condition on the facets of the cone, for example the conifold.

The existence of a Kähler resolution is guaranteed if the subdivision of each generating cone σ is induced from a strictly convex support function that is equal to zero on the facets $\partial\sigma$.

9.2 The Canonical Bundle

Consider a holomorphic $(d,0)$-form on a d-dimensional variety. In a local patch U_0, it is of the form

$$f_0(z_1,\dots,z_d)\, \mathrm{d}z_1 \wedge \dots \wedge \mathrm{d}z_d \tag{76}$$

The holomorphic transition functions and analytic continuation determine the coefficient f_i then in each other coordinate patch U_i. We can phrase this as saying that there is a rank-1 sheaf of $(d,0)$-forms. If the variety is smooth then this is a line bundle, but for singular varieties we have to admit more general sheaves. Specifying a different $f_0(z_1,\dots,z_d)$ in the first patch yields a different section of the same sheaf of $(d,0)$-forms. This sheaf is called the *canonical* sheaf (or bundle). On general grounds we can write it as $\mathcal{O}(K)$ for some Weil divisor class K, the canonical divisor class or just canonical class. The class of K is uniquely specified by the toric variety, but the actual divisor depends on the coefficient of the $(d,0)$-form in the initial patch.

The importance of the canonical class is that it determined purely by the intrinsic geometry of the variety. It is one of the most fundamental invariants of a variety, and the properties of the canonical divisor are important quantities. On a toric variety, you can find the canonical sheaf from the monad presentation of the tangent sheaf. The result is that

Theorem 11. *On the toric variety $\mathbb{P}_\Sigma$ defined by the fan $\Sigma \in N_\mathbb{Q}$, the canonical class is*

$$K_{\mathbb{P}_\Sigma} = - \sum_{\rho \in \Sigma(1)} V(\rho). \tag{77}$$

One class of varieties that you might have encountered before are called Fano:

Definition 25. *A variety X is called Fano if it is smooth and its anticanonical class $-K_X$ is ample.*

Note that some authors do not require a Fano to be smooth. I will use the convention that Fano implies smooth unless it is specified otherwiese.

The best-known Fano manifolds are the del Pezzo surfaces, which are $\mathbb{P}^1 \times \mathbb{P}^1$ and the blow-up of $\mathbb{P}^2$ in 0, ..., 8 sufficiently general points. The self-intersection of the canonical class is called the *degree*, it is 8 for $\mathbb{P}^1 \times \mathbb{P}^1$

```
sage: P1xP1 = toric_varieties.P1xP1()
sage: K = P1xP1.K()
sage: P1xP1.integrate( K.cohomology_class()^2 )
8
```

and $9 - k$ for the blow-up $\mathrm{Bl}_k(\mathbb{P}^2)$ of $\mathbb{P}^2$ at k points. The Fano surfaces of degree $(9-k) \leq 6 \Leftrightarrow k \leq 3$ are toric varieties. If $k \geq 4$, then the del Pezzo surface has complex structure moduli, while toric varieties are determined by purely combinatorial data and, therefore, have no continuous complex structure moduli. This is $\mathrm{Bl}_4(\mathbb{P}^2)$ and further blow-ups cannot be toric varieties.

FIXME: picture

9.3 Gorenstein

The Gorenstein property limits exactly how singular a variety can be, namely

Definition 26 (Gorenstein toric variety)**.** *The toric variety $\mathbb{P}_\Sigma$ is called Gorenstein if and only if K_Σ is a Cartier divisor.*

A smooth variety is Gorenstein because all Weil divisors are Cartier. But thats not true in general, and one can easily find examples of non-Cartier canonical divisors.

Exercise 20. *Find an example of a non-Gorenstein toric variety.*

It turns out that arbitrarily singular Fano varieties are often too badly behaved, and one needs to impose some sort of regularity. The good notion that allows some but not all singluarities turns out to be Gorenstein. Since Gorenstein also hints at non-singular varieties we implicitly allow certain compact singluar Fano varieties in

Definition 27. *A Gorenstein Fano variety is a compact variety X such that the anti-canonical divisor K_X is both Cartier and ample.*

The particular importance of Gorenstein Fano toric varieties is that they are the ones that are given by reflexive polytopes, which plays a crucial part in mirror symmetry.

