High dimensional computation of fundamental domains

Mathieu Dutour Sikirić

March 21, 2023

Introduction

- Over many years, I have worked on polyhedral computation using symmetries.
- The fields considered are lattice theory, optimization, topology, group theory, number theory.
- Most of the computations were done using **GAP/Sage**, but **GAP** has several limitations (slowness, large memory usage, threading limitation).
- Thus I decided to rewrite most of what I need in **C++**:
  - The code is completely available on github and I contribute daily to it.
  - It is Open Source and everyone can contribute.
  - Installations issues are addressed with **dockerfile** which allow easy installs.

`https://github.com/MathieuDutSik/permutalib`
`https://github.com/MathieuDutSik/polyhedral_common`
`https://hub.docker.com/r/mathieuds/polyhedralcpp`

I. Library Design

Polyhedral and matrix functionality

- The code of **cdd** and **lrs** has been translated into **C++** by using a template parameter **T**. It allows to compute the dual description of polytopes that is the facets given the vertices.
- The linear programming has been coded as well.
- For removing redundancies in a set of defining inequalities, we have Clarkson method that uses linear programming very efficiently.
- We have similar functionality for matrices where we have functions for fields **T** and functions for rings **Tint** with Euclidean division.
- For the LLL algorithm we have functions with two template parameters:
  - **T** for the gram matrix entries.
  - **Tint** for the reduction basis.

Possible Numerical type

- The most classical case used is **mpq_class** that allows to have arbitrary precision arithmetic.
- Other types available are $\mathbb{Q}(\sqrt{d})$ for d a square free integer. Difficulty is in computing signs. We used the formula

$$a + b\sqrt{d} = \frac{a - b\sqrt{d}}{a^2 - b^2 d}$$

  to decide signs.
- We have a classic **C++** class **QuadField**<**Trat,int**>.
- For more complex real fields like $\mathbb{Q}(\alpha)$, e.g. with $\alpha = 2\cos(2\pi/7)$ of degree 3 we used following approach:
  - We used a type **C++** class **RealAlgebraic**<**Trat,int**> with the integer being the index pointing to a global variable that has the field description.
  - Addition, ok. Product with a loop.
  - Inverse requires solving a linear system.
  - Deciding signs is done by using N precomputed continuous fraction approximants that allow with interval arithmetic to decide the sign. If failing we have a clean error.

permutalib design goals

- For general groups, we have some algorithms but in general groups can be very wild. When algorithms exist, they invariably depend on some finite group subcalls.
- Practically for finite groups we use permutation groups. **GAP** has implementation of most functionality we may need.
- What we need for applications:
  - Computing the stabilizer of a set under a permutation group.
  - Testing if two sets are equivalent under a permutation group.
  - Finding the canonical form of the action of a group on sets.
  - Iterating over all group elements.
  - TODO: Double coset decomposition and testing group conjugacy.
- The **C++** implementation is based on **GAP** code and 3 to 100 times faster than **GAP**.
- There are other code by Christopher Jefferson (Vole) that provides some functionality in Rust.

Linear symmetry groups of a polyhedral cone

- Suppose C is a full-dimensional polyhedral cone generated by vectors $(v_i)_{1\leq i\leq N}$ in $\mathbb{R}^n$.
- The linear symmetry group $Lin(C)$ is the group of transformations $\sigma \in \mathrm{Sym}(N)$ such that there exist $A \in \mathrm{GL}_n(\mathbb{R})$ with $Av_i = v_{\sigma(i)}$ (There are other groups related to polyhedral cones)
- We define the form

$$Q = \sum_{i=1}^{N} {}^t v_i v_i$$

- Define the edge colored graph $E(C)$ on N vertices with vertex and edge color

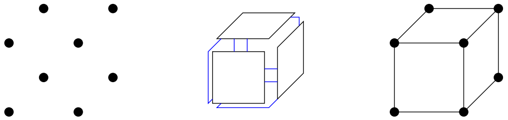
$$c_{ij} = v_i Q^{-1} {}^t v_j$$

- The automorphism group of the edge colored graph is $Lin(C)$ and we use partition backtrack algorithms for computing the group and also testing equivalence.

II. Dual description using symmetries

Dual description problem

- Given a polytope P defined by vertices we want to find its facets



- The problem of going from the facets to the vertices is equivalent to this one by duality (done via cdd/ppl/lrs).
- The dual description problem is useful for many different kind of computations.
- Typically, the polytopes of interest are the one with a large symmetry group. We do not want the full set of facets, just representatives.
- n-dimensional Hypercube has $2n$ facets but 2^n vertices. Combinatorial explosion is to be expected in general.

