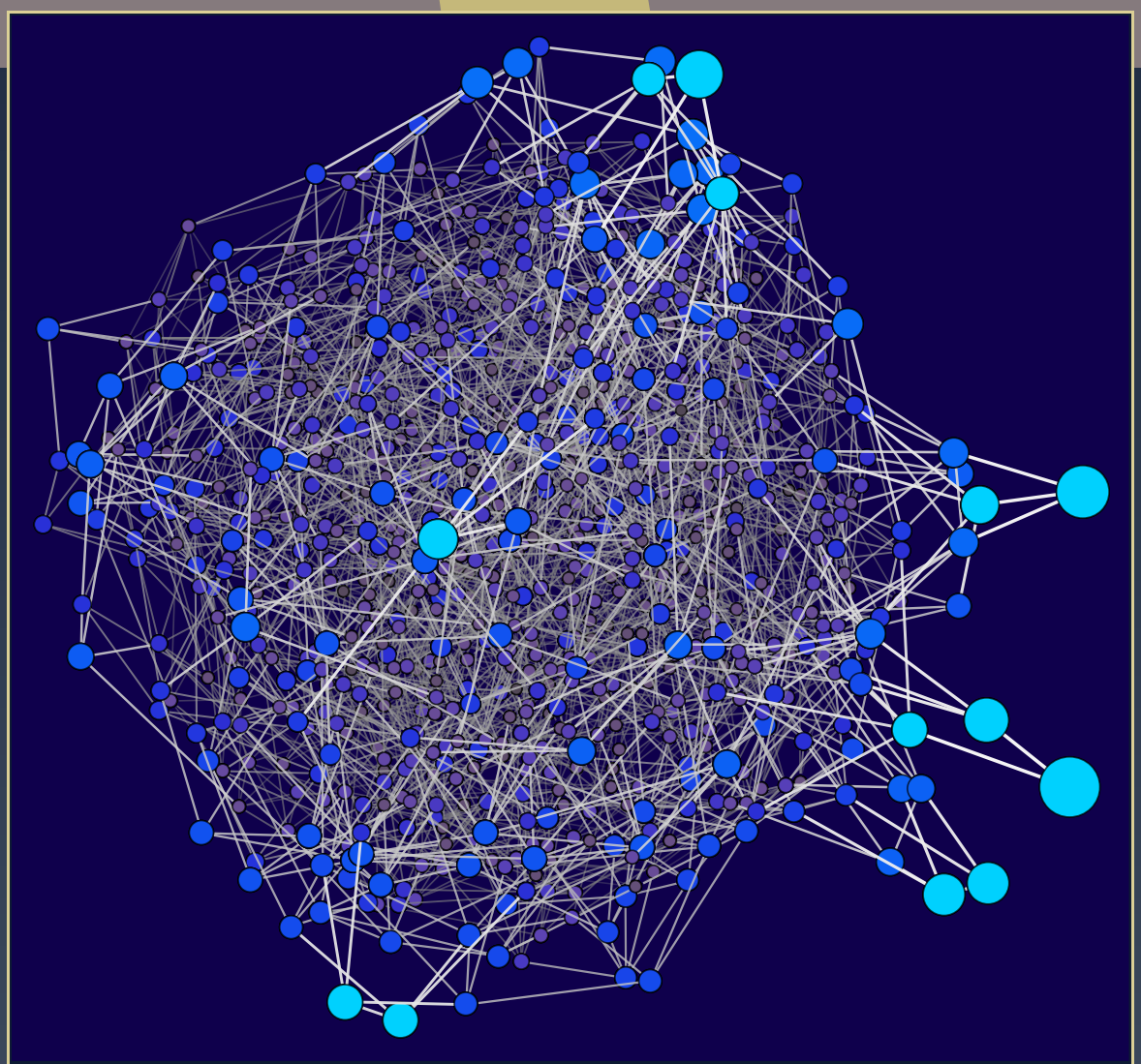
THE OPEN BOOK SERIES 4

ANTS XIV
Proceedings of the Fourteenth Algorithmic Number Theory Symposium

A canonical form for positive definite matrices

Mathieu Dutour Sikirić, Anna Haensch, John Voight, and Wessel P.J. van Woerden



msp

THE OPEN BOOK SERIES **4** (2020)
Fourteenth Algorithmic Number Theory Symposium
https://doi.org/10.2140/obs.2020.4.179

A canonical form for positive definite matrices

Mathieu Dutour Sikirić, Anna Haensch, John Voight, and Wessel P.J. van Woerden

We exhibit an explicit, deterministic algorithm for finding a canonical form for a positive definite matrix under unimodular integral transformations. We use characteristic sets of short vectors and partition-backtracking graph software. The algorithm runs in a number of arithmetic operations that is exponential in the dimension n, but it is practical and more efficient than canonical forms based on Minkowski reduction.

1. Introduction

1.1. ***Motivation.*** For n a positive integer, let $\mathcal{S}^n$ denote the $\mathbb{R}$-vector space of symmetric real $n \times n$-matrices and $\mathcal{S}^n_{>0} \subset \mathcal{S}^n$ denote the cone of positive definite symmetric $n \times n$-matrices. For $A \in \mathcal{S}^n_{>0}$, the map $x \mapsto x^{\mathsf{T}}Ax$ (where $^{\mathsf{T}}$ denotes transpose) defines a positive definite quadratic form, with A its Gram matrix in the standard basis; for brevity, we refer to $A \in \mathcal{S}^n_{>0}$ as a *form*. The group $\mathrm{GL}_n(\mathbb{Z})$ of unimodular matrices acts on $\mathcal{S}^n_{>0}$ by the action $(U, A) \mapsto U^{\mathsf{T}}AU$; the stabilizer of a form A under this action is the finite group

$$\mathrm{Stab}(A) := \{U \in \mathrm{GL}_n(\mathbb{Z}) : U^{\mathsf{T}}AU = A\}. \tag{1.1.1}$$

Two forms $A, B \in \mathcal{S}^n_{>0}$ are said to be (*arithmetically*) *equivalent* if there exists a unimodular matrix $U \in \mathrm{GL}_n(\mathbb{Z})$ such that

$$A = U^{\mathsf{T}}BU. \tag{1.1.2}$$

In the geometry of numbers [39], forms arise naturally as Gram matrices of Euclidean lattices under a choice of basis; in this context, two forms are arithmetically equivalent if and only if they correspond to isometric lattices.

Plesken and Souvignier [35] exhibited algorithms to compute stabilizers and test for arithmetic equivalence among forms, and these have been used widely in practice [2; 8; 10; 21; 37]. In a more theoretical direction, Haviv and Regev [13] proposed algorithms based on the shortest vector problem and an isolation lemma for these purposes as well, with a time complexity of $n^{O(n)}$.

MSC2010: 11H55, 11H56, 15A21.
Keywords: canonical form, quadratic form, positive definite matrix, lattice isomorphism, graph isomorphism.

While these algorithms have been sufficient for many tasks, they suffer from an unfortunate deficiency. Suppose we have many forms $A_1, \ldots, A_m \in \mathcal{S}^n_{>0}$ and we wish to identify them up to equivalence. A naive application of an equivalence algorithm requires $O(m^2)$ equivalence tests (in the worst case). The number of tests can be somewhat mitigated if useful invariants are available, which may or may not be the case.

Our approach in this article is to compute a *canonical form* $\mathrm{Can}_{\mathrm{GL}_n(\mathbb{Z})}(A)$ for $A \in \mathcal{S}^n_{>0}$. This canonical form should satisfy the following two basic requirements:

(i) For every $A \in \mathcal{S}^n_{>0}$, $\mathrm{Can}_{\mathrm{GL}_n(\mathbb{Z})}(A)$ is equivalent to A.

(ii) For every $A \in \mathcal{S}^n_{>0}$ and $U \in \mathrm{GL}_n(\mathbb{Z})$, $\mathrm{Can}_{\mathrm{GL}_n(\mathbb{Z})}(U^{\mathsf{T}}AU) = \mathrm{Can}_{\mathrm{GL}_n(\mathbb{Z})}(A)$.

(The equivalence in (i) is unique up to $\mathrm{Stab}(A)$.) Combining a canonical form with a hash table, the identification of equivalence classes in a list of m forms takes only m canonical form computations (and m hash table lookups) and so has the potential to be much faster.

