Practical computations with indefinite forms

Mathieu Dutour Sikiric

Institute Rudjer Boskovic

Definition and problems considered

We work with integral quadratic forms A defined on $\mathbb{Z}^n$ with signature (p, q). We consider following questions:

1. Compute a generating set of the group of inversible integral transformations preserving A.
2. Given two forms A_1 and A_2 test if there is an inversible integral transformation ϕ such that $A_2[x] = A_1[\phi(x)]$.
3. Given $C \neq 0$ find the orbit representatives of solutions of $A[x] = C$.
4. Find the orbit representatives of solutions of $A[x] = 0$ with x primitive.
5. For $k \geq 2$ find the orbit representatives of k totally isotropic spaces.

The example that guide us is $U + U(2) + \mathrm{E}_8(-2)$ for *Moduli of polarized Enriques surfaces – computational aspects*, Mathieu Dutour Sikirić and Klaus Hulek, in preparation.

General plan of the method

1. For positive definite and hyperbolic forms there are well known techniques.
2. The integral group algorithms that are used everywhere.
3. The general indefinite case $p, q \geq 2$.
4. The indefinite LLL speedup.
5. Isotropic vectors and k-planes.
6. Bonus: The equivariant edgewalk algorithm

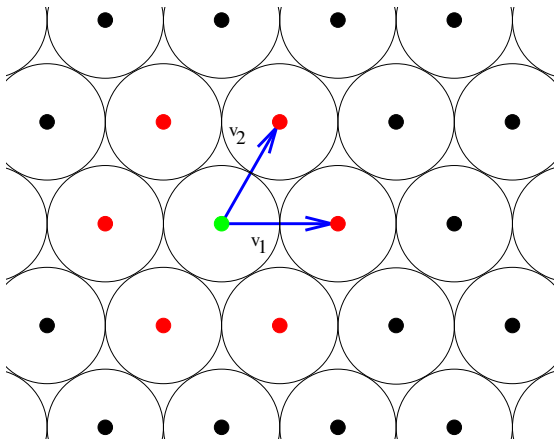
All the techniques here are relatively elementary. The code is implemented in `GAP` and `C++`.

I. Positive definite and hyperbolic cases

Solution for positive definite (and positive negative) forms

- For positive definite forms, the Plesken-Souvignier algorithm allows to test isomorphism and computing automorphism group of positive forms.
- The method uses short vectors, that is computes the following set of vectors for some λ:

$$Min_\lambda(A) = \left\{ \begin{array}{c} x \in \mathbb{Z}^n \text{ s.t.} \\ A[x] \leq \lambda \end{array} \right\}$$



Example: For A_2 lattice:

$$\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

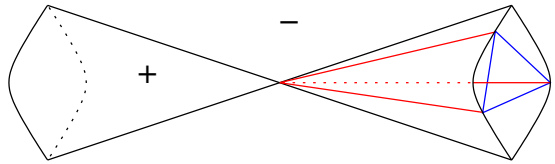
- Once we have the automorphism group we can compute the orbit representatives of solutions of $A[x] = C$.

Hyperbolic cones

- We have a quadratic form A of signature $(1, n-1)$.
- We define the positive set

$$C = \{x \in \mathbb{R}^n - \{0\} \text{ s.t. } A[x] \geq 0\}.$$

It splits into two connected components C_1 and C_2. We select one and call it C_+.



- What we work with is the integral part of this set: $C_+ \cap \mathbb{Z}^n$.
- The perfect forms correspond to the facets of $\text{conv}(C_+ \cap \mathbb{Z}^n)$.
- The perfect forms have a finite number of vertices and a finite stabilizer group.
- There is a finite number of perfect forms up to equivalence.

