

## Polyhedra: a package for computations with convex polyhedral objects

RENÉ BIRKNER

ABSTRACT. Convex polyhedral objects play an important role in numerous areas of algebraic geometry. The package *Polyhedra* enables one to work with such convex polyhedral objects in *Macaulay 2*. In particular, *Polyhedra* implements objects representing polyhedra, cones, and fans. Many standard operations, such as intersections, duality, convex hulls, and calculation of Hilbert bases are included. The package also provides functions to create convex polyhedral objects from some algebraic objects.

THE PACKAGE AND ITS APPLICATIONS. Many objects in algebraic geometry can be described by convex polyhedral objects. For example, the Gröbner degenerations of a homogeneous ideal are in one-to-one correspondence with the faces of its state polytope; see [St, Chapter 2]. Geometric invariant theory (GIT) quotients coming from linearizations of the trivial bundle provide a second class of examples. For an effective action of the torus  $(\mathbb{C}^*)^k$  on affine space  $\mathbb{A}_{\mathbb{C}}^n$ , there is an associated exact sequence of lattices  $0 \rightarrow K \rightarrow \mathbb{Z}^n \rightarrow \mathbb{Z}^k \rightarrow 0$ . The GIT-quotients of this action are given by the normal fans of certain polyhedra in the associated  $\mathbb{Q}$ -vector space  $K_{\mathbb{Q}}$ ; for more information see [B, §III.2]. The package *Polyhedra* provides methods within *Macaulay 2* [M2] for creating and manipulating convex polyhedral objects such as polyhedra and cones. Some objects from algebraic geometry can also be translated into their corresponding convex polyhedral objects.

*Polyhedra* implements three new data types in *Macaulay 2*: Polyhedron, Cone, and Fan. The first two types of objects can be created from scratch by entering a set of vertices, a set of rays, and a lineality space. Dually, polyhedra and cones can be created by intersecting half-spaces and hyperplanes. New objects also arise from the convex hull, positive hull, or intersection of other polyhedra. Fans are constructed by specifying a collection of cones or by using functions that transform other objects into fans, such as `normalFan`.

This package contains most of the standard operations on convex polyhedral objects; for details see [Z]. For a Cone, *Polyhedra* allows the user to check smoothness, create the dual cone, compute all its faces, or determine its Hilbert basis. For a Polyhedron, functions are included to compute the polar polyhedron, create the normal fan, check for compactness, and find the lattice points of a compact polyhedron. The package provides methods for computing direct sums and calculating affine images or preimages for all three types of convex polyhedral objects. The user can also conveniently access some of the most common polyhedral objects such as the positive orthant, the hypercube, or the cross polytope.

In *Polyhedra*, every convex polyhedral object is represented using the double description method. This means that for a polyhedron, the vertices and rays as well as the defining affine half-spaces and

---

2000 *Mathematics Subject Classification*. 52B55, 52-04, 52B20, 52C07, 14M25.  
*Polyhedra* version 1.0.5.

hyperplanes are part of the representation. For a cone, the representation is given by the linear half-spaces and hyperplanes in addition to the rays and the lineality space. If only one of the descriptions is given, then the dual description is automatically computed using the Fourier-Motzkin algorithm which is available within *Macaulay 2* via the *FourierMotzkin* package [Sm]. A fan is stored as the list of generating cones, i.e. all cones in the fan that are not a face of any other cone in the fan.

Through the basic functionality for convex geometry it provides, the package *Polyhedra* promises to form the basis for a number of further applications. Relying heavily on *Polyhedra*, the *ToricVB* package [BIP] computes the cohomology of equivariant vector bundles following the combinatorial approach given in [K]. Polyhedral divisors, which describe toric varieties with action by some subtorus and are defined in [AH], can also be calculated by using *Polyhedra*. Finally, the *toricCodes* package [I] utilizes *Polyhedra* to generate toric codes and *T*-codes.

EXAMPLES. Consider a regular function  $f$  on the 3-dimensional torus  $(\mathbb{C}^*)^3$ . This function  $f$  can be expressed as a Laurent polynomial in 3 variables. One important invariant of such a polynomial is its Newton polytope  $P$ ; see [Z, Exercise 9.4]. *Polyhedra* can compute this invariant as follows.

In the polynomial ring

```
i1 : loadPackage "Polyhedra";
i2 : printWidth = 82;
i3 : R = QQ[x,y,z];
```

consider the polynomial

```
i4 : f = x*z^3 - x^2*z^2 + y*z^3 - x^3*y + x*y^2*z - x^2*y^2;
```

If we compute the Newton polytope of  $f$ , we see that it is a hexagon in  $\mathbb{Q}^3$ :

```
i5 : P = newtonPolytope f
o5 = {ambient dimension => 3
      dimension of lineality space => 0
      dimension of polyhedron => 2
      number of facets => 6
      number of rays => 0
      number of vertices => 6
o5 : Polyhedron
```

The hypersurface  $Z = \mathbf{V}(f) \subset (\mathbb{C}^*)^3$  can be compactified by considering its closure in the embedding of  $(\mathbb{C}^*)^3$  into the projective toric variety  $X$  corresponding to the polytope  $P$ . The affine invariant charts of  $X$  correspond to cones in the normal fan of  $P$ . Since  $I = \langle f \rangle$  is a principal ideal in  $R$ ,  $P$  is in fact a state polytope of  $I$ ; see [St, Chapter 2]. Hence, the normal fan of  $P$  is the Gröbner fan of  $I$ .

```
i6 : F = normalFan P
o5 = {ambient dimension => 3
      number of generating cones => 6
      number of rays => 6
      top dimension of the cones => 3
o6 : Fan
```

