

What Is OSCAR?

Michael Joswig

TU Berlin & MPI MiS, Leipzig

OSCAR Boot Camp, 03 July 2024

joint w/ W. Decker, C. Fieker, M. Horn
and many others

1 First Few Examples

2 Inner Workings

Project organization



Installation

3 Application: Triangulating a K3 Surface

4 Application: Galois Groups of Ehrhart Polynomials

OSCAR = Open Source Computer Algebra Resource

Solving a Polynomial System, I

OSCAR v1.1.0 (21 June 2024)

```
julia> R, (u,v) = polynomial_ring(QQ, ["u","v"])  
(Multivariate polynomial ring in 2 variables over QQ,  
 ↪ QQMPolyRingElem[u, v])
```

```
julia> I = ideal(R, [(u+1)^2-v^3, v-u^2-1]);
```

```
julia> G = groebner_basis(I, ordering=lex(R))
```

Gröbner basis with elements

1 -> $v^6 - 2*v^4 + v^2 - 4*v + 4$

2 -> $2*u - v^3 + v$

with respect to the ordering

$\text{lex}([u, v])$

Solving a Polynomial System, II

OSCAR v1.1.0 (21 June 2024)

```
julia> S, x = QQ["x"]  
(Univariate polynomial ring in x over QQ, x)
```

```
julia> phi = hom(R, S, [0, x]);
```

```
julia> roots(QQBarField(), phi(G[1]))  
6-element Vector{QQBarFieldElem}:  
Root 1.37504 of  $x^5 + x^4 - x^3 - x^2 - 4$   
Root 1.00000 of  $x - 1$   
Root  $0.243924 + 1.05124im$  of  $x^5 + x^4 - x^3 - x^2 - 4$   
Root  $0.243924 - 1.05124im$  of  $x^5 + x^4 - x^3 - x^2 - 4$   
Root  $-1.43145 + 0.669930im$  of  $x^5 + x^4 - x^3 - x^2 - 4$   
Root  $-1.43145 - 0.669930im$  of  $x^5 + x^4 - x^3 - x^2 - 4$ 
```

Automorphism Group of a Polytope

OSCAR v1.1.0 (21 June 2024)

```
julia> P = icosahedron()  
Polytope in ambient dimension 3 with  
↪ EmbeddedAbsSimpleNumFieldElem type coefficients
```

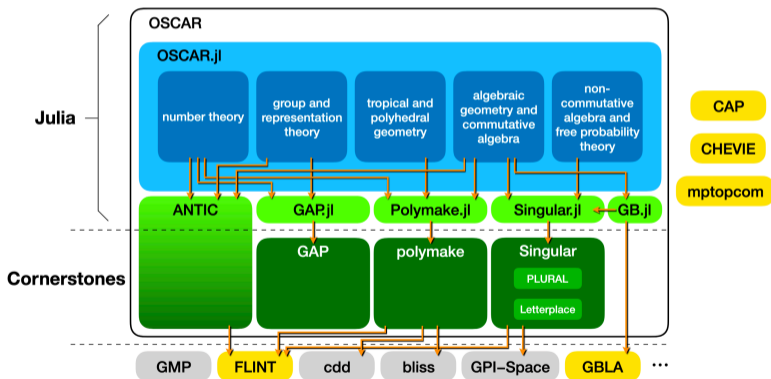
```
julia> G = automorphism_group(P)[:on_vertices]  
Permutation group of degree 12
```

```
julia> describe(G)  
"C2 x A5"
```

Organization

<http://oscar-system.org/>

- joint software project of the CRC TRR 195, funded by DFG
 - written in Julia
 - planned duration: 2017–2028, three phases (current version: 1.1)



Documentation, Part I

<https://docs.oscar-system.org/dev/>

OSCAR

SYMBOLIC TOOLS

Oscar.jl

- Welcome to OSCAR
- General >
- Groups >
- Rings >
- Fields >
- Linear Algebra >
- Number Theory >
- Polyhedral Geometry >
- Commutative Algebra >
- Invariant Theory >
- Algebraic Geometry >

Welcome to OSCAR

[GitHub](#) [✉](#) [⚙](#) [^](#)

Welcome to OSCAR

OSCAR is a new computer algebra system. OSCAR features functions for groups, rings, and fields as well as linear and commutative algebra, number theory, algebraic and polyhedral geometry, and more. It is built upon several well established systems for mathematical research joined via the Julia programming language. Have a look at our [Architecture](#) page for a detailed overview and at our [installation instructions](#) for installing OSCAR.

