The Vinberg algorithm for Lorentzian lattices: Algorithmic aspects

Mathieu Dutour Sikiric

Institute Rudjer Boskovic

I. Coxeter groups of Lorentzian lattices

Lorentzian lattices and their roots

- A Lorentzian lattice is a lattice $\mathbb{Z}^n$ with an integer quadratic form G of signature $(n-1, 1)$.
- Note: The convention in algebraic geometry is to take signature $(1, n-1)$.
- A root of a Lorentzian lattice is a vector $v \in \mathbb{Z}^n$ with $G[v] = k$ such that the reflection along this root defines an unimodular integral transformation.
- In term of the quadratic form this is equivalent to

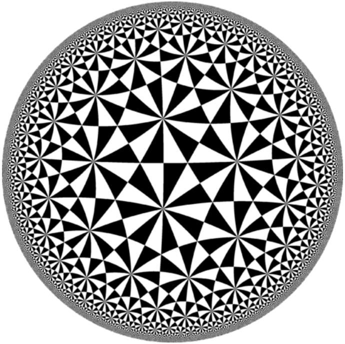
$$G[v] = k \text{ and } 2Gv/k \in \mathbb{Z}^n$$

- There are Lorentzian lattices without roots (by Gael Collinet):

$$\begin{pmatrix} 0 & 0 & 49 \\ 0 & 49 & 7 \\ 49 & 7 & 3 \end{pmatrix}$$

Hyperbolic Coxeter groups

- The hyperbolic Coxeter group $Cox(L)$ of a Lorentzian lattice L is the group generated by hyperbolic reflections of L.
- Define H^{n-1} the hyperbolic space formed by one component of $\{x \text{ s.t. } q(x) < 0\}$.
- $Cox(L)$ has a fundamental domain $Fund(L)$ in H^{n-1}.
- Classical example of the $(2, 3, 7)$ triangle group (though not a Lorentzian lattice):



Reflectivity and relation to K3 surfaces

- For a Lorentzian lattice L, $Cox(L)$ is a normal subgroup of the group of isometries $Isom(L)$ of L.
- A Lorentzian lattice is reflective if $Cox(L)$ is a finite index subgroup of $Isom(L)$.
- For K3 surfaces, the Picard group has a structure of a Lorentzian lattice and the automorphism group of the surface is isomorphic to the quotient $Isom(L)/Cox(L)$.
- The group $Isom(L)/Cox(L)$ is represented as a group of isometries preserving $Fund(L)$.
- A Lorentzian lattice is reflective if and only if $Fund(L)$ has finite covolume.

Fundamental domain

- A fundamental domain D is determined by a number of roots $(r_1, \dots, r_N)$ with N possibly infinite.
- The Coxeter matrix of scalar product is $(a_{ij})_{1 \leq i,j \leq N}$ with $a_{ij} = r_i^T G r_j$.
- We have $r_i^T G r_j \leq 0$.
- The fundamental domain is defined by $r_i^T G x \leq 0$. The vertices of the fundamental domain allow to determine many properties:
  - Whether the fundamental domain determines a cocompact hyperbolic group. This corresponds to all extreme rays $e = \mathbb{R}_+ v$ having $G[v] < 0$.
  - Whether the fundamental domain determines a finite covolume hyperbolic group. This corresponds to all extreme rays $e = \mathbb{R}_+ v$ having $G[v] \leq 0$.

Subdiagrams of a hyperbolic Coxeter diagram

- A subdiagram is a collection of vertices of the diagram that defines a face of the fundamental domain.
- The vertices that have $G[e] < 0$ (resp. $G[e] \leq 0$) correspond to spherical (resp. Euclidean) subdiagrams of the diagram.
- This implies that interior vertices have all the same incidence to the facets.
- The software `CoxIter` can determine all subdiagrams of a given Coxeter matrix and decide several properties like finite covolume of cocompact accordingly.

II. The Vinberg algorithm

Possible root lengths

- For a Lorentzian lattice of Gram matrix G.
- Define the adjoint matrix $coadj(G)$ and the greatest common divisor of the coefficient.
- Define $E(G)$ to be