Voronoi algorithm

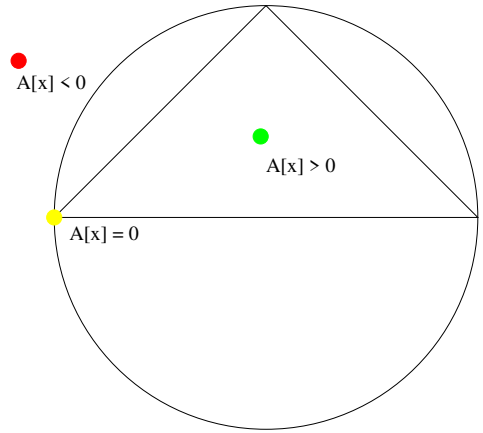
- Take again A a quadratic form of signature $(1, n-1)$.
- Suppose we have a perfect form P from a known list $\mathcal{P}$ with vertices $\{v_1, \ldots, v_m\}$ with a defining linear functional l_P. Then we can compute the facets of P. From each facet F we can
  - Compute the two-dimensional space of linear forms that contains F in their kernel.
  - Find the linear form that correspond to the perfect form P' adjacent to P on F (lifting procedure).
  - Check if the obtained perfect form P' is equivalent to one we already have. If not, insert in into $\mathcal{P}$.
- An initial perfect form can be computed by applying the lifting procedure, starting from just one vector and getting upward in dimension.
- This is a direct analog of the Voronoi algorithm for perfect forms in Euclidean space.

Solution of the automorphism group/equivalence problems

- If two forms are equivalent, then their cones $C_+ \cap \mathbb{Z}^n$ are also equivalent and so the perfect forms are matching.
- Therefore, given two forms A_1 and A_2 it suffices to compute one perfect form C^1 for A_1 and enumerate the perfect forms C_i^2 of A_2. Then test whether one of the C_i^2 is isomorphic to C^1.
- A generating set of the automorphism group $Aut(L)$ is obtained by taking:
  1. Generating sets of the stabilizer of each perfect forms C_i of L.
  2. The equivalences obtained when applying Voronoi algorithm.
- Michael H. Mertens, *Automorphism groups of hyperbolic lattices*, Journal of Algebra 408 (2014) 147–165

Enumerating vector representatives

- From the perfect cones we can get the vectors of zero (isotropic vectors are vertices) and positive norm (by copositive programming enumeration):



- By using the polyhedral structure given by the perfect forms, identify the orbits by bookkeeping.
- An alternative strategy for testing isomorphism of two vectors x and y is to compute their orthogonal lattice $x^\perp$ and $y^\perp$ check existence of an isomorphism ϕ and if so:
  1. If x, y are isotropic then extend ϕ from $x^\perp$ to $y^\perp$ and check for integrality.
  2. If x, y are of positive norm, extend ϕ from $\mathbb{Z}x + x^\perp$ and $\mathbb{Z}y + y^\perp$ and int. check.
- For negative norms, no known strategies.

II. Integral group algorithms

Linear symmetry groups of a vector configuration

- We work with full dimensional vector configuration $(v_i)_{1 \leq i \leq N}$ in $\mathbb{R}^n$.
- The linear symmetry group $Lin((v_i))$ is the group of transformations $\sigma \in \mathrm{Sym}(N)$ such that there exist $A \in \mathrm{GL}_n(\mathbb{R})$ with $Av_i = v_{\sigma(i)}$.
- Define the form

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

- Define the edge colored graph $E((v_i))$ 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 $Lin((v_i))$.

Integral groups problems

We will need to resolve many times following kind of problems for a group $G \subset \mathrm{GL}_n(\mathbb{Q})$:

1. Compute the intersection $G \cap \mathrm{GL}_n(\mathbb{Z})$.
2. For $h \in \mathrm{GL}_n(\mathbb{Q})$ find $g \in G$ such that $gh \in \mathrm{GL}_n(\mathbb{Z})$ if it exists.
3. Compute the right cosets of $G \cap \mathrm{GL}_n(\mathbb{Z})$ in G.