If you have questions about OSCAR, please have a look at our [Frequently Asked Questions](#) and feel free to contact us under the channels mentioned on [our community page](#). Our main communication channels are [Slack](#) and [Github](#).

If you are a new developer or interested in developing OSCAR, have a look at our [Introduction for new developers](#).

If you have used OSCAR in the preparation of a paper, please cite it as described [here](#).

[Architecture](#) »

Powered by [Documeneter.jl](#) and the [Julia Programming Language](#).



Documentation, Part II

<https://docs.oscar-system.org/dev/>

OSCAR
SYMBOLIC TOOLS
Oscar.jl

Search docs (Ctrl + /)

- Linear Algebra >
- Number Theory >
- Polyhedral Geometry ▾
 - Introduction
 - Polyhedra >
 - Cones
 - Polyhedral Fans
 - Polyhedral Complexes
 - Linear Programs
 - Introduction
 - Constructions
 - Solving a linear program - an example

Polyhedral Geometry / Linear Programs

[GitHub](#) [📄](#) [⚙️](#) [^](#)

Linear Programs

Introduction

The purpose of a linear program is to optimize a linear function over a polyhedron.

Constructions

Linear programs are constructed from a polyhedron and a linear *objective function* which is described by a vector and (optionally) a translation. One can select whether the optimization problem is to maximize or to minimize the objective function.

linear_program - Function

```
linear_program(P, c; k = 0, convention = :max)
```

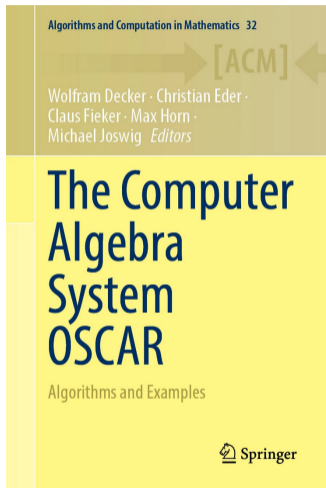
The linear program on the feasible set P (a Polyhedron) with respect to the function $x \mapsto \text{dot}(c,x)+k$.

Solving a linear program - an example

Let $P = [-1, 1]^3$ be the 3-dimensional cube in \mathbb{R}^3 , and consider the linear function ℓ , given by $\ell(x, y, z) =$



Documentation, Part III



Springer, Sep 2024

- four introductory chapters
 - commutative algebra and algebraic geometry
 - number theory
 - group theory
 - polyhedral geometry
- nine advanced chapters
 - elliptic fibrations of K3 surfaces
 - monomial bases in Lie theory
 - matroids
 - tropical implicitization
 - F-theory
 - ...

Julia

<https://julialang.org/>

- open source (MIT License)
- supports Linux, BSD, MacOS, Windows
- friendly C/Python-like (imperative) syntax
- JIT compilation: near C performance
- easy/efficient C interoperability; good C++ support
- designed by mathematically minded people
- OSCAR uses the same language for user facing and internal functions

Complete Installation (e.g., on Fedora Linux)

<https://www.oscar-system.org/install/>

Step 1: Install prerequisites

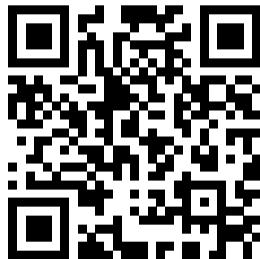
```
sudo dnf install gcc-c++ make
```

Step 2: Install Julia

```
curl -fsSL https://install.julialang.org | sh
```

Step 3: Install OSCAR

```
julia> using Pkg; Pkg.add("Oscar")  
julia> using Oscar
```



Exploring an Abstract Simplicial Complex

```
julia> K3 = simplicial_complex([[1,2,3,4,8],[1,2,3,4,12],[1,2,3,8,12]])  
Abstract simplicial complex of dimension 4 on 16 vertices
```

```
julia> size(facets(K3))  
(288,)
```

```
julia> describe(fundamental_group(K3))  
"1"
```

```
julia> is_manifold(K3)  
true
```

J., Lofano, Lutz & Tsuruga, J. Appl. Comput. Topol. **6** (2022)

Recognizing K3

```
julia> [ cohomology(K3, i) for i in 0:4 ]
```

```
5-element Vector{FinGenAbGroup}:
```

```
ℤ
```

```
ℤ/1
```

```
ℤ22
```

```
ℤ/1
```

```
ℤ
```

```
julia> Oscar.pm_object(K3).INTERSECTION_FORM
```

```
PropertyValue wrapping polymake::topaz::IntersectionForm
```

```
0 3 19
```