1.2. ***Minkowski reduction and characteristic sets.*** The theory of Minkowski reduction provides one possible approach to obtain a canonical form. The Minkowski reduction domain [31] is a polyhedral domain $P_n \subset \mathcal{S}^n_{>0}$ with the property that there exists an algorithm for *Minkowski reduction*, taking as input a form A and returning as output an equivalent form in P_n. For example, for $n = 2$ we recover the familiar Gaussian reduction of binary quadratic forms. An implementation of Minkowski reduction is available [34]; however, this reduction is quite slow in practice, and it is unsuitable for forms of large dimension n (say, $n \geq 12$).

For those forms whose Minkowski reduction lies in the *interior* of the domain P_n, the Minkowski reduction is unique [7, page 203], thereby providing a canonical form. Otherwise, when the reduction lies on the boundary of P_n, there are finitely many possible Minkowski reduced forms; one can then order the facets of the polyhedral domain P_n to choose a canonical form among them. This approach was carried out explicitly by Seeber (in 1831) for $n = 3$; and, citing an unpublished manuscript, Donaldson claimed "Recently, Hans J. Zassenhaus has suggested that Minkowski reduction can be applied to the problem of row reduction of matrices of integers" [7, page 201]. An extension to $n = 5, 6, 7$ is possible at least in principle, since P_n is known in these cases [39]. However, the problem of determining the facets of the Minkowski reduction domain is hard in itself and so this strategy seems unrealistic in higher dimensions. Other reduction theories [11; 24] suffer from the same problem of combinatorial explosion on the boundary.

In contrast, the approach taken by Plesken and Souvignier [35] for computing the stabilizer and checking for equivalence of a form A uses the following notion.

Definition 1.2.1. A *characteristic vector set* function is a map that assigns to every $n \geq 1$ and form $A \in \mathcal{S}^n_{>0}$ a finite subset of vectors $\mathcal{V}(A) \subseteq \mathbb{Z}^n$ such that

(i) $\mathcal{V}(A)$ generates $\mathbb{Z}^n$ (as a $\mathbb{Z}$-module); and

(ii) for all $U \in \mathrm{GL}_n(\mathbb{Z})$, we have $U^{-1}\mathcal{V}(A) = \mathcal{V}(U^{\mathsf{T}}AU)$.

The basic idea is then given a form A to define an edge-weighted graph from a characteristic vector set $\mathcal{V}(A)$; using this graph, equivalence and automorphisms of forms becomes a problem about isomorphism and automorphisms of graphs (see Lemma 3.1.1). The graph isomorphism problem has recently been proved to be solvable in quasipolynomial time by Babai (see the exposition by Helfgott [15]); however, the current approaches to computing characteristic vector sets (including ours) use algorithms to solve the shortest vector problem which is known to be NP-hard [29], so it is difficult to take advantage of this complexity result in the general case. Nevertheless, we may hope to leverage some practical advantage from this approach.

1.3. ***Our approach.*** In this article, we adopt the approach of characteristic vector sets, using very efficient programs [17; 28] that compute a canonical form of a graph using partition backtrack. A subfield F of $\mathbb{R}$ is *computable* if it comes equipped with a way of encoding elements in bits along with deterministic, polynomial-time algorithms to test equality, to perform field operations, and to compute (binary) expansions to arbitrary precision (for generalities, see e.g., Stoltenberg-Hansen and Tucker [40]). For example, a number field with a designated real embedding is computable using standard algorithms.

Theorem 1.3.1. *There exists an explicit, deterministic algorithm that, on input a (positive definite) form $A \in \mathcal{S}^n_{>0}$ with entries in a computable subfield $F \subset \mathbb{R}$, computes a canonical form for A. For fixed $n \geq 1$, this algorithm runs in a bounded number of arithmetic operations in F and in a polynomial number of bit operations when $F = \mathbb{Q}$.*

This theorem is proven by combining Proposition 3.4.2 for the first statement and Corollary 4.1.2 for the running time analysis. The running time in Theorem 1.3.1 is exponential in n, as we rely on short vector computations; we are not aware of general complexity results, such as NP-hardness, for this problem. In light of the comments about Minkowski reduction in the previous section, the real content of Theorem 1.3.1 is in the word *explicit*. We also find this algorithm performs fairly well in practice (see Section 4.2) — an implementation is available online [1].

1.4. ***Contents.*** In Section 2 we present the construction of some characteristic vector set functions. In Section 3 we present how to construct a canonical form from a given characteristic set function. In Section 4 we consider the time complexity of our algorithm; we conclude in Section 5 with extensions and applications.

2. Construction of characteristic vector sets

In this section we build two characteristic vector set functions that can be used for the computation of the stabilizer, canonical form, and equivalence of forms.

2.1. ***Vector sets.*** The sets of vectors that we use throughout this work are based on short or shortest vectors. Given a set of vectors $\mathcal{V} \subseteq \mathbb{Z}^n$, let $\operatorname{span}(\mathcal{V})$ be the (not necessarily full) lattice spanned over $\mathbb{Z}$

by $\mathcal{V}$. For $A \in \mathcal{S}^n$ and $x \in \mathbb{R}^n$, we write

$$A[x] := x^{\mathsf{T}} A x \in \mathbb{R}. \tag{2.1.1}$$

For a form $A \in \mathcal{S}^n_{>0}$ we define the *minimum*

$$\min(A) := \min_{x \in \mathbb{Z}^n \smallsetminus \{0\}} A[x], \tag{2.1.2}$$

the set of *shortest* (or *minimal*) *vectors* and its span

$$\begin{aligned} \operatorname{Min}(A) &:= \{v \in \mathbb{Z}^n : A[v] = \min(A)\}, \\ \mathcal{L}_{\min}(A) &:= \operatorname{span}(\operatorname{Min}(A)). \end{aligned} \tag{2.1.3}$$

The set of shortest vectors satisfies the desirable transformation property

$$\operatorname{Min}(U^{\mathsf{T}} A U) = U^{-1} \operatorname{Min}(A) \tag{2.1.4}$$

for all $U \in \mathrm{GL}_n(\mathbb{Z})$. If $\operatorname{Min}(A)$ is full-dimensional, then A is called *well-rounded*.

Two obstacles remain for using $\operatorname{Min}(A)$ as a characteristic vector set:

PB1. If $n \geq 2$, then $\operatorname{span}(\operatorname{Min}(A))$ may not have rank n.

PB2. If $n \geq 5$, then $\operatorname{span}(\operatorname{Min}(A))$ may have rank n but may not equal $\mathbb{Z}^n$.

Thus we have to consider other vector sets. For $\lambda > 0$, let

$$\operatorname{Min}_A(\lambda) := \{v \in \mathbb{Z}^n \smallsetminus \{0\} : A[v] \leq \lambda\}. \tag{2.1.5}$$

The vector set used for computing the stabilizer and automorphisms in the `AUTO/ISOM` programs of Plesken and Souvignier [35] is:

$$\mathcal{V}_{\mathrm{PS}}(A) := \operatorname{Min}_A(\operatorname{maxdiag}(A)), \tag{2.1.6}$$

where $\operatorname{maxdiag}(A) := \max\{A_{ii} : 1 \leq i \leq n\}$ is the maximum of the diagonal elements of A. The vector set $\mathcal{V}_{\mathrm{PS}}(A)$ contains the standard basis as a subset and as a result is adequate for computing the stabilizer. Typically LLL-reduction [25] is used, leading to a decrease in $\operatorname{maxdiag}(A)$, to prevent large sets. However, when computing equivalence we have a potential problem since two forms A and B can be equivalent but satisfy $\operatorname{maxdiag}(A) \neq \operatorname{maxdiag}(B)$. This is a limitation of `ISOM`, which for equivalence can be resolved by taking the bound $\max\{\operatorname{maxdiag}(A), \operatorname{maxdiag}(B)\}$ (something we cannot do for our canonical form).

To prevent this problem we can use a more reliable vector set that consists of those vectors whose length is at most the minimal spanning length:

$$\begin{aligned} \mathcal{V}_{\mathrm{ms}}(A) &:= \operatorname{Min}_A(\lambda_{\min}), \text{ where} \\ \lambda_{\min} &:= \min\{\lambda > 0 : \operatorname{span}(\operatorname{Min}_A(\lambda)) = \mathbb{Z}^n\}. \end{aligned} \tag{2.1.7}$$

This vector set $\mathcal{V}_{\mathrm{ms}}(A)$ is a characteristic vector set. However, $\mathcal{V}_{\mathrm{ms}}(A)$ can still be very large, making it impractical to use.