Notes:

- For a general group G there is no reason to expect an algorithm to exist.
- What we require for it to work is that there exist a lattice $L \subset \mathbb{Z}^n$ which is preserved by G.
- This condition is satisfied by finite groups and all groups of this study.

Computing $G \cap \mathrm{GL}_n(\mathbb{Z})$

- Let L be an integral lattice left invariant by G.
- We express G within L and it becomes an integral linear group.
- The lattice $L' = \mathbb{Z}^n$ satisfies $L \subset L'$.
- Define d the smallest integer such that $L \subset L' \subset \frac{1}{d}L$.
- We have a natural mapping of G into $\mathrm{GL}_n(\mathbb{Z}_d)$ and L' corresponds to a subset of $(\mathbb{Z}_d)^n$.
- We can apply the partition backtracking technique and find the stabilizer of L'. Remarks:
  - If the group is infinite we have to use Schreier's Lemma which leads to huge generating sets, for finite groups, this is simpler.
  - We can go prime by prime in the prime factorization of d.
  - We can go orbit of vector by orbit of vector in L'.
  - We can also use subspace filtration if known.
  - Sometimes those techniques are insufficient.
- An alternative approach is to compute the stabilizer directly by computing the orbit of L' under G.

III. Full indefinite case $p, q \geq 2$

Exceptional isomorphisms

- Define the Lie group $SO(p,q)$ to be the group of real automorphisms of a quadratic form of signature (p,q) of determinant 1. Then we have the exceptional isomorphisms:

$SO(2,2) = SL(2,\mathbb{R})^2$	$SO(2,1) = SL(2,\mathbb{R})$
$SO(3,1) = SL(2,\mathbb{C})$	$SO(4,1) = Sp(1,1)$
$SO(5,1) = SU^*(4)$	$SO(3,2) = Sp(2,\mathbb{R})$
$SO(3,3) = SL(4,\mathbb{R})$	$SO(4,2) = SU(2,2)$

- What we use for later is $SO(2,2) = SL(2,\mathbb{R}) \times \mathrm{SL}(2,\mathbb{R})$.
- Define $M_2(\mathbb{R})$ the set of 2×2-matrices. We have

$$det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc = (abcd) \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} /2$$

- So, $M_2(\mathbb{R})$ equipped with det is of signature $(2,2)$ and multiplication on the left right by matrices of determinant 1 preserve it which gives us the identification.

Eichler's criterion

- Defines U the quadratic form $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ which corresponds to an hyperbolic plane.
- Eichler's criterion applies to lattices of the form $L = U + U + W$ with W an even lattice (integral and even on the diagonal). It gives:
  - A set of generators named Eicher's transvection that defines a subgroup K of $O(L)$.
  - For each $h \neq 0$ a finite list of orbit representatives and for $h = 0$ of primitive orbit representatives for the group K.
- As far as I know there is no other general results. It would be nice to use the exceptional isomorphism $SO(2,1) = SL(2,\mathbb{R})$ to solve the hyperbolic case faster.

Approximate models

- For a lattice L an approximate model is an algorithm that do the following:
  - Provide a finite set of generators of $Aut(L)$ that generates an approximate symmetry group $Ap(L)$
  - Provide an oracle function $Ap(L, h)$ that given $h \neq 0$ returns a finite list $v_1, \ldots, v_{k(h)}$ such that any vector of norm β is equivalent by $Ap(L)$ to one of the v_i. For $h = 0$ the oracle function returns a list of primitive vectors of norm 0 such that any primitive vector of norm 0 is equivalent to one such vector by an element of $Ap(L)$. The v_i are named the approximate orbit representatives.
- Example: Eichler's criterion provides an approximate model for $U + U + W$.
- If we have a finite index subgroup then it would provide an approximate symmetry group. But we need an algorithm which is typically not known.

Approximate model for a sublattice

Theorem: Suppose L' and L are two lattices of rank n with $L' \subset L$ and we have an approximate model for L. Then we have an approximate model for L'.