Freedman, J. Differ. Geom. **17** (1982)
J., Proc. ICMS 2002

Casella & Kühnel, Topology **40** (2001)
Spreer, PhD Thesis, U Stuttgart (2011)

The Ehrhart Polynomial of a Polytope

```
julia> C = cube(3)
```

```
Polytope in ambient dimension 3
```

```
julia> LC = ehrhart_polynomial(C); @show factor(LC);
```

```
factor(LC) = 1 * (2*x + 1)^3
```

```
julia> LC(1)
```

```
27
```

```
julia> LC(2)
```

```
125
```

```
julia> galois_group(LC)
```

```
(Permutation group of degree 3 and order 1, Galois context for  
↪ 2*x + 1 and prime 3)
```

Fano Simplices

```
julia> vertices(fano_simplex(2))
3-element SubObjectIterator{PointVector{QQFieldElem}}:
 [1, 0]
 [0, 1]
 [-1, -1]
```

```
julia> S = fano_simplex(14)
Polytope in ambient dimension 14
```

```
julia> is_smooth(normal_toric_variety(face_fan(S)))
true
```

```
julia> @time G = galois_group(ehrhart_polynomial(S))
 3.075857 seconds (8.26 M allocations: 537.105 MiB,
 4.21% gc time, 78.65% compilation time)
(Permutation group of degree 14 and order 645120,
Galois context for  $15x^{14} + 105x^{13} + 24115x^{12} + 143325x^{11} + 772471$ 
```

The Ehrhart Polynomial of the Fano Simplex

```
julia> Qt, t = QQ["t"]
(Univariate polynomial ring in t over QQ, t)

julia> LS(d) = binomial(t+d+1,d+1) - binomial(t,d+1)
LS (generic function with 1 method)

julia> roots(QQBarField(), LS(6))
6-element Vector{QQBarFieldElem}:
Root -0.500000 + 0.446648*im of  $7x^6 + 21x^5 + 385x^4 + 735x^3 + 2128x^2 + 2128x + 7$ 
Root -0.500000 - 0.446648*im of  $7x^6 + 21x^5 + 385x^4 + 735x^3 + 2128x^2 + 2128x + 7$ 
Root -0.500000 + 2.15772*im of  $7x^6 + 21x^5 + 385x^4 + 735x^3 + 2128x^2 + 2128x + 7$ 
Root -0.500000 - 2.15772*im of  $7x^6 + 21x^5 + 385x^4 + 735x^3 + 2128x^2 + 2128x + 7$ 
Root -0.500000 + 6.81137*im of  $7x^6 + 21x^5 + 385x^4 + 735x^3 + 2128x^2 + 2128x + 7$ 
Root -0.500000 - 6.81137*im of  $7x^6 + 21x^5 + 385x^4 + 735x^3 + 2128x^2 + 2128x + 7$ 

julia> describe(galois_group(LS(6))[1])
"C2 x S4"
```


Data Base of Smooth Fano Polytopes




Paffenholz, Lorenz, Øbro: polydb.org

```
julia> db = Polymake.Polydb.get_db();
julia> collection = db["Polytopes.Lattice.SmoothReflexive"];
julia> query = Dict("DIM"=>6);
julia> results = Polymake.Polydb.find(collection, query);

julia> PP6 = [ polarize(polyhedron(P)) for P in results ];
julia> E6 = [ ehrhart_polynomial(P) for P in PP6 ];

julia> MSet(order(galois_group(e)[1]) for e in E6)
MSet{Any} with 7622 elements:
 4   : 623
16   : 310
 6   :
12   : 44
 8   : 22
48   : 6622
```

Some References

-  Mara Belotti, Michael Joswig, Chiara Meroni, Victoria Schleis, and Johannes Schmitt, *Algebraic and geometric computations in OSCAR*, SIAM News **56** (2023), no. 07, 9–10.
-  Wolfram Decker, Christian Eder, Claus Fieker, Max Horn, and Michael Joswig (eds.), *The computer algebra system OSCAR: Algorithms and examples*, Algorithms and Computation in Mathematics, vol. 32, Springer, 2024.
-  Claus Fieker, Tommy Hofmann, and Michael Joswig, *Computing Galois groups of Ehrhart polynomials in OSCAR*, Sém. Lothar. Combin. **86B** (2022), Art. 87, 9.

<https://www.oscar-system.org/install/>