The recursive adjacency decomposition method

- Basic idea:
  - Compute some initial facet by linear programming
  - For each untreated facet, we compute the adjacent facet (by dual description)
  - We test for equivalence of the facets.
  - Terminate when all have been trated.
- Improvements:
  - We can prematurely terminate using some criterion.
  - We can apply the method recursively.
  - We can use any additional symmetry in the computation.
  - We can parallelize the enumeration.
- Reinvented many times (D. Jaquet 1993, T. Christof and G. Reinelt 1996).
- Used for many different enumerations and also in some infinite group settings.

Dual description of $W(H_4)$

- The Weyl group $W(H_4)$ defines a set of 14400 vectors in $\mathbb{R}^{16}$.
- The symmetry group has size $2 \times (14400)^2$ (multiplication on the left, right and inverse)
- We take the convex hull of those vectors and this defines a polytope.
- According to a conjecture of
  - N. McCarthy, D. Ogilvie, I. Spitkovsky, and N. Zobin, Birkhoff's theorem and convex hulls of Coxeter groups, Linear Algebra Appl. 347 (2002), 219–231.

  The corresponding polytope has only one orbit of facets.
- The conjecture is actually false and it has 1063 orbits of facets.

III. Other functionalities

Perfect forms and related computations

- We define $S^n_{\geq 0}$ the cone of positive semidefinite matrices.
- For a positive definite symmetric matrix A we define

$$\begin{array}{rcl} min(A) & = & \min_{x\in\mathbb{Z}^n-\{0\}} A[x] \\ Min(A) & = & \{x\in\mathbb{Z}^n \text{ s.t. } A[x]=min(A)\} \end{array}$$

  A if perfect if defined uniquely by $min(A)$ and $Min(A)$.
- For A perfect

$$Dom(A) = \text{conv}\{x^T x \text{ for } x\in Min(A)\}.$$

- Related computations:
  - Enumerating perfect forms.
  - For a configuration of vectors testing if there is a symmetric matrix realizing it.
  - For a given positive form finding B such that $B\in Dom(A)$.
- $S^n_{\geq 0}$ is self-dual.

Copositive programming

- Given a symmetric quadratic form $Q \in S^n$ it is called **copositive** if
$$Q[x] \geq 0 \text{ for } x \in \mathbb{R}^n_+$$
This defines a cone COP_n.
- It is called **completely positive** if
$$Q = \sum_{i=1}^{N} \alpha_i v_i^T v_i$$
for $\alpha_i \geq 0$ and $v_i \in \mathbb{R}^n_+$. This gets a cone $CP_n = COP_n^*$.
- We have
$$COP_n \subset S^n_{\geq 0} \subset CP_n$$
- We coded algorithm for deciding membership questions by using cellular decompositions and linear programming.

Edgewalk algorithm by Allcock

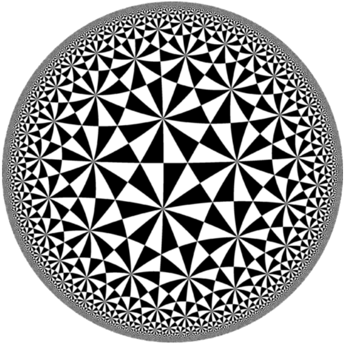
- A classical problem for a lattice of signature $(n, 1)$ for which we want to find the Hyperbolic Coxeter group if it exists.
- Finding the Coxeter group is equivalent to finding the fundamental domain according to Coxeter group.
- The classical algorithm is Vinberg's algorithm that builds the domain facets by facet by using some short vector problem.
- Allcock in his paper
  - Allcock, D.*An Alternative to Vinberg's Algorithm*, ArXiv:2110.03784

  introduced another approach: compute vertices and from the vertices find the adjacent vertices. When all vertices have been treated we have actually the polytope.
- This is somewhat more general and efficient than Vinberg's algorithm.

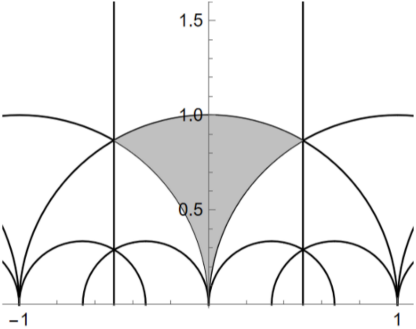
IV. Fundamental domains of some cocompact groups (With Paul Gunnells)

Cocompact groups and fundamental domain

- A group G acting on a space X is cocompact if the quotient X/G is compact. Example $\mathbb{Z}^2$ acting on $\mathbb{R}^2$ has quotient a two dimensional torus.
- A fundamental domain is a polyhedral domain D such that for D tiles X by its orbit and every point in the interior of X has trivial stabilizer.