$$E(G) = \frac{|det(G)|}{gcd(coadj(G))}$$

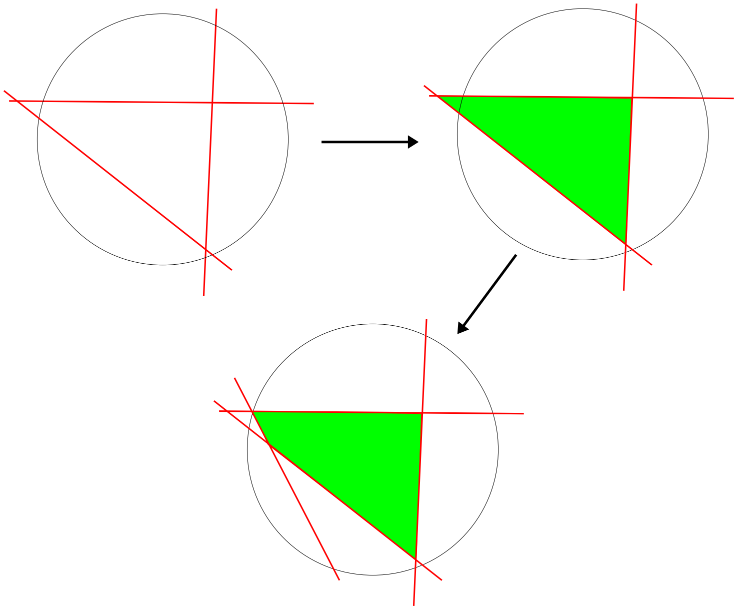
- The possible root lengths must divide $2E(G)$.
- This is a necessary, but not a sufficient condition.
- For example for $U + 2E_8 + \langle 2 \rangle$ this gives 1, 2 and 4. 1 can be excluded by evenness. It turns out that 4 does not show up when the computation is finished.
- Outcome: We can easily compute the set of possible root lengths.

Vinberg algorithm

- The algorithm allows to find a fundamental domain of an hyperbolic Lorentzian lattice.
- It requires the choice of a vector v_0 of negative norm. We define $H = v_0^{\perp}$ the orthogonal space to the vector v_0. It is positive definite for the scalar product induced by G.
- We first look at the roots in the space H and determine a connected component of the hyperplane arrangement.
- The lattice $\mathbb{Z}^n$ is an union of translates of H: $\mathbb{Z}^n = \cup_{i\in\mathbb{Z}}(iw + H)$ for some vector w.
- The idea is to iterate over i and to find roots over the space $iw + H$.

It is not really an algorithm, since if the lattice is not reflective, then the number of facets is infinite and so it never terminates.

Schematic of the algorithm



Fincke-Pohst algorithm

- It is an algorithm that allows to determine the integer points of an ellipsoid.
- It works with backtracking, so do not use memory. The principle is to write the quadratic form as

$$q(x) = a_{11}(x_1 + \sum_{j>1} b_{j1}x_j)^2 + a_{22}(x_2 + \sum_{j>2} b_{j2}x_j)^2 + \cdots + a_{nn}x_n^2$$

  with $a_{ii} > 0$
- For resolving the equation $q(x) = k$ what we have is $a_{nn}x_n^2 \leq k$ which give us a set of possibilities for x_n.
- For each such possibility we consider it and are led to $a_{n-1,n-1}(x_{n-1} + b_{n-1,n}x_n)^2 \leq k - a_{nn}x_n^2$ and so a number of possibilities for x_{n-1}.
- For q positive definite, this allows to solve $q(x) = k$ but also $q(x - c) = k$.

Testing finite covolume of a domain

- Vinberg gave a characterization of the finite covolume fundamental domains.
- The formulation depends on the enumeration of rank $n-1$ and $n-2$. There is also an adjacency condition to check.
- The problem is that enumerating the subdiagram is done by exhaustive enumeration of the subdiagrams.
- In terms of polytope geometry, this is actually equivalent to enumerating all the cells of the polytope, not just the ones of maximal rank.
- This is typically a bad idea since in terms of polytope geometry we have for the n-dimensional simplex a number of cells of the form $\binom{n}{k}$. So, exponential in the middle dimension but linear at the extremes.
- We can avoid storing the full list of subdiagrams and instead pass over it by a tree search (named "Orderly enumeration").

III. Improving the Vinberg algorithm

Reducing the root lattice H

- The condition on roots is $2Gv/k \in \mathbb{Z}^n$.
- Thus it is suboptimal to enumerate the solutions of $G[v] = k$ for $v \in iw + H$ and then filter out by the condition $2Gv/k \in \mathbb{Z}^n$.
- A better idea is to write the condition as $(v, w) \in \mathbb{Z}^{2n}$ with the condition $2Gv = kw$. We find the nullspace and this allows to find a smaller sublattice.
- For $k = 1$ or $k = 2$ this does not give us an improvement.
- The slowest case are the case $k = 1$ and 2.

Improving the Fincke-Pohst algorithm

- If we have the known roots $(r_1, \ldots, r_N)$ we have the inequalities $r_i G r \leq 0$ for an additional root r. This define a polyhedral cone.
- We can use those inequalities to improve the enumeration of the point in the ellipsoid.
- If the polytope is defined by equations $f_k(x) \leq b_k$ and we have fixed say $x_{j+1}, \ldots x_n$ then we are led to a simplified system
- $g_k(x_1, \ldots, x_j) \leq b_k$ we can maximize x_j or minimize it by linear programming and this gets us better bounds for the Fincke-Pohst method.
- But we have to face the problem that doing linear programming at each step is an expensive operation to do. Possible ways to improve this by heuristics.