Example 2.1.8. For example, the matrix $A_\lambda = \left(\begin{smallmatrix} 1 & 0 \\ 0 & \lambda \end{smallmatrix}\right)$ for $\lambda \geq 1$ gives

$$\mathcal{V}_{\mathrm{ms}}(A_\lambda) = \{\pm e_2\} \cup \{\pm e_1, \pm 2e_1, \ldots, \pm \lfloor \sqrt{\lambda} \rfloor e_1\}.$$

while $\{\pm e_1, \pm e_2\}$ would be adequate. This problem is related to **PB1**.

2.2. ***An inductive characteristic vector set, using closest vectors.*** Building on the observations made in the previous section, we now present a construction that deals with **PB1** and allows us to build a suitable characteristic vector set.

For a set of vectors $\mathcal{V} \subseteq \mathbb{Z}^n$, the saturated sublattice (of $\mathbb{Z}^n$) spanned by $\mathcal{V}$ is

$$\mathrm{satspan}(\mathcal{V}) := \mathbb{Q}\mathcal{V} \cap \mathbb{Z}^n. \tag{2.2.1}$$

Beyond shortest vectors, we use the *closest vector distance*: for $v \in \mathbb{Q}^n$, we define

$$\mathrm{cvd}(A, v) := \min_{x \in \mathbb{Z}^n} A[x - v] \tag{2.2.2}$$

as the minimum distance from $\mathbb{Z}^n$ to the vector v and

$$\mathrm{CV}(A, v) := \{x \in \mathbb{Z}^n : A[x - v] = \mathrm{cvd}(A, v)\} \tag{2.2.3}$$

the set of closest vectors achieving this minimum.

Characteristic and closest vector sets behave well under restriction to a sublattice. The following lemma describes this explicitly, in terms of bases.

Lemma 2.2.4. *Let $\mathcal{V}$ be a characteristic vector set function, $A \in \mathcal{S}^n_{>0}$ a form, and $L \subset \mathbb{R}^n$ a lattice of rank r. Let $B \in \mathrm{M}_{n,r}(\mathbb{R})$ be such that the columns are a $\mathbb{Z}$-basis of L; let c be in the real span of L and let $c_B := B^{-1}c \in \mathbb{R}^r$ be the unique vector such that $Bc_B = c$. Then the sets*

$$B\mathcal{V}(B^{\mathsf{T}}AB) \quad \textit{and} \quad B\,\mathrm{CV}(B^{\mathsf{T}}AB, c_B)$$

are independent of B (depending only on L, c).

Proof. The form $A|_B := B^{\mathsf{T}}AB \in \mathcal{S}^r_{>0}$ is the restriction of A to L in the basis B, so $B\mathcal{V}(A|_B)$ is the characteristic vector set of this restricted form, as elements of $L \subset \mathbb{R}^n$. Similarly, $B\,\mathrm{CV}(A|_B, c_B)$ is the set of vectors in $L \subset \mathbb{R}^n$, which are closest to c. Both sets only depend on L and are independent of the chosen basis. □

Suppose that A is well-rounded. Let $v_1, \ldots, v_n$ be a $\mathbb{Z}$-basis of the full rank lattice $\mathcal{L}_{\min}(A)$ spanned by $\mathrm{Min}(A)$ and let $B \in \mathrm{M}_{n\times n}(\mathbb{Z})$ be the matrix with columns $v_1, \ldots, v_n$. We then define

$$\mathcal{V}_{\mathrm{wr-cv}}(A) := \mathrm{Min}(A) \cup \bigcup_{c \in \mathbb{Z}^n/\mathcal{L}_{\min}(A)} (c - B\,\mathrm{CV}(B^{\mathsf{T}}AB, B^{-1}c)). \tag{2.2.5}$$

(It is possible to reduce the size of this set, e.g., by removing 0 or filtering by length.) The set $\mathcal{V}_{\mathrm{wr-cv}}(A)$ consists of the union of the shortest vectors together with the set of points in each coset closest to the origin. By Lemma 2.2.4, the set $\mathcal{V}_{\mathrm{wr-cv}}(A)$ is well-defined, independent of the choice of basis.

Furthermore it satisfies the necessary transformation property and spans $\mathbb{Z}^n$ (as a $\mathbb{Z}$-module) because it contains at least one point from each coset in $\mathbb{Z}^n/\mathcal{L}_{\min}(A)$.

For a general form A, in geometrical terms we follow the filtration defined from the minimum [4]. We define a set of vectors $\mathcal{V}_{\mathrm{cv}}(A)$ inductively (described in an algorithmic fashion), as follows:

(1) Compute the set $\mathrm{Min}(A)$ of vectors of minimal length and compute the saturated sublattice $L_1 := \mathrm{satspan}(\mathrm{Min}(A))$ spanned by these vectors.

(2) Compute a $\mathbb{Z}$-basis $v_1, \ldots, v_r$ of L_1, where r is its rank. Let $B_1 \in \mathrm{M}_{n,r}(\mathbb{R})$ be the matrix with columns $v_1, \ldots, v_r$, and let $A_1 := B_1^{\mathsf{T}} A B_1 \in \mathcal{S}^r_{>0}$. Note that A_1 is well-rounded by construction.

(3) Let $\mathrm{proj}\colon \mathbb{Z}^n \to \mathbb{R}^n$ be the orthogonal projection on $L_1^{\perp}$ with respect to the scalar product defined by A.

(4) Compute a basis $w_1, \ldots, w_{n-r}$ of $L_2 := \mathrm{proj}(\mathbb{Z}^n)$ and let $B_2 \in \mathrm{M}_{n,(n-r)}(\mathbb{R})$ the matrix with columns $w_1, \ldots, w_{n-r}$. Let $A_2 := B_2^{\mathsf{T}} A B_2$.

(5) If $r = n$, let $\mathcal{V}_{\mathrm{cv}}(A_2) := \varnothing$; otherwise, compute $\mathcal{V}_{\mathrm{cv}}(A_2)$ recursively and let

$$\mathcal{V}_{\mathrm{cv}}(A) := B_1 \mathcal{V}_{\mathrm{wr-cv}}(A_1) \cup \bigcup_{v \in B_2 \mathcal{V}_{\mathrm{cv}}(A_2)} \mathrm{CV}(A, v). \tag{2.2.6}$$

Theorem 2.2.7. *The following statements hold*:

(a) *The set $\mathcal{V}_{\mathrm{cv}}(A)$ is well-defined* (*independent of the choices of bases*).

(b) *The association $A \mapsto \mathcal{V}_{\mathrm{cv}}(A)$ is a characteristic vector set function.*

(c) *We have $\#\mathcal{V}_{\mathrm{cv}}(A) = n^{O(n)}$.*

(d) *There is an explicit, deterministic algorithm that on input A computes the set $\mathcal{V}_{\mathrm{cv}}(A)$ in $n^{O(n)}$ arithmetic operations over F. For $F = \mathbb{Q}$ it has bit complexity $n^{O(n)} s^{O(1)}$ with s the input size of A.*

Proof. We prove (a) by induction in the dimension n that $\mathcal{V}_{\mathrm{cv}}$ is a characteristic vector set. The base case $n = 0$ is trivial. For $n > 0$, note that A_1 is well rounded and A_2 has dimension at most $n - 1$ and thus $B_1 \mathcal{V}_{\mathrm{wr-cv}}(A_1)$ and $B_2 \mathcal{V}_{\mathrm{cv}}(A_2)$ are independent of the choice of basis by induction and Lemma 2.2.4. The lattice L_2 is uniquely defined by the projection.

For (b), by part (a), we may choose convenient bases. Running the algorithm for A and $A' = U^{\mathsf{T}} A U$ we can assume that $v_i' = U^{-1} v_i$ and $w_i' = U^{-1} w_i$ by using the transformation property of $\mathrm{Min}(A)$. Then $A_i' = A_i$ and $B_i' = U^{-1} B_i$ for $i = 1, 2$. We conclude by noting that CV also has the compatible transformation property

$$\mathrm{CV}(U^{\mathsf{T}} A U, U^{-1} v) = U^{-1} \mathrm{CV}(A, v). \tag{2.2.8}$$

For (c), by Keller, Martinet and Schürmann [20, Proposition 2.1] for a well-rounded lattice the index of the sublattice determined by the shortest vectors is at most $\lfloor \gamma_n^{n/2} \rfloor$ with γ_n the Hermite constant satisfying $\gamma_n^{n/2} \leq (2/\pi)^{n/2} \cdot \Gamma(2 + n/2) = n^{O(n)}$. The bound on $\mathcal{V}_{\mathrm{cv}}$ follows by combining this with exponential upper bounds on the kissing number [18] and the upper bound 2^n on $\#\,\mathrm{CV}(A, v)$ [6, Proposition 13.2.8].