Cocompact



Not cocompact

Applications of fundamental domains

- We can compute the rational homology of the group.
  - This requires knowing the cell complex up to the group action.
  - Possible since the stabilizers are finite by the cocompactness.
- We can compute the integral homology of the group.
  - This requires computing resolution of the face stabilizers and using the CTC Wall lemma.
- We can also compute the Hecke operators for elements in the group $G(\mathbb{Q})$.
  - This requires being able to move faces to the skeleton of the fundamental domain
  - See Voight J. *Computing fundamental domains for Fuchsian groups* JTNB **21** (2009) 467–489.
- We can compute a presentation of the group.
  - This is done by using the Poincaré Polyhedron Theorem.
- Computing normalizers of the group.
  - We can compute the automorphism group $Aut(D)$ of the fundamental domain D in $SL_n(\mathbb{R})$ and $Aut(D)$ should be a subgroup of $N(G, SL_n(\mathbb{R}))/G$.

Poincaré Polyhedron Theorem

- We have a group G acting on a space X.
- If we have a domain D that tiles X by the action of the group G. Suppose D has N facets $F_1, \ldots, F_N$.
- To each facet F_i corresponds a group element g_i.
- **Theorem**: The elements g_i generate the group and they provide a presentation of the group subject to the following relations:
  - The facet F_i corresponds to a facet F_j in the image $g_i(D)$. The transformation g_j maps it back. Thus we have $g_i g_j = e$.
  - For each codimension 2 face we have a sequence of transformations $g_{i_1} g_{i_2} \ldots g_{i_k} = e$.
- For the Coxeter group this gets us the Coxeter presentation.

Venkov reduction theory

- Suppose that we have a positive definite quadratic form A and the group $\mathrm{GL}_n(\mathbb{Z})$ acts on it.
- We obtain the full orbit $Orb(A)$ under the group.
- The idea is to consider:

$$Venkov(A) = \left\{ \begin{array}{c} B \in S^n \text{ s.t. } Tr(BA) \leq Tr(BP^T AP) \\ \text{for } P \in \mathrm{GL}_n(\mathbb{Z}) \end{array} \right\}$$

  It is a little similar to Voronoi polytope.
- Idea introduced by Venkov but few studies of it:
  - Crisalli, A. J. *The fundamental cone and the Minkowski cone*. J. Reine Angew. Math. 277 (1975), 74–81.
  - Tammela, P. P. *On the theory of the reduction of positive quadratic forms in the sense of Venkov*.
  - Venkov B.A. *On the Reduction of Positive Definite Quadratic forms*.
- It is polyhedral for any n and the cone is known for $n \leq 3$.

Fundamental domain for general discrete linear groups

- Suppose we have a group G acting on a space X. Let $x \in X \subset \mathbb{R}^n$.
- We define

$$X^* = \left\{ y \in \mathbb{R}^n \text{ s.t. } \min_{g \in G} \langle y, x.g \rangle > -\infty \right\}$$

- We define the Venkov domain in the following way:

$$Venkov(x, G) = \{ y \in \mathbb{R}^n \text{ s.t. } \phi_g(y) \geq 0 \text{ for } g \in G \}$$

with

$$\phi_g(y) = \langle y, x.g \rangle - \langle y, x \rangle$$

- The domains $Venkov(x, G)$ defines a tessellation of the space X^*. It is a face to face tiling.
- The action of G on X^* is by the contragredient representation $(g^T)^{-1}$.

Shortest Group Element Problem

- For a group G acting on a set X, $x \in X$, and $a \in X^*$ the SGE is

$$\min_{g \in G} \langle a, xg \rangle$$

  and the elements g realizing it.
- The problem is unsolvable in full generality.
- But a subproblem is actually solvable:
  - We have some constant C.
  - We know there exist a $g \in G$ such that $\langle a, xg \rangle < C$.
  - We want just one such g.
  - We have a generating set S of G.

  The solution is to iterate over all the group elements by using S.
- It turns out that we do not need anything more.

Iterative scheme

- Start from a cocompact group and a point x in the interior of X.
- We want to find the fundamental domain.
- Select a number of elements in the group $g_1, \dots, g_N$. Then iterates the following:
  - Form the polyhedral cone C defined by the inequalities ϕ_{g_i}.
  - By Clarkson method, we can identify which inequalities are redundant and eliminate them from the list.
  - If C is a fundamental domain then we are done.
  - If not, then we can find some new elements to add to the list.
- We hope that finitely many inequalities suffice.
  - Facet inequalities have to match.
  - Facet have to match.
  - Ridges should be coherent.