- We can compute the stabilizer S of L' under $Ap(L)$ by the integral group algorithms.
- We compute the right coset decomposition of $Ap(L)$ under S with coset representatives $g_1, \ldots, g_m$.
- For $h \in \mathbb{R}$, the approximate model of L gives us representatives $x_1, \ldots, x_t$ of the orbits of vectors of norm h.
- We then considers all the elements of the form $g_j x_i$ and keep the ones that are contained in L'. This gets us our approximate orbit representatives $Ap(L', h)$

Cor: $U(c) + U(d) + W$ has an approximate model since we have an embedding $L \subset U + U + W$ by the formula

$$(x_1, x_2, y_1, y_2, w) \mapsto (cx_1, x_2, dy_1, y_2, w)$$

Finding approximate models

Take a lattice L of dimension greater than 7.

- Take the dual L^*.
- It has at least two isotropic vectors v_1, v_2 (we can use Denis Simon program to find them) which we can assume not orthogonal.
- Apply the same argument to $L^* \cap (\mathbb{Z}v_1 + \mathbb{Z}v_2)^\perp$ and get v_3, v_4.
- Define $K = L^* \cap (\mathbb{Z}v_1 + \mathbb{Z}v_2 + \mathbb{Z}v_3 + \mathbb{Z}v_4)^\perp$, take the dual and obtain

$$L \subset U(c) + U(d) + K^*$$

- We find a factor $\alpha > 0$ such that $K^*(\alpha)$ is even, αc and αd are integers.
- We then have the embedding

$$U(\alpha c) + U(\alpha d) + K^*(\alpha) \subset U + U + K^*(\alpha)$$

and we have an approximate model

This is not an optimal construction.

The recursion on $s(L) = \min(p, q)$

- Suppose that L is a lattice of signature (p, q) with $p \leq q$.
- Select $h > 0$ such that there exist vectors of norm h. Compute the approximate list of orbit representatives $x_1, \ldots, x_m$.
- For such a vector x, computes the orthogonal lattice $x^\perp$ for which $s(x^\perp) = s(L) - 1$.
- Compute the automorphism group of $x^\perp$, computes its embedding in $Aut(\mathbb{Z}x + x^\perp)$ and then use the integral group algorithm.
- For each pairs of orbit representatives, x_i and x_j test equivalence of $x_i^\perp$ and $x_j^\perp$ and whether this extends to an automorphism of L.
- All together, this gets us a generating set of the automorphism group of L.

Similarly we can solve the equivalence problem and the problem of resolving $A[x] = h$ for $h \neq 0$. For isotropic questions, more is needed.

IV. Indefinite LLL speedup

Memoization technique

- The recursive nature of the algorithm makes its practical usage expensive.
- Most of our computations had been done for lattices with $s(L) = 2$ and only once for a lattice with $s(L) = 3$.
- Memoization techniques rely on computing something and keeping the result of the computation in memory so that it can be retrieved when needed.
- For example for computing automorphism group, we could keep the result of an automorphism group computation.
- So, if we have $\mathrm{Aut}(L)$ and kept it and want to compute again $\mathrm{Aut}(L)$ we could use the stored value.
- But what about computing $\mathrm{Aut}(L')$ for a new lattice L' which could be equivalent to L?
- Testing exactly for isomorphism is about as hard as computing $\mathrm{Aut}(L')$.

The indefinite LLL algorithm

- The LLL algorithm is a tremendous tool for simplifying existing geometrical problem.
- The idea of indefinite LLL (by Denis Simon) is simply to replace $A[x]$ by $|A[x]|$.
- If we find an isotropic vector in the computation then we randomly shuffle the vectors and iterate until we do not get any further reduction in L_1 norm of A.
- The simplified matrix is then canonicalized by action of the group $Sym(n) \times \mathbb{Z}_2^n$ on its coefficients.
- This gets us an approximate canonical form.
- If the approximate canonical form matches then we can use the result of previous computations.
  If not, then well we recompute, no harm.