The running time estimate (d) for arithmetic operations follows by combining single exponential upper estimates for algorithms to solve the CVP and SVP (see e.g., Micciancio and Voulgaris [30]). We conclude with the bit complexity analysis for $F = \mathbb{Q}$. The bit complexity of SVP and CVP algorithms is indeed polynomial time in the input size [16; 36]. (We lack a reference for more general fields, and although we do not see major obstacles doing such an analysis, it would be out of the scope of this work). For the computed projection, the Gram–Schmidt orthogonalization process also has a polynomial bit complexity in the input size (in bounded dimension, by induction). The remaining steps in computing $\mathcal{V}_{\mathrm{cv}}(A)$, including computing a basis out of a spanning set, computing a basis for the saturated sublattice, and computing representatives of the cosets $Z^n/\mathcal{L}_{\min}(A)$, are standard applications of the computation of a Hermite normal form (HNF) — see also Section 3.4. A careful HNF computation can be achieved in polynomial time in the input size [19]. In particular, the obtained basis vectors and coset representatives also have a bit size that is polynomially bounded in the input size. Thus for $F = \mathbb{Q}$ all arithmetic operations while computing $\mathcal{V}_{\mathrm{cv}}(A)$ have a bit complexity polynomial in s. We note for completeness that efficient versions of SVP, CVP, and HNF algorithms depend heavily on the famous LLL-algorithm. □

Although the cost of computing many closest vector problems may make it quite expensive to compute $\mathcal{V}_{\mathrm{cv}}(A)$ in the worst case, we find in many cases that it gives a substantial improvement in comparison to other characteristic vector sets.

Example 2.2.9. Returning to Example 2.1.8, we find that $\mathcal{V}_{\mathrm{cv}}(A_\lambda) = \{\pm e_1, \pm e_2\}$.

The construction of $\mathcal{V}_{\mathrm{cv}}$ addresses **PB1**, but **PB2** remains — even for well-rounded lattices $\#(\mathbb{Z}^n/\mathcal{L}_{\min}(A))$ can possibly be very large.

Example 2.2.10. The self-dual Niemeier lattice N_{23} [5, Chapter 18], whose root diagram is $24\mathsf{A}_1$ is well-rounded: it has minimum 2 with 48 shortest vectors, and $\#\mathcal{V}_{\mathrm{ms}}(N_{23}) = 194352$. Since the index of the lattice spanned by the shortest vectors in N_{23} is 2^{24}, the size of $\mathcal{V}_{\mathrm{cv}}(N_{23})$ is at least $48 + 2^{24}$.

Remark 2.2.11. It may be possible to deal with some cases (but still not Example 2.2.10) by working with characteristic vector sets on forms attached in a canonical way to A: for example, one could work with the *dual* form attached to A, for sometimes the dual has few minimal vectors (even if A has many).

2.3. ***A characteristic vector set, using Voronoi-relevant vectors.*** A well-known geometric shape associated to lattices is the *Voronoi cell*. The Voronoi cell is the set of all points closer to 0 with respect to A than to any other integer point. For a form A, the (open) Voronoi cell is the intersection of half-spaces

$$\mathrm{Vor}(A) := \bigcap_{x \in \mathbb{Z}^n \setminus \{0\}} H_{A,x}, \tag{2.3.1}$$

with $H_{A,x} := \{y \in \mathbb{R}^n : A[y] < A[y-x]\}$. However, almost all vectors in this intersection are superfluous, and we only consider the set of *Voronoi-relevant vectors* $\mathcal{V}_{\mathrm{vor}}(A)$, i.e., the (unique) minimal set of vectors

such that

$$\mathrm{Vor}(A) = \bigcap_{x \in \mathcal{V}_{\mathrm{vor}}(A)} H_{A,x}. \tag{2.3.2}$$

Lemma 2.3.3. *The following statements hold*:

(a) *The association $A \mapsto \mathcal{V}_{\mathrm{vor}}(A)$ is a characteristic vector set function.*

(b) *We have $\#\mathcal{V}_{\mathrm{vor}}(A) \leq 2 \cdot (2^n - 1)$.*

(c) *There is an explicit, deterministic algorithm that on input A computes the set $\mathcal{V}_{\mathrm{vor}}(A)$ in $2^{2n+o(n)}$ arithmetic operations over F. For $F = \mathbb{Q}$ it has bit complexity $2^{2n+o(n)} s^{O(1)}$ with s the input size of A.*

Proof. Property (ii) of a characteristic vector set for $\mathcal{V}_{\mathrm{vor}}$ follows from the geometric definition, fully independent of the basis. For property (i), note that for any nonzero $x \in \mathbb{Z}^n$, we have $x \notin \mathrm{Vor}(A)$, and thus there is a vector $v \in \mathcal{V}_{\mathrm{vor}}(A)$ such that $x - v$ lies strictly closer to 0 with respect to A. Repeating this (a finite amount of time by a packing argument) we eventually end up at 0 and thus x is the sum of Voronoi-relevant vectors. The remaining statements follow from Micciancio and Voulgaris [30]. □

Although this characteristic vector set has great theoretical bounds, we refrain from using it in practice: most lattices actually attain the $2 \cdot (2^n - 1)$ Voronoi bound, whereas constructions based on short and close vectors often beat the theoretical worst-case bounds and give much smaller vector sets in practice.

3. Construction of a canonical form

Suppose now that we have chosen a characteristic vector set function $\mathcal{V}$, as in Section 2.2 or 2.3. From this, we will construct a canonical form, depending on $\mathcal{V}$.

3.1. ***Graph construction.*** Given a form A, let $\mathcal{V}(A) = \{v_1, \ldots, v_p\}$. We define G_A to be the edge- and vertex-weighted complete (undirected) graph on p vertices $1, \ldots, p$ such that vertex i has weight $w_{i,i} = A[v_i]$ and the edge between i and j has weight $w_{i,j} = v_i^{\mathsf{T}} A v_j = w_{j,i}$. In other words, G_A is the weighted complete graph whose adjacency matrix is $B^{\mathsf{T}} A B$, where $B \in \mathrm{M}_{n,p}(\mathbb{R})$ is the matrix whose columns are v_i. (The graph G_A depends on $\mathcal{V}$, but we do not include it in the notation as we consider $\mathcal{V}$ fixed in this section.)

Lemma 3.1.1. *For a form $A \in \mathcal{S}^n_{>0}$ and the graph G_A constructed from a characteristic vector set $\mathcal{V}(A)$ we have a group isomorphism*

$$\mathrm{Stab}(A) \simeq \mathrm{Stab}(G_A) := \{\sigma \in S_p : w_{i,j} = w_{\sigma(i),\sigma(j)} \text{ for all } 1 \leq i, j \leq p\}. \tag{3.1.2}$$

Proof. We first define the map $\mathrm{Stab}(A) \to \mathrm{Stab}(G_A)$. Let $U \in \mathrm{Stab}(A)$. Then by property (ii) of a characteristic vector set, we have $U\mathcal{V}(A) = \mathcal{V}(U^{-\mathsf{T}} A U^{-1}) = \mathcal{V}(A)$; therefore, U permutes the set $\mathcal{V}(A)$, giving a permutation $\sigma_U \in S_p$ characterized by $\sigma_U(i) = j$ if and only if $U v_i = v_j$. Accordingly, we have

$$w_{i,j} = v_i^{\mathsf{T}} A v_j = v_i^{\mathsf{T}} U^{\mathsf{T}} A U v_j = v_{\sigma_U(i)} A v_{\sigma_U(j)} \tag{3.1.3}$$

so moreover $\sigma_U \in \mathrm{Stab}(G_A)$. It is then straightforward to see that this map defines a group homomorphism. To show this map is an isomorphism, we use property (i) that $\mathcal{V}(A)$ spans $\mathbb{Z}^n$. Indeed, the map is injective because if σ_U is the identity, then $Uv_i = v_i$ for all i so U is the identity. Similarly, it is surjective: any $\sigma \in \mathrm{Stab}(G_A)$ fixes pairwise inner products with respect to A, so we obtain a unique $\mathbb{Q}$-stabilizer $U \in \mathrm{GL}_n(\mathbb{Q})$ such that $U^{\mathsf{T}}AU = A$; however, because $\mathcal{V}(A)$ spans $\mathbb{Z}^n$, we obtain $U\mathbb{Z}^n = \mathbb{Z}^n$ so $U \in \mathrm{Stab}(A)$. □

3.2. ***Graph transformations.*** The software `nauty` [28] and `bliss` [17] allow to test equivalence and find the automorphism group and a canonical vertex ordering of vertex weighted graphs. Thus, we need graph transformations that allow to translate our vertex and edge weighted complete graphs into vertex weighted graphs (see also the `nauty` manual [28]).