We can check whether the toric variety  $X$  is smooth by checking the smoothness of the fan  $F$ .

```
i7 : isSmooth F
o7 = false
```

Two of the six generating cones are not smooth.

```

i8 : C = select(genCones F, c -> not isSmooth c)
o8 = {{ambient dimension => 3          }, {ambient dimension => 3          }}
      dimension of lineality space => 1  dimension of lineality space => 1
      dimension of the cone => 3         dimension of the cone => 3
      number of facets => 2              number of facets => 2
      number of rays => 2                number of rays => 2
o8 : List

```

These cones have the following rays and common lineality space:

```

i9 : (apply(C, rays), linSpace F)
o9 = ({{| 0 0 |, | 0 0 |}, | 1 |)
      | 0 -2 | | 0 2 | | 1 |
      | -1 -1 | | 1 1 | | 1 |
o9 : Sequence

```

Thus, by computing the Hilbert bases

```

i10 : HB = apply(C, hilbertBasis)
o10 = {{| 1 |, | 0 |, | 1 |}, {| -2 |, | 0 |, | -2 |}}
      | 0 | | 0 | | -1 | | -1 | | 0 | | 0 |
      | 0 | | -1 | | 0 | | -1 | | 1 | | -1 |
o10 : List

```

we get interior rays for each cone, which we can use to construct a smooth refinement of  $F$ . For this, we need the codimension 1 faces of the cone:

```

i11 : L = apply(C, x -> faces(1,x))
o11 = {{{ambient dimension => 3          }, {ambient dimension => 3          }},
      dimension of lineality space => 1  dimension of lineality space => 1
      dimension of the cone => 2         dimension of the cone => 2
      number of facets => 1              number of facets => 1
      number of rays => 1                number of rays => 1
-----
      {{ambient dimension => 3          }, {ambient dimension => 3          }}
      dimension of lineality space => 1  dimension of lineality space => 1
      dimension of the cone => 2         dimension of the cone => 2
      number of facets => 1              number of facets => 1
      number of rays => 1                number of rays => 1
o11 : List

```

For each face, we take the corresponding Hilbert basis element that is not a ray of the original cone and compute the convex hull of this element and the face:

```

i12 : L = apply(2, i -> apply(L#i, C -> posHull{C, HB#i#0}))
o12 = {{{ambient dimension => 3          }, {ambient dimension => 3          }},
      dimension of lineality space => 1  dimension of lineality space => 1
      dimension of the cone => 3         dimension of the cone => 3
      number of facets => 2              number of facets => 2
      number of rays => 2                number of rays => 2
-----

```

```

{{ambient dimension => 3          }, {ambient dimension => 3          }}}
  dimension of lineality space => 1  dimension of lineality space => 1
  dimension of the cone => 3         dimension of the cone => 3
  number of facets => 2             number of facets => 2
  number of rays => 2              number of rays => 2

```

```
o11 : List
```

We now add these cones to the smooth cones of  $F$  to construct a new fan  $F1$ .

```
i13 : F1 = makeFan flatten(select(genCones F, c -> isSmooth c)|L)
```

```
o13 = {ambient dimension => 3          }
      number of generating cones => 8
      number of rays => 8
      top dimension of the cones => 3

```

```
o13 : Fan
```

This fan is a refinement of  $F$ ; the corresponding toric variety  $X'$  is an equivariant blowup of  $X$  in two torus fixed points. We can check that  $X'$  is in fact smooth by checking the smoothness of  $F1$ .

```
i14 : isSmooth F1
```

```
o14 = true
```

#### REFERENCES.

- [AH] K. Altmann and J. Hausen, *Polyhedral divisors and algebraic torus actions*, Math. Ann. **334** (2006), no. 3, 557 – 607.
- [B] R. Birkner, *Quotients of closed subvarieties in toric varieties*, Diploma thesis, Freie Universität Berlin, 2007.
- [BIP] R. Birkner, N. Ilten, and L. Petersen, *ToricVB, a Macaulay 2 package for computing Čech cohomology of vector bundles on toric varieties*. available at [page.mi.fu-berlin.de/rbirkner/docsen.htm](http://page.mi.fu-berlin.de/rbirkner/docsen.htm).
- [I] N. Ilten, *toricCodes, a Macaulay 2 package for computing toric codes*, available at [people.cs.uchicago.edu/~nilten](http://people.cs.uchicago.edu/~nilten).
- [M2] D.R. Grayson and M.E. Stillman, *Macaulay 2, a software system for research in algebraic geometry*, available at [www.math.uiuc.edu/Macaulay2/](http://www.math.uiuc.edu/Macaulay2/).
- [Sm] G.G. Smith, *FourierMotzkin, a Macaulay 2 package for convex hull and vertex enumeration*, available at [www.math.uiuc.edu/Macaulay2/Packages/](http://www.math.uiuc.edu/Macaulay2/Packages/).
- [St] B. Sturmfels, *Gröbner bases and convex polytopes*, University Lecture Series, vol. 8, American Mathematical Society, Providence, RI, 1996.
- [K] T. Kaneyama, *On equivariant vector bundles on an almost homogeneous variety*, Nagoya Math. J. **57** (1975), 65 – 86.
- [Z] G.M. Ziegler, *Lectures on polytopes*, Graduate Texts in Mathematics, vol. 152, Springer-Verlag, New York, 1995.

RECEIVED : 2009-03-30

REVISED : 2009-07-30

ACCEPTED : 2009-09-07

[rbirkner@math.fu-berlin.de](mailto:rbirkner@math.fu-berlin.de) : Mathematisches Institut, Freie Universität Berlin, Arnimallee 3, 14195 Berlin, Germany.