V. Isotropic vectors (and k-spaces)

The fundamental lemma

Lemma: If $\mathbf{v}$ is an isotropic vector in L then any automorphism of $\mathbf{v}^{\perp}$ extend to a rational automorphism of L.

- Define $H = \mathbf{v}^{\perp}$ and g an isometry of H.
- We want to extend it to an isometry of L.
- We select a vector u not in H which gets us

$$g(u).g(w) = u.w \quad \text{for} \quad w \in H$$

- Thus $g(u) = u' + C\mathbf{v}$ for some $C \in \mathbb{R}$ with $u' \notin H$.
- $g(u).g(u) = u.u$ is a linear equation with a unique solution because $\mathbf{v}$ is isotropic and $u'.\mathbf{v} \neq 0$.
- So extension is unique but may not be integral.

Thus isotropic vectors are handled in the same way as non-isotropic ones with the integral group algorithms.
Example: For $U + U(2) + \mathrm{E}_8(-2)$ we found 2 orbits of lines.

Isotropic k-spaces with $k > 1$

We have $e_1, \ldots, e_n$ a basis, $Is = \mathbb{Z}e_{k+1} + \cdots + \mathbb{Z}e_{2k}$ an isotropic k-space and $Is^{\perp} = \mathbb{Z}e_{k+1} + \cdots + \mathbb{Z}e_n$. The lattice B is written below with J diagonal, A non-degenerate and the automorphism P stabilizing B as:

$$B = \begin{pmatrix} 0 & J & 0 \\ J & 0 & 0 \\ 0 & 0 & A \end{pmatrix}, P = \begin{pmatrix} P_1 & P_2 & P_3 \\ 0 & P_4 & 0 \\ 0 & P_5 & P_6 \end{pmatrix}$$

- Entries P_1 and P_3 are uniquely determined by P_4, P_5 and P_6.
- The equation for P_2 is of the form $P_2 J + J P_2^T = W$.
- if $k = 1$ then we have an unique solution.
- If $k > 1$ then for given (P_4, P_5, P_6) we have a continuous of possible solutions.
- By selecting an adequate d the additional condition $P_2 \in M_k(\mathbb{Z})/d$ still defines a group.
- So, we can apply the integral group algorithms.

Example: For $U + U(2) + \mathrm{E}_8(-2)$ we found 2 orbits of planes.

VI. Equivariant edgewalk algorithm (by Allcock)

The problem of hyperbolic lattices

- The most time intensive part of the algorithms deployed are:
  1. The integer group algorithms.
  2. The computation with hyperbolic lattices.
- One possible alternative would be to use Vinberg algorithm, but it does not work in general:
  1. For reflective lattices it works but they are very few such lattices.
  2. For non-reflective lattices, the fundamental domain has infinitely many facets with an infinite group acting on it.
  3. But: There are lattices without roots.
  4. But: It is possible that there is no vertices.
  5. But: The set of vertices with its adjacency relation may not be connected
- Our prior is thus that one cannot use the structure of roots for testing hyperbolic lattice isomorphism in general but it may be useful in some cases.

The equivariant edgewalk algorithm

- The algorithm takes an integral hyperbolic form and returns:
  1. A connected component C of the set of vertices of the fundamental domain $\mathcal{F}$ of L.
  2. A list of orbits of vertices of C.
  3. A list of orbits of facets of $\mathcal{F}$ containing C in their incident vertices.
  4. A list of generating set of $Stab(C, \mathrm{Aut}(L))$.
- Daniel Allcock, *An Alternative to Vinberg's Algorithm*, math/arXiv:2110.03784
- `https://github.com/MathieuDutSik/polyhedral_common`
- If there is just one connected component of vertices then we get a set of generating elements of $\mathrm{Aut}(L)/\mathrm{Cox}(L)$.
- This could help getting the automorphism group of K3 surfaces.

THANK YOU