Let G be a complete (undirected) graph on p vertices with vertex weights $w_{i,i}$ and edge weights $w_{i,j}$. We construct a complete (undirected) graph $T_1(G)$ on $p+2$ vertices which is only edge weighted, as follows. Let $a := 1 + \max_{i,j} w_{i,j}$ and $b := a + 1$ be two distinct weights that do not occur as $w_{i,j}$. We define the new edge weight $w'_{i,j}$ for $i < j$ to be

$$w'_{i,j} := \begin{cases} w_{i,j} & \text{if } i < j \le p, \\ w_{i,i} & \text{if } i \le p \text{ and } j = p+1, \\ a & \text{if } i \le p \text{ and } j = p+2, \\ b & \text{if } i = p+1 \text{ and } j = p+2. \end{cases} \tag{3.2.1}$$

We have a natural bijection $\mathrm{Isom}(G, G') \xrightarrow{\sim} \mathrm{Isom}(T_1(G), T_1(G'))$ of morphisms in the categories of edge-and-vertex-weighted and edge-weighted graphs, hence taking $G' = G$, we have $\mathrm{Aut}(G) \simeq \mathrm{Aut}(T_1(G))$.

The next transformation takes a complete graph G with edge weights $w_{i,j}$ and returns a vertex weighted graph $T_2(G)$. Let S be the list of possible edge weights, ordered from the smallest to the largest, and let w be the smallest integer such that $\#S \le 2^w$. For an edge weight $s \in S$, denote $l_k(s)$ the k-th value in the binary expansion of the position of s in S. If G has p vertices then $T_2(G)$ will have pw vertices of the form (i,k) with $1 \le i \le p$ and $0 \le k \le w-1$. The weight of the vertex (i,k) is k. Two vertices (i,k) and (i',k') are adjacent in the following cases:

(1) $i = i'$.

(2) $k = k'$ and $l_k(w_{i,i'}) = 1$.

Condition (i) implies that vertices of G correspond to cliques in $T_2(G)$. Condition (ii) means that each digit k corresponds to a subgraph of $T_2(G)$. We have again have a natural bijection $\mathrm{Isom}(G, G') \xrightarrow{\sim} \mathrm{Isom}(T_2(G), T_2(G'))$.

Combining this we can lift an isomorphism between $T_2(T_1(G_A))$ and $T_2(T_1(G_B))$ to an isomorphism between G_A and G_B and thus to an isomorphism between A and B by solving an overdetermined linear system. Similarly, we can compute the group $\mathrm{Aut}(A)$ from $\mathrm{Aut}(T_2(T_1(G_A)))$.

3.3. ***Canonical orderings of characteristic vector sets.*** The canonical vertex ordering functionality of `nauty` and `bliss` gives an ordering of the vertices of vertex weighted graphs. It is canonical in the sense that two isomorphic graphs will after this reordering be identical. We do not know a priori what this ordering is as it depends on the software, its version and the chosen running options. We still call it *canonical*, following standard terminology.

We need to lift the ordering of the vertex set of $T_2(T_1(G_A))$ into an ordering of the vertex set of G_A and so the characteristic vector set. Every vertex i of G corresponds to a set S_i of w vertices in $T_2(G)$ with $S_i \cap S_j = \varnothing$ for $i \neq j$. For two vertices i, j of G we set $i < j$ if and only if $\min S_i < \min S_j$ in the canonical vertex ordering of $T_2(G)$. Similarly every vertex i of G maps to one vertex $\phi(i)$ of $T_1(G)$ with $\phi(i) \neq \phi(j)$ if $i \neq j$. Thus we set $i < j$ if and only if $\phi(i) < \phi(j)$ in the canonical ordering.

Combining the above we obtain a canonical ordering of the vertex set of G_A and thus of the characteristic vector set of the matrix A.

3.4. ***Canonical form.*** We have a canonical ordering of the characteristic vector set $\mathcal{V}(A)$, which we write as $v_1, \ldots, v_p$. This ordering is only canonical up to $\mathrm{Stab}(A)$: for another canonical ordering, there is an element $S \in \mathrm{Stab}(A)$ such that $w_i = Sv_i$ for $i = 1, \ldots, p$, and conversely. We will now derive a canonical form from the vectors v_i.

The Hermite normal form (HNF) of a matrix $Q \in \mathrm{M}_{m,n}(\mathbb{Z})$ is the unique matrix $H = (h_{ij})_{i,j} \in \mathrm{M}_{m,n}(\mathbb{Z})$ for which there exists $U \in \mathrm{GL}_m(\mathbb{Z})$ such that $Q = UH$ and moreover:

(i) The first r rows of H are nonzero and the remaining rows are zero.

(ii) For $1 \leq i \leq r$, if h_{i,j_i} is the first nonzero entry in row i, then $j_1 < \cdots < j_r$.

(iii) $h_{i,j_i} > 0$ for $1 \leq i \leq r$.

(iv) If $1 \leq k < i \leq r$, then $0 \leq h_{k,j_i} < h_{i,j_i}$.

In the cases that interest us, the matrix Q_A with columns $v_1, \ldots, v_p$ defined by the characteristic vector set $\mathcal{V}(A)$ is of full rank and so the matrix U, obtained from the Hermite normal form $Q_A = UH$, is uniquely defined as well. Note that any other ordering $Sv_1, \ldots, Sv_p$ would lead to the matrix SU for some $S \in \mathrm{Stab}(A)$. We denote the matrix U by $U_{\mathcal{V}(A)}$ and note that its coset representative in $\mathrm{Stab}(A)\backslash\mathrm{GL}_n(\mathbb{Z})$ is well-defined (determined by $\mathcal{V}(A)$).

We now define

$$\mathrm{Can}_{\mathrm{GL}_n(\mathbb{Z})}(A) := U_{\mathcal{V}(A)}^{\mathsf{T}} A U_{\mathcal{V}(A)} \in \mathcal{S}_{>0}^n. \tag{3.4.1}$$

Then $\mathrm{Can}_{\mathrm{GL}_n(\mathbb{Z})}(A)$ depends only on $\mathcal{V}(A)$ and A. Proposition 3.4.2 proves the first statement of our main result, Theorem 1.3.1 (for any characteristic vector set function $\mathcal{V}$).

Proposition 3.4.2. *The matrix* $\mathrm{Can}_{\mathrm{GL}_n(\mathbb{Z})}(A)$ *is a canonical form for* A.

Proof. Property (i) is clear by definition. For (ii), given $P \in \mathrm{GL}_n(\mathbb{Z})$, we have

$$U_{\mathcal{V}(P^{\mathsf{T}}AP)} \equiv U_{P^{-1}\mathcal{V}(A)} \equiv P^{-1} U_{\mathcal{V}(A)} \in \mathrm{Stab}(P^{\mathsf{T}}AP)\backslash\mathrm{GL}_n(\mathbb{Z}). \tag{3.4.3}$$

Thus $\mathrm{Can}_{\mathrm{GL}_n(\mathbb{Z})}(P^{\mathsf{T}}AP) = \mathrm{Can}_{\mathrm{GL}_n(\mathbb{Z})}(A)$, as desired. □

Remark 3.4.4. An alternative to computing the canonical form would be to keep the canonicalized version of the graph G_A. However, this graph can be quite large, and the positive definite form allows a more compact representation even taking into account coefficient explosion that might occur with the Hermite normal form.