Definition 28. *Let $\nabla \in N_{\mathbb{Q}}$ a polytope containing the origin. The polar polytope is*

$$\nabla^{\circ} = \left\{ m \in M_{\mathbb{Q}} \;\middle|\; \langle m, n \rangle \geq -1 \; \forall n \in \nabla \right\} \tag{78}$$

One can easily see that taking the polar twice yields the original polytope, $\nabla^{\circ\circ} = \nabla$. But note that the definition does not refer to the lattices. In fact, the polar of a lattice polytope is usually not a lattice polytope any more. For example,

```
sage: nabla = Polyhedron(vertices=[(-1,-1),(1,0),(0,2)])
sage: nabla.is_lattice_polytope()
True
sage: delta = nabla.polar()
sage: delta.is_lattice_polytope()
False
sage: delta.Vrepresentation()
[A vertex at (9/5, -3/5), A vertex at (-6/5, -3/5), A vertex at
   (-3/5, 6/5)]
```

In fact, in each dimension there are finitely many lattice polytopes whose polar is again a lattice polytope:

Definition 29. *A reflexive polytope is a lattice polytope whose polar is again a lattice polytope.*

Exercise 21. *Use Sage to draw the 16 reflexive polytopes in dimension 2. Hint: Use `ReflexivePolytopes(2)`. For each one, construct the toric variety defined by its face fan and check that the anticanonical divisor is Cartier and ample.*

We will come back to the relation with Calabi-Yau manifolds in Section 10.

9.4 Kahler and Mori Cone

For a 3-dimensional variety, the Kähler cone is the subset of the $J \in H^{1,1}(X)$ satisfying

$$\int_C J > 0, \qquad \int_S J \wedge J > 0, \quad \dots, \int_X J \wedge \cdots \wedge J > 0. \tag{79}$$

Part IV

Mirror Symmetry

10 Calabi-Yau Hypersurfaces

In order to compactify string theory from 10 to 4 dimensions, the space-time (at least far away from any black holes) should look like $\mathbb{R}^{3,1} \times M^{(6)}$, where $M^{(6)}$ is some 6-dimensional space. Of course the metric and other fields have to obey their equations of motion, and understanding the most general solution is still an active field of research. However, a particular subclass of admissible compactification manifolds is relatively well-understood, the Calabi-Yau manifolds. By definition, these are compact Kähler manifolds with vanishing first Chern class. For our purposes, can be characterized as the compact smooth subvarieties of a toric variety with trivial canonical bundle.

From the toric expression for the canonical divisor Theorem 11 it is clear that $\mathcal{O}(K)$ is never trivial. So toric varieties are never Calabi-Yau manifolds. Instead, we will construct Calabi-Yau manifolds from hypersurfaces in toric varieties. In general, the canonical class of a hypersurface $Y \subset X$ is governed by the adjunction formula

$$K_Y = \big(K_X + [Y]\big)\big|_Y \tag{80}$$

The easiest example would be quintic, that is, a degree-5 hypersurfaces in the projective space $\mathbb{P}^4$. The fan of $\mathbb{P}^4$ contains 5 rays, all linearly equivalent to each other. Therefore, there is only one divisor class, the hyperplane class $H \in \operatorname{Pic}(\mathbb{P}^4)$. The canonical class is

$$K_{\mathbb{P}^4} = -\sum_{\rho\in\Sigma(1)} D_\rho = -5H \tag{81}$$

To cancel the canonical class, the hypersurface must be in the class $[5H]$, that is, the zero set of a quintic homogeneous polynomial. Here is one particular quintic:

```
sage: P4 = toric_varieties.P(4)
sage: P4.anticanonical_hypersurface(monomial_points='vertices')
Closed subscheme of 4-d CPR-Fano toric variety covered by 5 affine
    patches defined by:
  a0*z0^5 + a2*z1^5 + a1*z2^5 + a3*z3^5 + a4*z4^5
sage: P4.anticanonical_hypersurface(monomial_points='vertices',
   coefficients=[1]*5)
Closed subscheme of 4-d CPR-Fano toric variety covered by 5 affine
    patches defined by:
  z0^5 + z1^5 + z2^5 + z3^5 + z4^5
```

But the most generic hypersurface has 126 monomials. Minus $\dim GL(5, \mathbb{C}) = 25$ rescalings leaves 101 complex structure moduli. Moreover, there is

```
sage: P4.Kaehler_cone()
1-d cone in 1-d lattice
```

a single Kähler modulus.