Matching facet inequalities

- For a fundamental domain D defined by inequalities ϕ_g for $g \in S$ we have that if $g \in S$ then $g^{-1} \in S$.
- The problem is interesting for $n > 2$ because for $n = 2$ taking inverse is linear.
- It frequently happens that in the intermediate step of enumeration we have that ϕ_g defines a non-redundant facet but $\phi_{g^{-1}}$ is redundant.
- This means that ϕ_g is made redundant by other inequalities yet to be discovered:
  - By linear programming, we can find a point y_g interior to the facet inequality defined by ϕ_g
  - By using SGE we can find a $h \in G$ such that

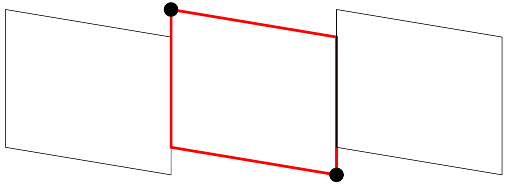
    $$\langle y_g, x.h \rangle < \langle y_g, x \rangle$$

  - We insert h and h^{-1} into the list of inequalities.
- Eventually the processus converges

Face-to-face tilings

It is possible that the inequalities of a domain and the adjacent one are the same up to sign but do not match:

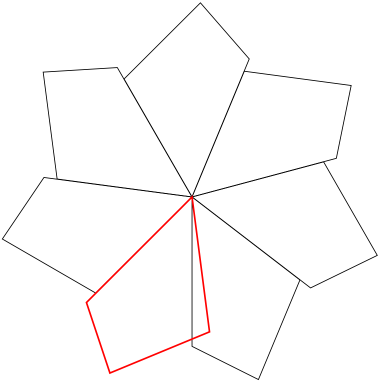
- In that case, we can find vertices of the polytope that should not be present.



- Such vertices can be found either by linear programming or a dual description
- We apply SGE on them and find corresponding elements to insert.
- Iteratively we resolve the problem.

Ridge matching

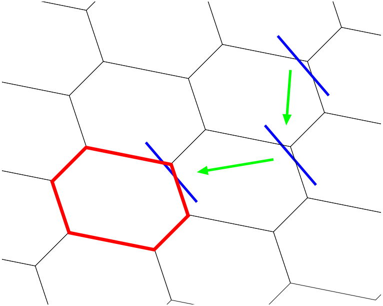
- Around a ridge ($(n-2)$-dimensional cell) we need to have a concordance of the fundamental domains. We want to avoid:



- If a collision happens then it means that an element has been missed.

Inserting elements

- Suppose that we have built a complex, how can we insert new elements?



- We do some step to improve using the facet generators.
- When we cannot improve any more, we used the facet inequalities

There is still room for improvement.

Proving correcteness

- We typically run the process with a set of generators that we send into the system.
- This is the difference with a perfect form based system where the genrators are built as part of the process.
- It is not always the case that we have a generating set.
- One way to address this is to use volume arguments:
  - The volume would be computed from abstract argument.
  - The volume could be computed by numerical integration over the fundamental domain D.
  - If equal then we have concluded (equality needs to be within a factor of two)
- If that fails, then some group element have been missed and thus the iteration should last longer.

V. Example

Witte cocompact subgroup

- Let F be a totally real Galois cubic field with ring of integers R. Let $\sigma\colon F \to F$ generate the Galois group $Gal(F/\mathbb{Q})$.
- Let $p \in \mathbb{Z}$ be such that the central simple algebra $A/\mathbb{Q}$ constructed from the data $[F, \sigma, p]$ is a division algebra.
- Given $(x, y, z) \in R^3$ we define a matrix $\phi(x, y, z) \in M_3(\mathbb{R})$ by

$$\phi(x,y,z) = \begin{pmatrix} x & y & z \\ p\sigma(z) & \sigma(x) & \sigma(y) \\ p\sigma^2(y) & p\sigma^2(z) & \sigma^2(x) \end{pmatrix}.$$

  Then G is the group of all $\{\phi(x, y, z) \mid x, y, z \in R\}$ of determinant 1.
- G is cocompact.

Results

- Our example is $F = \mathbb{Q}(\alpha)$ with $\alpha = 2\cos(2\pi/7)$ of satisfying $\alpha^3 + \alpha^2 - 2\alpha - 1 = 0$ and $p = 2$.
- We consider the ring $R = \mathbb{Z}[\alpha]$ of discriminant 49 and $p = 2$.
- We make it act on the cone of positive definite matrices and obtain a 6 dimensional representation and take $x = Id_3$.
- One advantage is that the stabilizer is trivial. Not a given. All the theory can be done for finite stabilizers.
- Already considered in
  - Braun, Coulagean, Nebe, Schönnenbeck, *Computing in arithmetic groups with Voronoi's algorithm*, Journal of Algebra **435** (2015) 263–285
- We make the group act on $S^3_{\geq 0}$ and we set $x = Id_3$.
- We take the full orbit and incrementally add elements. Doing the facet matching we get 706 elements.
  We need to to the ridge matching.