4. Analysis

4.1. ***Theoretical time complexity.*** We now analyze the algorithmic complexity of computing a canonical form using the characteristic vector set in Section 2.3.

Theorem 4.1.1. *Given as input a positive definite symmetric matrix $A \in \mathcal{S}^n_{>0}$ with entries in a computable subfield $F \subset \mathbb{R}$, and a characteristic vector set $\mathcal{V}(A)$, we can compute a canonical form for A in time* $\exp(O(\log(N)^c) + s^{O(1)}$ *where $N := \#\mathcal{V}(A)$, s is the input size of $(A, \mathcal{V}(A))$, and $c > 1$ is a constant.*

Proof. Given the characteristic vector set $\mathcal{V}(A)$ the corresponding graph can be computed in time polynomial in the input size of A and $\mathcal{V}(A)$ as this part is mostly dominated by the computation of $v^{\mathsf{T}}Aw$ for $v, w \in \mathcal{V}(A)$. Computing a Hermite normal form can be done in time polynomial in the matrix input size which is the same as $\mathcal{V}(A)$ [19]. Because the initial graph has at most $O(N^2)$ distinct weights the final constructed vertex-weighted graph $T_2(T_1(G_A))$ is of polynomial size in N. We can conclude if we have a quasipolynomial algorithm to find a canonical form of a graph. For this we refer to a recent report by Babai [14]. □

Corollary 4.1.2. *For all $n \geq 1$ and $A \in \mathcal{S}^n_{>0}$ with entries in a computable subfield $F \subset \mathbb{R}$, we can compute a canonical form in at most $2^{O(n^c)}$ arithmetic operations in F for some constant $c > 1$. If $F = \mathbb{Q}$, the bit complexity is at most $2^{O(n^c)} + s^{O(1)}2^{O(n)}$ with s the input size of A.*

Proof. By Lemma 2.3.3 we have an characteristic vector set function $\mathcal{V}_{\mathrm{vor}}$ such that $\mathcal{V}_{\mathrm{vor}}(A)$ has cardinality at most $2(2^n - 1)$ and can be computed in at most $2^{O(n)}$ arithmetic operations. For the rational case, the bit complexity (and output size) is at most $s^{O(1)}2^{O(n)}$, with s the input size of A. We conclude by Theorem 4.1.1. □

4.2. ***Practical time complexity.*** We give a short experimental review of the practial time complexity of our implementation [1]. We selected a diverse set of test cases to benchmark our implementation: random forms, more than 500 000 perfect forms [8] and more than 100 special forms from the *catalog of lattices* [32]. For the random n-dimensional forms a basis matrix B is constructed with entries uniform from $\{-n, \ldots, n\}$, which, if full rank, is turned into a form $A = B^T B$. The set of perfect forms contains all 10 963 perfect forms of dimension 2 up to 8 and in addition 524 288 perfect forms of dimension 9. The set of special forms consists of a diverse subset from the catalog up to dimension 16, including all laminated lattices. Up to dimension 20 we used 32-bit integers and above that (much slower) arbitrary precision integers to prevent overflow. The implementation currently supports the characteristic vector

			Time (s)			$\#\mathcal{V}_{\text{ms}}$		
Type	Samples	n	min	avg	max	min	avg	max
Perfect	10 963	2–8	0.00041	0.0032	0.086	6	73.74	240
	524 288	9	0.0039	0.00594	0.11	90	94.04	272
Random	100	10	0.0015	0.08	2.03	20	100.36	988
	100	20	0.016	0.17	4.18	40	114.34	812
	100	30	2.43	23.41	511.42	60	93.46	310
	100	40	5.18	24.91	251.51	82	107.7	240
Catalog	107	2–16	0.00018	2.12	36.71	4	630.47	4320

Table 1. Timings of our implementation [1].

set function $\mathcal{V}_{\text{ms}}$ and has not been highly optimized. The main bottleneck seemed to be constructing the characteristic vector sets and the computation of all pairwise inner products (in arbitrary precision) for the graph. Perhaps surprisingly, determining the canonical graph itself took negligible time in most cases. In low dimensions where we can still use basic integer types, computing a canonical form takes a few milliseconds up to a few seconds. For random lattices we can expect relatively small characteristic sets even in large dimensions, therefore enumerating the minimal vectors quickly becomes the bottleneck in high dimensions. For special forms in higher dimensions such as the Leech lattice with 196 560 minimal vectors one can expect that the main bottleneck is related to the huge graph. Both storing the graph and computing a canonical representative might barely be in the feasible regime.

5. Extensions and applications

We conclude with an extension and a description of some applications.

5.1. ***Extension to symplectic groups.*** Let $J_n := \left(\begin{smallmatrix} 0 & I_n \\ -I_n & 0 \end{smallmatrix}\right)$ represent the standard alternating pairing and

$$\mathrm{Sp}_{2n}(\mathbb{Z}) := \{Q \in \mathrm{GL}_{2n}(\mathbb{Z}) : Q^{\mathsf{T}} J_n Q = J_n\}. \tag{5.1.1}$$

The group $\mathrm{Sp}_{2n}(\mathbb{Z})$ acts on $\mathcal{S}^{2n}_{>0}$ and we seek a canonical form for this action [27].

Theorem 5.1.2. *Given a ordered set of vectors $\mathcal{V} = (v_1, \ldots, v_m)$ that generates $\mathbb{Z}^{2n}$ as a lattice, there exists an effectively computable symplectic basis* $\mathrm{SympBas}(\mathcal{V})$ *of $\mathbb{Z}^{2n}$ such that for every $P \in \mathrm{Sp}_{2n}(\mathbb{Z})$ we have* $\mathrm{SympBas}(\mathcal{V}P) = \mathrm{SympBas}(\mathcal{V})P$.

Proof. Let w_1 be the first nonzero vector in $\mathcal{V}$ divided by the gcd of its coefficients. Since the family of vectors spans $\mathbb{Z}^n$, the gcd of the symplectic products $\omega(w_1, v_j)$ is 1. Thus we can find in a deterministic manner integers α_i such that $w_{2n} = \sum_{i=1}^{m} \alpha_i v_i$ satisfies $\omega(w_1, w_{2n}) = 1$. We can then replace the vectors v_i of the vector family by $v_i' = v_i - \omega(v_i, w_{2n})w_1 + \omega(v_i, w_1)w_{2n}$. They satisfy $\omega(v_i', w_1) = \omega(v_i', w_{2n}) = 0$. Thus we apply the same construction inductively on them and get our basis. The invariance property follows from the fact that we never use specific coordinate systems. □

A canonical representative for a form $A \in \mathcal{S}^{2n}_{>0}$ under the action of $\mathrm{Sp}_{2n}(\mathbb{Z})$ can also be computed using our canonical form, as follows:

(1) Compute a characteristic vector family using e.g., $\mathcal{V}_{\mathrm{cv}}$.

(2) Compute a graph on this characteristic set of vector by assigning to two vectors v, v' the weight (vAv', vJ_nv').

(3) Apply the canonicalization procedure and get a canonical ordering of $\mathcal{V}_{\mathrm{cv}}$.

(4) Use Theorem 5.1.2 in order to get a symplectic basis which then gives a reduction matrix.

5.2. ***Lattice databases.*** Several efforts have sought to enumerate lattice genera of either bounded discriminant or satisfying some arithmetic conditions such as small (spinor) class number. For example, the Brandt–Intrau tables [9] of reduced ternary forms with discriminant up to 1000, Nipp's tables [33] of positive definite primitive quaternary quadratic forms with discriminant up to 1732, and more recently the complete table of lattices with class number one due to Kirschmer and Lorch [22], to name a few. A current project of interest in number theory is an extension of the L-functions and modular forms database (LMFDB) [26] to include lattices.

The general strategy for generating these tables can take several forms. For example, a list of isometry class candidates can be generated by extending lattices of lower rank in some systematic way [9; 33]. Classes can also be generated by Kneser's method of neighboring lattices [38] (see Section 5.4 below). Although the completeness of the list of genus representatives can be verified using the Minkowski–Siegel mass formula, one critical bottleneck in most of these schemes is eliminating redundancy in the lists generated, especially for lattices with high rank and class number — it is here where we profit significantly from a canonical form.