Improving finite covolume test

- The problem of the characterization by subdiagrams is that we are forced to enumerate all the subdiagrams of any rank of the fundamental domain.
- So, instead, a better approach is to enumerate all the vertices of the polytope from the facets.
- This is a dual-description problem. Still a subject of research, but much less hard than enumerating all the faces.
- If we have a vertex of positive norm, then we know it is not of finite covolume and we can terminate.
- This can be integrated to dual description enumeration codes, so as to stop the enumeration once a vertex of positive norm is found.

Premature termination of Vinberg enumeration

- If a lattice is not reflective, then the enumeration of roots will go on without end.
- Vinberg found a way to terminate it by finding an infinite order automorphism.
- Such automorphism can be found by having pairs of adjacent interior vertices (v, v').
- For pair of adjacent vertices, we find the list of facets which are normal to either of them. They form a space of dimension n. We find the transformations that maps pairs of vertices in the cone.
- We have to see which ones are of infinite order.

Full implementation

- The code is written in **C++** and combines many different software capabilities.
- The code is open source and I contribute daily to it.
- The docker code allows to install the code directly without the need for compilation.
- It is based on code by Alexander Perepechko and Nikolay Bogachev.

The code is available on

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

PS: It is not a Vinberg specific code, it also has functionality for Dual description, canonical form of lattice/polytope, automorphism group of polytope, perfect forms, Delaunay polytope, copositive programming, shortest vector configuration, sparse solver, etc.

IV. The number ring case

The number ring case

- We want to consider quadratic forms of signature $(n-1,1)$ with something like $q(x) = x_1^2 + x_2^2 - \sqrt{2}x_3^2$
- Formally, the settings is the following:
  - We have a Galois group G acting on a ring R
  - We have a quadratic form q such that q is of signature $(n-1,1)$ and for all $\sigma \in G - \{e\}$ the form q^σ is of signature $(n,0)$.
- We still have the inequalities $rGr' \leq 0$

The Fincke-Pohst algorithm

- We have a set of equations $q(x) = k$ and $q^\sigma(x^\sigma) = k^\sigma$.
- So, we write $x = (x_1, \ldots, x_n)$ and each x_i is written as $x_i = \sum \alpha_{i,j} u_j$ with $\{u_1, \ldots, u_d\}$ a $\mathbb{Z}$-basis of R over $\mathbb{Z}$.
- The formulation becomes a Fincke-Pohst like algorithm with inequalities of the form $a_{nn} x_n^2 \leq k$ and $a_{nn}^\sigma (x_n^\sigma)^2 \leq k^\sigma$.
- This means that we have to replace the intervals by a convex set of points.
- The code is implemented by Rémi Bottinelli and available at `https://github.com/bottine/VinbergsAlgorithmNF/`

V. The edge-walking algorithm (by Allcock)

Limitations of the Vinberg algorithm

- When running the Vinberg algorithm we face the problem of having to solve many different batches
- In dimension 2 the root equation to solve is $x^2 - ay^2 = k$ and Vinberg algorithm is to simply iterate from $x = 1$ to the one that we want. There are better solution method in Number theory as this is known as General Pell's equation.
- Note that for $x^2 - 61y^2 = 1$ the smallest solution is $(1766319049, 226153980)$ so iterating over the batches is going to be quite inefficient. For Pell's equation, we have the continuous fraction algorithm by Lagrange.
- The Fincke-Pohst algorithm is intrinsically slow. There are some theoretical reasons to think it cannot be improved.
- The weakness of the Vinberg algorithm is that it does not use the polyhedral structure of the fundamental domain.

The edge walking algorithm

- We first need to find one vertex of the Fundamental domain.
- From each vertex, we can find the direction in which we can find other vertices.
- Allcock has an algorithm for finding the adjacent vertex.
- So, by a graph traversal algorithm, we can iterate until all the vertices have been treated.
- It still has the same problem as Vinberg's algorithm. In the non-reflective case, it still runs forever.
- This algorithm seems limited to the $\mathbb{Z}$ case.

Not yet implemented.

The edge walking algorithm, next generation

Another major weakness of the Vinberg algorithm is that it cannot use the symmetries because the vector v_0 is arbitrary.

- We can keep track of the pairs of adjacent vertices.
- When we find a new pair, we can check for equivalence with the list of known pairs.
- If equivalent, then we have a generator of $Isom(L)/Cox(L)$ and if not a new vertex.
- When the program terminates, we get as output
  - A generating set of $Isom(L)/Cox(L)$
  - List of orbit representatives of vertices of $Fund(L)$
  - List of orbit representatives of facets of $Fund(L)$

Science fiction?