11 Periods and Picard-Fuchs Equations

The canonical class being trivial, $K_Y = 0$, means that there is a nowhere vanishing holomorphic section of $\mathcal{O}(K_Y) = \mathcal{O}$. So in local coordinates there is a $(3,0)$ form

$$\Omega = \Omega(z_1, z_2, z_3)\ \mathrm{d}z_1 \wedge \mathrm{d}z_2 \wedge \mathrm{d}z_3 \tag{82}$$

whose coefficient does not vanish anywhere on the Calabi-Yau manifold Y.

12 The Mirror Map

Bibliography

[1] D. E. Knuth, *The art of computer programming*, vol. 3. Addison-Wesley Longman Publishing Co., Boston, MA, USA, 2nd ed., 1998. 1.1

[2] T. Becker and V. Weispfenning, *Gröbner bases*, vol. 141 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1993. A computational approach to commutative algebra, In cooperation with Heinz Kredel. 1.4, 3

[3] H. Derksen and G. Kemper, *Computational invariant theory*. Invariant Theory and Algebraic Transformation Groups, I. Springer-Verlag, Berlin, 2002. Encyclopaedia of Mathematical Sciences, 130. 1.4

[4] W. Bosma, J. Cannon, and C. Playoust, "The Magma algebra system. I. The user language," *J. Symbolic Comput.* **24** (1997), no. 3-4, 235–265. Computational algebra and number theory (London, 1993). 2.1

[5] F. Berchtold, J. Hausen, and M. Widmann, "TorDiv," `http://www.mfo.de/organisation/institute/hausen/TorDiv/index.html`. 2.1

[6] D. R. Grayson and M. E. Stillman, "Macaulay2, a software system for research in algebraic geometry." Available at `http://www.math.uiuc.edu/Macaulay2`. 2.1

[7] G.-M. Greuel, G. Pfister, and H. Schönemann, "Singular 3.0," a computer algebra system for polynomial computations, Centre for Computer Algebra, University of Kaiserslautern, 2005. `http://www.singular.uni-kl.de`. 2.1

[8] The GAP Group, *GAP – Groups, Algorithms, and Programming, Version 4.4.12*, 2008. `http://www.gap-system.org`. 2.1

[9] D. Joyner, "GAP package toric," `http://www.opensourcemath.org/toric`. 2.1

[10] W. Stein *et al.*, *Sage Mathematics Software (Version 4.6.2)*. The Sage Development Team, 2011. `http://www.sagemath.org`. 2.1

[11] F. Pérez and B. E. Granger, "IPython: a System for Interactive Scientific Computing," *Comput. Sci. Eng.* **9** (May, 2007) 21–29. 5

[12] W. Stein and P. Jipsen, "Sage Quick Reference," `http://wiki.sagemath.org/quickref`. 2.3

[13] G. van Rossum, *The Python Tutorial.* `http://docs.python.org/tutorial`. 2.5

[14] D. Eisenbud and J. Harris, *The Geometry of Schemes*, vol. 197 of *Grad. Texts in Math.* Springer-Verlag, New York, 2000. 3.1

[15] D. A. Cox, J. B. Little, and H. Schenck, *Toric Varieties.* American Mathematical Society, 2011. 3.3, 4

[16] R. Davies, "Quotients of the conifold in compact Calabi-Yau threefolds, and new topological transitions," `0911.0708`. 10

[17] R. Davies, "Hyperconifold Transitions, Mirror Symmetry, and String Theory," `1102.1428`. 10

[18] W. Fulton, *Introduction to Toric Varieties.* Princeton University Press, 1993. 8

[19] V. I. Danilov, "THE GEOMETRY OF TORIC VARIETIES," *Russian Mathematical Surveys* **33** (1978), no. 2, 97. 8, 6.4

[20] D. A. Cox, "The homogeneous coordinate ring of a toric variety," *J. Algebraic Geom.* **4** (1995), no. 1, 17–50. 7

[21] P. Berglund, S. H. Katz, and A. Klemm, "Mirror symmetry and the moduli space for generic hypersurfaces in toric varieties," *Nucl.Phys.* **B456** (1995) 153–204, `hep-th/9506091`.