Another current shortcoming of the database has been the lack of a deterministic naming scheme for lattices. Although lattices up to equivalence can be classified by dimension, determinant, level, and class number, beyond that point many genera of such lattices can exist, and each genus can potentially contain multiple classes. Finding a canonical form for lattices provides a way to establish a deterministic labeling. This has long been known to be a challenge: for example, it is exactly the problem of the boundary of a fundamental domain in Minkowski reduction (mentioned in the introduction) that is at issue. Ad hoc enumeration and labeling suffers from the deficiency that a computer failure or other issues in the database could result in new and different enumeration. A canonical form provides a mechanism for a canonical label for lattices. Such a scheme would still depend on the graph canonical form being called in the algorithm; but in the event of a switch a bijective dictionary could easily be stored between the new naming and the old, giving still a nearly permanent deterministic naming of lattices.

5.3. ***Application to enumeration of perfect forms.*** A canonical form really shows its strength compared to pairwise equivalence checks when the number of forms to be classified becomes very large. This is certainly the case during the enumeration of perfect forms using Voronoi's algorithm in dimension 9 or higher. In dimension 9 already more than 20 million (inequivalent) perfect forms are found and the total

number could be on the order of half a billion [42]. Even though there are some useful invariants such as the number of miminal vectors, the determinant and the size of the automorphism group, the number of remaining candidates for equivalence for each found perfect form can become quite large. Removing equivalent forms is a large part of the computational cost during the enumeration.

Therefore, efficiently finding a canonical form seems to be a necessity in completing the full enumeration in dimensions 9 or higher. Luckily by the definition of a perfect form we always have that $\mathrm{Min}(A)$ is full dimensional. Furthermore for all perfect forms found so far $\mathrm{Min}(A)$ also spans $\mathbb{Z}^n$ and therefore the function $\mathcal{V}_{\mathrm{ms}}$ seems to be an efficient way to obtain a small characteristic vector set. In Section 4.2 we saw that computing a canonical perfect form in dimension 9 takes just a few milliseconds.

5.4. ***Application to algebraic modular forms.*** Finally, we present an application to speed up computations of orthogonal modular forms, a special case of the theory of *algebraic modular forms* as defined by Gross [12]. We shift our perspective slightly, varying lattices in a (fixed) quadratic space.

Let $L \subset V$ be a *(full) lattice*, the $\mathbb{Z}$-span of a $\mathbb{Q}$-basis for V. We say L is *integral* if $x^{\mathsf{T}}Ay \in \mathbb{Z}$ for all $x, y \in L$, and suppose that L is integral. We represent L in bits by a basis $\{v_1, \ldots, v_n\}$; letting U_L be the change of basis matrix, we obtain a form

$$A_L := (v_i^{\mathsf{T}} A v_j)_{1 \le i,j \le n} = U_L^{\mathsf{T}} A U_L. \tag{5.4.1}$$

(It is not necessarily the case that A_L is arithmetically equivalent to A — the change of basis need only belong to $\mathrm{GL}_n(\mathbb{Q})$.)

In order to organize these lattices, we define the *orthogonal group*

$$\mathrm{O}(V) := \{P \in \mathrm{GL}_n(\mathbb{Q}) : P^{\mathsf{T}} A P = A\}. \tag{5.4.2}$$

Integral lattices $L, L' \subset V$ are *isometric*, written $L \simeq L'$, if there exists $P \in \mathrm{O}(V)$ such that $P(L) = L'$. Choosing bases for L, L', we see that $L \simeq L'$ if and only if A_L and $A_{L'}$ are arithmetically equivalent.

We repeat these definitions replacing $\mathbb{Q}$ (and $\mathbb{Z}$) by $\mathbb{Q}_p$ (and $\mathbb{Z}_p$) for a prime p, abbreviating $L_p := L \otimes_{\mathbb{Z}} \mathbb{Z}_p$. Then the *genus* of L is

$$\mathrm{Gen}(L) := \{L' \subset V : L_p \simeq L'_p \text{ for all primes } p\}. \tag{5.4.3}$$

Finally, we define the *class set* $\mathrm{Cls}(L)$ as the set of isometry classes in $\mathrm{Gen}(L)$. By the geometry of numbers, we have $\#\,\mathrm{Cls}(L) < \infty$.

The theory of p-neighbors, due originally to Kneser [23], gives an effective method to compute representatives of the class set $\mathrm{Cls}(L)$, as follows. Let p be prime (allowing $p = 2$) not dividing $\det(A_L)$. We say that a lattice $L' < V$ is a *p-neighbor* of L, and write $L' \sim_p L$, if L' is integral and

$$[L : L \cap L'] = [L' : L \cap L'] = p \tag{5.4.4}$$

(index as abelian groups). If $L \sim_p L'$, then $\mathrm{disc}(L) = \mathrm{disc}(L')$ and $L' \in \mathrm{Gen}(L)$ [10, Lemma 5.7]. The set of p-neighbors can be computed in time $O(p^{m+\epsilon} H_n(s))$, where s is the input size and H_n

is a polynomial depending on n. Moreover, by strong approximation [10, Theorem 5.8], there is an effectively computable finite set S of primes such that every $[L'] \in \mathrm{Cls}(L)$ is an *iterated S-neighbor* $L \sim_{p_1} \cdots \sim_{p_r} L_r \simeq L'$ with $p_i \in S$. Typically, we may take $S = \{p\}$ for any $p \nmid \mathrm{disc}(L)$. In this way, we may compute a set of representatives for $\mathrm{Cls}(L)$ from iterated S-neighbors.

The space of *orthogonal modular forms for L* (with trivial weight) is

$$M(\mathrm{O}(L)) := \mathrm{Map}(\mathrm{Cls}(L), \mathbb{C}). \tag{5.4.5}$$

In the basis of characteristic functions $\delta_{[L']}$ for $[L'] \in \mathrm{Cls}(L)$ we have $M(\mathrm{O}(L)) \simeq \mathbb{C}^h$ where $h := \#\mathrm{Cls}(L)$. For $p \nmid \mathrm{disc}(L)$, define the *Hecke operator*

$$\begin{aligned} T_p \colon M(\mathrm{O}(L)) &\to M(\mathrm{O}(L)) \\ T_p(f)([L']) &= \sum_{M' \sim_p L'} f([M']). \end{aligned} \tag{5.4.6}$$

The operators T_p commute and are self-adjoint (with respect to a natural inner product); accordingly, there exists a basis of simultaneous eigenvectors for the Hecke operators, called *eigenforms*.

In this way, to compute the matrix representing the Hecke operator T_p, for each $[L'] \in \mathrm{Cls}(L)$, we need to identify the isometry classes of the p-neighbors of L'. Here is where our canonical form algorithm applies, returning to our original motivation: after computing canonical forms for $\mathrm{Cls}(L)$, for each p-neighbor, we compute their canonical forms and then a hash table look up on $\mathrm{Cls}(L)$. This reduces our computation from $O(h^2)$ *isometry tests* to $O(h)$ *hash table lookups*. For medium-sized values of n, we hope that the use of canonical forms will allow us to peer more deeply into the world of automorphic forms on orthogonal groups.

Acknowledgments

This work was advanced during the conference *Computational Challenges in the Theory of Lattices* at the Institute for Computational and Experimental Research in Mathematics (ICERM) and further advances were made during a visit to the Simons Institute for the Theory of Computing. The authors would like to thank ICERM and Simons for their hospitality and support. Voight was supported by a Simons Collaboration grant (550029) and Van Woerden was supported by the ERC Advanced Grant 740972 (ALGSTRONGCRYPTO). We also thank Achill Schürmann and Rainer Schulze-Pillot for help on Minkowski reduction theory and the anonymous referees for their detailed feedback.

References

[1] *Polytopes, lattices and quadratic forms programs*, https://github.com/MathieuDutSik/polyhedral_common, 2018.

[2] E. Bayer-Fluckiger and I. Suarez, *Modular lattices over cyclotomic fields*, J. Number Theory **114** (2005), no. 2, 394–411.

[3] D. Bremner, M. Dutour Sikirić, D. V. Pasechnik, T. Rehn, and A. Schürmann, *Computing symmetry groups of polyhedra*, LMS J. Comput. Math. **17** (2014), no. 1, 565–581.

[4] B. Casselman, *Stability of lattices and the partition of arithmetic quotients*, Asian J. Math. **8** (2004), no. 4, 607–637.

[5] J. H. Conway and N. J. A. Sloane, *Sphere packings, lattices and groups*, third ed., Grundlehren Math. Wiss., vol. 290, Springer-Verlag, New York, 1999.

[6] M. M. Deza and M. Laurent, *Geometry of cuts and metrics*, Algorithms and Combinatorics, vol. 15, Springer, Heidelberg, 2010.

[7] J. L. Donaldson, *Minkowski reduction of integral matrices*, Math. Comp. **33** (1979), no. 145, 201–216.

[8] M. Dutour Sikirić, A. Schürmann, and F. Vallentin, *Classification of eight-dimensional perfect forms*, Electron. Res. Announc. Amer. Math. Soc. **13** (2007), 21–32.

[9] K. Germann, *Tabellen reduzierter, positiver quaternärer quadratischer Formen*, Comment. Math. Helv. **38** (1963), 56–83.

[10] M. Greenberg and J. Voight, *Lattice methods for algebraic modular forms on classical groups*, Computations with modular forms, Contrib. Math. Comput. Sci., vol. 6, Springer, Cham, 2014, 147–179.

[11] D. Grenier, *Fundamental domains for the general linear group*, Pacific J. Math. **132** (1988), no. 2, 293–317.

[12] B. H. Gross, *Algebraic modular forms*, Israel J. Math. **113** (1999), no. 1, 61–93.

[13] I. Haviv and O. Regev, *On the lattice isomorphism problem*, Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, 2014, 391–404.

[14] L. Babai, *Canonical form for graphs in quasipolynomial time: preliminary report*, Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, 2019, 1237–1246.

[15] H. A. Helfgott, *Isomorphismes de graphes en temps quasi-polynomial*, Séminaire Bourbaki, vol. 2016/2017, Astérisque no. 407 (2019), exp. no. 1125, 135–182.

[16] B. Helfrich, *Algorithms to construct Minkowski reduced and Hermite reduced lattice bases*, Theoretical Computer Science **41** (1985), 125–139.

[17] T. Junttila and P. Kaski, *bliss*, http://www.tcs.hut.fi/Software/bliss/.

[18] G. A. Kabatjanskiĭ and V. I. Levenšteĭn, *Bounds for packings on the sphere and in space*, Problemy Peredači Informacii **14** (1978), no. 1, 3–25.

[19] R. Kannan and A. Bachem, *Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix*, SIAM J. Comput. **8** (1979), no. 4, 499–507.

[20] W. Keller, J. Martinet, and A. Schürmann, *On classifying Minkowskian sublattices*, Math. Comp. **81** (2012), no. 278, 1063–1092, with an appendix by M. Dutour Sikirić.

[21] M. Kirschmer, *One-class genera of maximal integral quadratic forms*, J. Number Theory **136** (2014), 375–393.

[22] M. Kirschmer and D. Lorch, *Ternary quadratic forms over number fields with small class number*, J. Number Theory **161** (2016), 343–361.

[23] M. Kneser, *Klassenzahlen definiter quadratischer Formen*, Archiv der Mathematik **8** (1957), no. 4, 241–250.

[24] A. N. Korkin and E. I. Zolotarev, *Sur les formes quadratiques*, Math. Ann. **6** (1873), no. 1, 366–389.

[25] A. K. Lenstra, H. W. Lenstra, and L. Lovász, *Factoring polynomials with rational coefficients*, Math. Ann. **261** (1982), 515–534.

[26] The LMFDB Collaboration, *The L-functions and Modular Forms Database*, http://www.lmfdb.org, 2020.

[27] R. MacPherson and M. McConnell, *Explicit reduction theory for Siegel modular threefolds*, Invent. Math. **111** (1993), no. 3, 575–625.

[28] B. D. McKay and A. Piperno, *nauty and Traces*, http://cs.anu.edu.au/people/bdm/nauty/.

[29] D. Micciancio, *The shortest vector problem is NP-hard to approximate to within some constant*, SIAM J. Comput. **30** (2001), 2008–2035.

[30] D. Micciancio and P. Voulgaris, *A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations*, SIAM J. Comput. **42** (2013), no. 3, 1364–1391.

[31] H. Minkowski, *Diskontinuitätsbereich für arithmetische äquivalenz*, J. Reine Angew. Math. **129** (1905), 220–274.

[32] G. Nebe and N. Sloane, *A catalogue of lattices*, http://www.math.rwth-aachen.de/~Gabriele.Nebe/LATTICES/index.html.

[33] G. L. Nipp, *Quaternary quadratic forms*, Springer-Verlag, New York, 1991.

[34] J. Opgenorth, W. Plesken, and T. Schulz, *CARAT, Crystallographic AlgoRithms And Tables*, v. 2.1b1 (2008), https://github.com/lbfm-rwth/carat/.

[35] W. Plesken and B. Souvignier, *Computing isometries of lattices*, Computational algebra and number theory (London, 1993), J. Symbolic Comput. **24** (1997), no. 3-4, 327–334.

[36] X. Pujol and D. Stehlé, *Rigorous and efficient short lattice vectors enumeration*, Advances in cryptology–ASIACRYPT 2008, Lecture Notes in Comput. Sci., vol. 5350, Springer, Berlin, 2008, 390–405.

[37] S. Schönnenbeck, *Simultaneous computation of Hecke operators*, J. Algebra **501** (2018), 571–597.

[38] R. Schulze-Pillot, *An algorithm for computing genera of ternary and quaternary quadratic forms*, Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation, ISSAC '91, ACM, 1991, 134–143.

[39] A. Schürmann, *Computational geometry of positive definite quadratic forms*, University Lecture Series, vol. 48, Amer. Math. Soc., Providence, RI, 2009.

[40] V. Stoltenberg-Hansen and J. V. Tucker, *Computable rings and fields*, Handbook of Computability Theory, Elsevier (1999), 363–447.

[41] A. Storjohann and G. Labahn, *Asymptotically fast computation of Hermite normal forms of integer matrices*, Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation, ISSAC '96, ACM, New York, 1996, 259–266.

[42] W. P. J. van Woerden, *Perfect quadratic forms: an upper bound and challenges in enumeration*, Master's thesis, Leiden University, 2018.

Received 28 Feb 2020.

MATHIEU DUTOUR SIKIRIĆ: mathieu.dutour@gmail.com
Institut Rudjer Bošković, Zagreb, Croatia

ANNA HAENSCH: annahaensch@gmail.com
Department of Mathematics and Computer Science, Duquesne University, Pittsburgh, PA, United States

JOHN VOIGHT: jvoight@gmail.com
Department of Mathematics, Dartmouth College, Hanover, NH, United States

WESSEL P.J. VAN WOERDEN: wessel.van.woerden@cwi.nl
Centrum Wiskunde & Informatica (CWI), Amsterdam, Netherlands

msp

VOLUME EDITORS

Stephen D. Galbraith
Mathematics Department
University of Auckland
New Zealand
https://orcid.org/0000-0001-7114-8377

The cover image is based on an illustration from the article "Supersingular curves with small noninteger endomorphisms", by Jonathan Love and Dan Boneh (see p. 9).

The contents of this work are copyrighted by MSP or the respective authors. All rights reserved.

Electronic copies can be obtained free of charge from http://msp.org/obs/4 and printed copies can be ordered from MSP (contact@msp.org).

The Open Book Series is a trademark of Mathematical Sciences Publishers.

ISSN: 2329-9061 (print), 2329-907X (electronic)

ISBN: 978-1-935107-07-1 (print), 978-1-935107-08-8 (electronic)

First published 2020.

MATHEMATICAL SCIENCES PUBLISHERS
798 Evans Hall #3840, c/o University of California, Berkeley CA 94720-3840
contact@msp.org **http://msp.org**

THE OPEN BOOK SERIES 4

Fourteenth Algorithmic Number Theory Symposium

The Algorithmic Number Theory Symposium (ANTS), held biennially since 1994, is the premier international forum for research in computational and algorithmic number theory. ANTS is devoted to algorithmic aspects of number theory, including elementary, algebraic, and analytic number theory, the geometry of numbers, arithmetic algebraic geometry, the theory of finite fields, and cryptography.

This volume is the proceedings of the fourteenth ANTS meeting, which took place 29 June to 4 July 2020 via video conference, the plans for holding it at the University of Auckland, New Zealand, having been disrupted by the COVID-19 pandemic. The volume contains revised and edited versions of 24 refereed papers and one invited paper presented at the conference.

TABLE OF CONTENTS